

A Reinforcement Learning Framework for Trajectory Prediction Under Uncertainty and Budget Constraint

Truc Viet Le¹ and Siyuan Liu² and Hoong Chuin Lau³

Abstract. We consider the problem of trajectory prediction, where a trajectory is an *ordered* sequence of location visits and corresponding timestamps. The problem arises when an agent makes sequential decisions to visit a set of spatial locations of interest. Each location bears a stochastic utility and the agent has a limited budget to spend. Given the agent’s observed *partial* trajectory, our goal is to predict the agent’s remaining trajectory. We propose a solution framework to the problem that incorporates both the stochastic utility of each location and the budget constraint. We first cluster the agents into groups of homogeneous behaviors called “agent types”. Depending on its type, each agent’s trajectory is then transformed into a discrete-state sequence representation. Based on such representations, we use reinforcement learning (RL) to model the underlying decision processes and inverse RL to learn the utility distributions of the spatial locations. We finally propose two decision models to make predictions: one is based on long-term optimal planning of RL and another uses myopic heuristics. We apply the framework to predict real-world human trajectories collected in a large theme park and are able to explain the underlying processes of the observed actions.

1 Introduction

How does a rational agent decide to visit a set of locations in space? Assuming there are distinct points of interest (POIs), then the act of visiting them has to happen sequentially. We call it *spatial* sequential decision-making. It is reasonable to assume that each location bears a non-negative utility (reward) to the decision-maker that would not be fully realized until it is visited. Until then, utilities remain *uncertain* and reflect the agent’s prior preferences. When making sequential decisions, a rational agent should also weigh in the long-term costs of visiting each of the locations in order to make an optimal plan, where “costs” here are assumed to be proportional to physical distances. Hence, answering the question above would require a model of the agent’s sequential decisions for selecting locations, whose utilities remain uncertain and costs are dynamic, and weighing in their long-term consequences into the decision-making [15].

In practice, the agent typically has a limited amount of resources (e.g., time) to run its plan, which we call a *budget*. Such a budget constraint can significantly shape the agent’s decision-making process and outcomes in non-obvious ways. In this paper, we propose a framework based on reinforcement learning [24] to model the agent’s

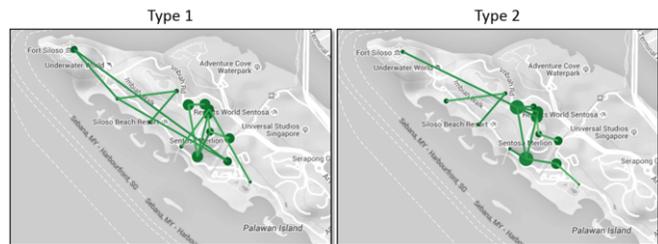


Figure 1: Visualizing the attractiveness of the same set of POIs in a real-world theme park environment (to be discussed in Sect. 7) and the pairwise transition probabilities (only those probabilities ≥ 0.20 are drawn) between them as observed by two groups of agents: “Type 1” and “Type 2”. Each group is given a certain amount of time budget to visit the set of POIs, where Type 1 has, on average, 114 minutes more than Type 2. The size of each POI is drawn to reflect its relative popularity (attractiveness) among members of each group.

spatial sequential decision-making, taking into account the uncertainty of the utilities and the budget constraint. Using the framework, we could discover the underlying processes that drive real-world behaviors such as the condition for making long-term optimal decisions in the defined setting. Indeed, traditional economic view of rational decision-making as solving an optimization problem often fails to predict reality due to *bounded rationality* [10]. Such discoveries would give insights into real-world human behaviors and help bridge the gap between human and machine intelligence [28, 15].

Our motivation comes from the problem of predicting the *next* sequence of location visits (called *trajectory*) of a mobile agent knowing its current trajectory and past observed trajectories of other similar agents. Accurate predictions of the agent’s next locations can enable numerous applications of location-based services such as real-time prediction of visitor arrivals and congestion at POIs or devising real-time advertising strategies or adaptive recommendation system for a mobile agent knowing its probable future trajectory.

Consider the example illustrated in Fig. 1, whose data were collected from real-world human behaviors in a theme park (to be discussed in Sect. 7). In this setting, suppose there are two groups of agents (human visitors) of equivalent sizes called “type 1” and “type 2”. Each agent in each group is to visit the same set of POIs within a given time frame (budget). Agent type 1 is given, on average, 114 minutes more than type 2. Such a budget difference can translate into starkly different behaviors as illustrated in the figure. Not only is the relative attractiveness of each of the POIs different, but the pairwise transition probabilities among them also become discernibly distinct. Type 1 appears to have a larger “coverage” of the POIs through their sequential transitions, while type 2 tends to visit those POIs that are

¹ Singapore Management University, 80 Stamford Rd., Singapore 178902. E-mail: trucviet.le.2012@smu.edu.sg

² Smeal College of Business, Pennsylvania State University, PA 16802, USA. E-mail: siyuan@psu.edu

³ Singapore Management University, 80 Stamford Rd., Singapore 178902. E-mail: hclau@smu.edu.sg

clustered together. These observations reflect the inherently different underlying decision processes used by these agent types. Thus, in order to make accurate trajectory predictions, it suffices to model the sequential decision-making process of each group separately.

In this paper, we develop on and extend the capabilities of the framework proposed by Le *et al.* [17] for spatial decision modeling. Specifically, we set out to predict an *ordered* sequence of an agent’s future locations (as opposed to an *unordered* bundle). Furthermore, our novel contribution is that we do not rely solely on a generative model as previously proposed to generate sequential actions (e.g., naive Bayes [14] or hidden Markov models (HMMs) [20]). Instead, we integrate one of such (i.e., HMMs) into a reinforcement learning framework to model an agent’s sequential decisions. We further propose decision models based on the learned utilities resulted from the framework for trajectory prediction. Doing so enables us to explain the underlying processes of the predicted outcomes, the effects of budget constraint on decision-making, and evaluate the appropriateness of the proposed decision models.

We summarize our contributions as follows:

- We model the sequential decision process of an agent in an integrated framework to predict its trajectory;
- Our framework takes into account both the stochasticity of rewards and the budget constraint;
- We propose two decision models for prediction: one is based on long-term optimal planning of reinforcement learning and another uses myopic heuristics;
- We empirically evaluate our framework using real-world human trajectories with compelling results.

2 Related Work

Trajectory prediction. The problem of predicting the future location(s) of a mobile agent is not entirely new. Krumm and Horvitz [14] propose a naive Bayes model called *Predestination* to predict the final destination of a driving trip given its partially observed GPS trajectory. In most recent work, some form of Markov model is often used to learn the observed transitions and infer future locations. For example, Mathew *et al.* [20] use hidden Markov models (HMMs) to identify clusters of locations from raw GPS data and Gambs *et al.* [9] propose a mobility model based on Markov chains to incorporate knowledge of the previous n visited locations. We also employ HMMs in our framework, but in a radically different way: to represent the environment in which the agent interacts with. Recently, Le *et al.* [17] propose a framework based on revealed preference learning to predict *unordered* bundles of spatial locations given an agent’s budget information. In this respect, our work expands on [17] for predicting *ordered* sequences of spatial decisions.

Sequential decisions. Modeling human sequential actions has been traditionally studied in the domain of human-computer interaction. For instance, mining sequential behaviors has been used to discover mobile users that share similar habits [19], or to imitate human behaviors in order to provide better automated care to the disabled and the elderly [11]. In this respect, modeling sequential decisions as Markov processes is commonly used to simplify the representation of the user’s knowledge [26]. A common shortcoming among these work is the lack of modeling of the users’ underlying decision processes in order to explain the discovered patterns.

Reinforcement learning. Understanding human behaviors requires finding the reward function that motivates the observed actions. Inverse reinforcement learning (IRL), first proposed by Russell

[22], provides an elegant framework to identify the reward function being optimized by the agents given observations of their activities. Ng and Russell [21] propose the original algorithms to tackle the problem based on linear programming. Ever since, there has been a wealth of algorithms developed to solve IRL [25]. IRL has enjoyed diverse applications in automated control systems that try to imitate the behaviors of expert human users (a.k.a. “learning from demonstrations”) such as learning how to drive [2], controlling helicopters [1], and predicting mouse movements [26]. In this respect, our framework integrates IRL to model the stochasticity of rewards.

3 Problem Statement

We consider a set \mathcal{D} of agents and a finite set \mathcal{G} ($|\mathcal{G}| = n$) of POIs (locations). Each agent $i \in \mathcal{D}$ has a utility vector u_i over each location $j \in \mathcal{G}$, where $u_{ij} \in \mathbb{R}_{\geq 0}$ is the utility of j to i . Agent i has a budget constraint B_i and wishes to visit a subset $s_i \subseteq \mathcal{G}$ such that $\sum_{j \in s_i} c_{ij} \leq B_i$, where c_{ij} is i ’s cost of visiting j . We denote s_i as agent i ’s *trajectory* that contains the *ordered* sequence of locations visited by i and the corresponding timestamps. Without loss of generality, we assume throughout that the costs and budget constraint are in terms of travel time and i makes a binary decision vector $s_i \in \{0, 1\}^n$. Hence, c_{ij} is a *dynamic* cost for each j that depends on the previous location in the sequence. We additionally assume the proportionality between distance and travel time, where all distances considered in this paper are spatial Euclidean distance.

Suppose \mathcal{D} can be divided into non-overlapping subsets called *agent types*, where each “type” implies homogeneous preferences and behaviors. Given an agent of a certain type, his partial trajectory (say the first k location visits) and the current budget, our goal is to predict the agent’s remaining trajectory. The notion of agent type comes from the idea that modeling each individual agent is impractical. It is much more feasible to divide them into finite and disjoint *clusters* of similar preferences and behaviors. Thus, we also use the terms “cluster” and “(agent) type” interchangeably.

Predicting an agent’s remaining trajectory requires sequential decision modeling under uncertainty and budget constraint. The uncertainty comes from the utility distributions of the remaining locations. While the relative attractiveness of the locations can be easily worked out using a simple frequency count, it is not straightforward how to learn their utility distributions from the observed trajectories and how to incorporate them into a sequential decision-making model.

4 Solution Overview

We propose an integrated framework to model and predict the next sequence of locations given an agent’s observed partial trajectory and budget constraint. The framework consists of two components: learning and prediction. Fig. 2 illustrates the overall framework. Table 1 summarizes the notations used in this paper.

Learning. We first divide the agents in the training set \mathcal{S} into K finite clusters, where each cluster Cl_j ($1 \leq j \leq K$) represents an agent type. K is typically chosen heuristically via some clustering coefficient (e.g., the silhouette index). Using the agents’ observed features and the K clusters as class labels, we train a multi-class classifier (e.g., multinomial logistic regression). We also model the environment that the agents interact with as a finite set of states S , where each state $s \in S$ has a distinct vector of features \mathbf{f}_s . We use hidden Markov models (HMMs) to transform the observed trajectories into finite sequences of states. Such a representation can then be

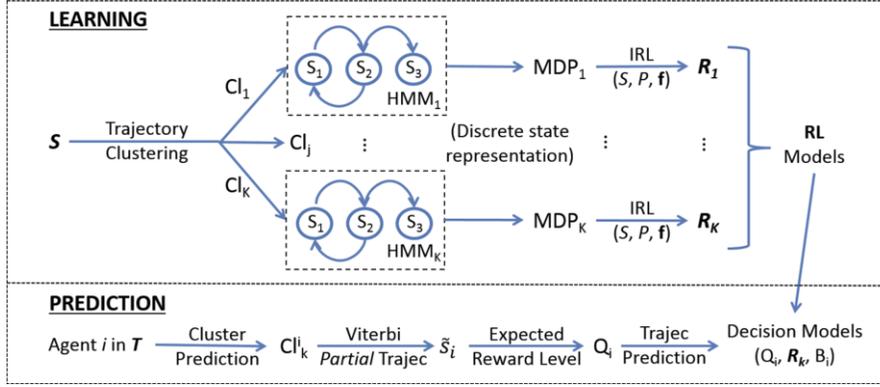


Figure 2: The proposed framework to model and predict the remaining trajectory of a test agent $i \in \mathcal{T}$ given its observed partial trajectory \tilde{s}_i , current budget B_i , and trajectories of the agents in the training set \mathcal{S} . “Trajec” stands for “trajectory”. S is the finite set of states, P is the matrix of state transition probabilities, and \mathbf{f} is the set of feature vectors of the states in S .

Table 1: Summary of notations used in the paper.

Notation	Description
$\mathcal{D}, \mathcal{S}, \mathcal{T}$	Total dataset, training set and test set, respectively
\mathcal{G}	Set of POIs, where $ \mathcal{G} = n$
u_{ij}, c_{ij}	Utility and cost of location $j \in \mathcal{G}$ for agent i
s_i, \tilde{s}_i, B_i	Trajectory, partial trajectory, and current budget of i
l_i	Trajectory (sequence) length of agent i
K	Number of clusters (agent types)
S	Finite set of states for each agent type ($ S = N$)
\mathbf{f}_s	Feature vector of each state $s \in S$
\mathbf{R}_k	State-reward matrix \forall agent type k ($1 \leq k \leq K$)
\mathbf{R}_a	Location-reward matrix for each location $a \in \mathcal{G}$
Q_i	Expected reward level (“personal goal”) of agent i

modeled as a Markov decision process (MDP). The utility of each action (i.e., location visit) can then be derived via the process of inverse reinforcement learning (IRL) using the agents’ observed actions (represented in the transition probability matrix P of the MDP). The final outcomes of IRL are the reward matrices \mathbf{R} .

Prediction. Given the observed partial trajectory and features of an agent i in the test set \mathcal{T} , we first predict i ’s type Cl_k^i using the trained classifier above. We then use the Viterbi algorithm [7] to find the most probable sequence of states \tilde{s}_i for the observed trajectory. We are then able to model i ’s goal Q_i (also called the “expected reward level”) and predict the next sequence of visits that can meet this goal within budget B_i . We finally propose two decision models that take into account the uncertainty of the utilities (represented by the matrix \mathbf{R}_k for each type k) and budget B_i .

In the following sections, we elaborate on each of the components of the proposed framework shown in Fig. 2.

5 Learning

5.1 Trajectory Clustering

For each agent $i \in \mathcal{S}$, let l_i ($1 \leq l_i \leq n$) be i ’s *sequence length*, which is the total number of locations visited by i . We denote the sequence of locations visited by i as $y^{(i)} = \{y_t^{(i)}\}_{t=1}^{l_i}$ and the sequence of corresponding timestamps as $\tau^{(i)} = \{\tau_t^{(i)}\}_{t=1}^{l_i}$. We define i ’s **trajectory** as $s^{(i)} = \{(y_t^{(i)}, \tau_t^{(i)})\}_{t=1}^{l_i}$. We are able to discretize $\tau^{(i)}$ into T segments, where Δ_τ is the duration of each segment. We can then derive a vector a_i of length T for each $s^{(i)}$, where each $a_{it} \in a_i$ ($1 \leq t \leq T$) indicates i ’s observed location at time t .

We can now cluster the agents based on the similarities a_i for all $i \in \mathcal{S}$ using, e.g., hierarchical clustering (because of its simplicity and effectiveness). In particular, we propose to use the agglomerative approach that clusters the vectors recursively from bottom up. To this end, we use the edit distance [6] to quantify the dissimilarity between a_i and a_j with the substitution cost being the distance between the pair of locations that differ. To select K , the hierarchical tree is “cut” at some height that splits \mathcal{S} into K clusters. The goodness of the clustering can then be quantified using, e.g., the silhouette coefficient. We choose K that best aligns with our domain knowledge and produces a good enough clustering coefficient.

5.2 Environment Modeling

We use hidden Markov models (HMMs) to model the environment the agents interact with as a finite set of states $S = \{S_1, S_2, \dots, S_N\}$. An HMM describes the relationship between an observed stochastic process and an unobserved (hidden) underlying process. The hidden process follows a Markov chain and the observations are conditionally independent given the sequence of hidden states. Let $\{Y_t\}_{t=1}^T$ and $\{X_t\}_{t=1}^T$ represent the observations and the corresponding hidden states, respectively. We denote $f(y_t | \Theta_{x_t}) = \Pr(Y_t = y_t; \Theta | X_t = x_t)$ as the (emission) density function of observation y_t parameterized over Θ given hidden state x_t . Each emission y_t is a tuple (y_k, τ_k) with the spatial component y_k being a discrete location drawn from \mathcal{G} and the temporal component τ_k being a continuous timestamp drawn from the Gaussian distribution $\mathcal{N}(\mu_k, \sigma_k)$ ($1 \leq k \leq N$).

An HMM with N states is completely specified by:

1. The finite set of hidden states $S = \{S_1, S_2, \dots, S_N\}$;
2. The state transition matrix $\mathbf{T} = \{t_{ij}\}$, where $t_{ij} = \Pr(X_t = S_j | X_{t-1} = S_i)$, $1 \leq i, j \leq N$;
3. The parameter vector Θ_i of the response (or emission) density function $f(y_t | \Theta_{x_t})$ for each S_i ; and
4. The vector of initial (state) probabilities $p = \{p_i\}$, where $p_i = \Pr(X_1 = S_i)$ and $\sum_{i=1}^N p_i = 1$.

Each hidden state of the HMM can be thought of as a *spatiotemporal cluster* of the visiting activities. Empirical observations confirm that nearby locations are much more likely to be visited sequentially in short periods of time, i.e., having “high” emission probabilities. We fit the HMMs using the trajectories $s^{(i)} \forall i \in \mathcal{S}$. A well-known method to estimate the parameters of an HMM is the Baum-Welch

algorithm [4]. For each HMM_{*j*} ($1 \leq j \leq K$), we select the optimal number of states N_j^* using the Bayesian Information Criterion (BIC) [8]. An important inference problem is that given a sequence of observations, find the most probable sequence of hidden states that produces it, which can be solved using the Viterbi algorithm [7].

5.3 Inverse Reinforcement Learning

5.3.1 Preliminaries.

Markov decision processes (MDPs) [5] provide an elegant framework to model sequential decisions in an environment represented as a finite state space S . At each state $s \in S$, the agent chooses an action $a \in A$. Upon which, the process transitions into the next state $s' \in S$ according to the probability $P_a(s, s') = \Pr(S_{t+1} = s' | S_t = s, a_t = a)$. The agent then receives a reward $R_a(s, s')$. The main concern of MDP is to find an optimal policy $\pi^* : S \mapsto A$ that maximizes the long-term *cumulative* reward $\sum_t R_{a_t}(s_t, s_{t+1})$.

Let $P_{\pi(s)}$ represent the transition probability matrix corresponding to the application of some policy π . A finite-horizon MDP is completely described by the tuple $(S, A, P_{\pi(s)}, R)$. The value function $V^\pi(s)$ of policy π at state s represents the expected cumulative reward from s . Thus, our goal is to find an optimal policy π^* such that $V^{\pi^*}(s)$ is maximized. It can be shown that there exists at least one optimal policy such that $V^{\pi^*}(s)$ is maximized for all $s \in S$ [24] that can be expressed as:

$$\pi^*(s) \in \arg \max_{a \in A} \sum_{s' \in S} P_a(s, s') [R(s, s') + \gamma V^{\pi^*}(s')]. \quad (1)$$

A fundamental property of the value function is, for any policy π and any state s :

$$V^\pi(s) = R_{\pi(s)}(s) + \sum_{s' \in S} P_{\pi(s)}(s, s') V^\pi(s'). \quad (2)$$

Eqn. (2) (called the Bellman equation) directly gives rise to efficient dynamic programming (DP) formulations to find a long-term optimal policy π^* called value iteration and policy iteration [5].

Inverse reinforcement learning (IRL) is the inverse problem to MDP, whose goal is to determine the reward function R that is being optimized given observations of the sequential decisions. Ng and Russell [21] originally propose LP formulations to solve the problem with constraints leading to the optimal observed policy. Abbeel and Ng [2] later propose a strategy of *matching feature expectations* between an observed policy and an agent's behaviors. The strategy is both necessary and sufficient to achieve the same performance as if the agent were in fact solving an MDP with reward function linear in the features of the states. Denote ξ_i a state-based trajectory (aka a "path"), \mathbf{f} the sequence of feature vectors of a path, and $\bar{\mathbf{f}} = \frac{1}{m} \sum_i \mathbf{f}_{\xi_i}$ the empirical expected feature count based on m trajectories. Matching feature expectations is described by:

$$\sum_{\xi_i} \Pr(\xi_i) \mathbf{f}_{\xi_i} = \bar{\mathbf{f}}. \quad (3)$$

We adopt the maximum entropy (MaxEnt) IRL algorithm [27] to learn the reward distribution of each state. MaxEnt IRL is an effective framework for modeling and understanding human activities, where the recovered reward function intuitively encodes an individual's set of preferences [12]. The notion of reward distribution comes from the fact that different people, even if classified into types, would still have different preferences (utilities) for the same thing. Such diversity in tastes can be best modeled as a probability distribution.

5.3.2 Maximum Entropy IRL (MaxEnt IRL).

Given a state-action sequence $\xi = \{(s, a)\}_i$, where $s_i \in S$ and $a_i \in A$, agent i is optimizing some function that linearly maps the features of each state $\mathbf{f}_{s_j} \in \mathbb{R}^k$ to a reward value that represents i 's utility of visiting that state. This function is parameterized by some weight vector θ and the reward of a trajectory is simply the sum of all the state rewards. The reward weights are applied to the path feature counts $\mathbf{f}_\xi = \sum_{s_j \in \xi} \mathbf{f}_{s_j}$ such that the reward of the trajectory is the weighted sum of the feature counts along the path:

$$R(\mathbf{f}_\xi) = \theta \cdot \mathbf{f}_\xi = \sum_{s_j \in \xi} \theta \cdot \mathbf{f}_{s_j}. \quad (4)$$

Since many distributions of paths may match the feature counts and any one distribution from among this set may exhibit a preference for some of the paths over others not implied by the path features. Such ambiguity is solved using the principle of *maximum entropy* by choosing the distribution that does not exhibit any additional preferences beyond matching feature expectations. The resulting distribution over the paths is parameterized by the weights θ :

$$\Pr(\xi_i | \theta) = \frac{1}{Z(\theta)} e^{\theta \cdot \mathbf{f}_{\xi_i}} = \frac{1}{Z(\theta)} e^{\sum_{s_j \in \xi_i} \theta \cdot \mathbf{f}_{s_j}}, \quad (5)$$

where $Z(\theta)$ is some *partition function* for the parameter weights. This distribution also provides a *stochastic policy* (i.e., a distribution over the actions at each state). Refer to [27] for more details.

We now build an MDP model (S, A, P, R) for each agent type, where S is the set of states of the corresponding HMM and A is the set \mathcal{G} of locations. We then need a set of state sequences in order to derive the transition matrix P and reward function R . To this end, we convert each trajectory into its most probable sequence of (hidden) states using the Viterbi algorithm [7]. P is then derived by sampling the observed state transitions and action taken at each state.

MaxEnt IRL additionally requires a set of features \mathbf{f}_s for each state $s \in S$. We use the spatiotemporal characteristics of each state as its features. Specifically, recall that each state S_i of the HMM is both a spatial cluster (i.e., what locations are likely to be visited) and a temporal cluster (described by the Gaussian mean μ_i). We use the tuple $(lo_i, la_i, \mu_i, \sigma_i)$ as the features \mathbf{f}_{S_i} of S_i , where lo_i and la_i are the "mean"⁴ longitude and latitude coordinates of S_i and μ_i and σ_i are the mean and standard deviation of the Gaussian emission, respectively. Such weighted sum of the coordinates are referred to as the "cluster centroids" of the states. Hence, each state S_i admits a unique cluster centroid C_i described by its (lo_i, la_i) .

Each run j of MaxEnt IRL produces a unique reward function $R_j : S_i \mapsto \mathbb{R}^+$, $\forall 1 \leq i \leq N$. In order to produce a *distribution* of reward for each state, we split the trajectories into subsets and run MaxEnt IRL on each subset to get a unique reward function. The probability of each reward value is the proportion of the subset in the original set. Towards this end, we split the trajectories into subsets of equal sequence lengths and run MaxEnt IRL on each of them.

We compute the distribution of reward for each location as follows. Let \mathbf{R}_s be a state-reward matrix. \mathbf{R}_s is of dimension $l \times N$, where N is the number of states and l is the maximum sequence length. For each state S_k ($1 \leq k \leq N$), let p_k of length $n = |\mathcal{G}|$ be the vector of multinomial emission probabilities of the HMM. Let Π be the multinomial emission matrix of dimension $N \times n$ whose row vectors are p_k . We compute the location-reward matrix \mathbf{R}_a as:

$$\mathbf{R}_a = \mathbf{R}_s \times \Pi. \quad (6)$$

⁴ Precisely, lo_i and la_i are the sum of the coordinates of the locations weighted by the multinomial emission probabilities at S_i .

We assume that the stochastic reward $R(a)$ of each location a follows a Gaussian distribution, whose mean and variance can be derived from the corresponding column vector of \mathbf{R}_a .

6 Prediction

In this paper, we present two decision models to the problem of trajectory prediction: Adaptive MDP (AMDP) and Value Ratio (VR). The former follows the long-term optimal policy of an MDP and the latter uses myopic greedy heuristics to make decisions.

6.1 Adaptive MDP

Empirical evidence shows that the sequence lengths of the trajectories typically follow normal distributions [16, 18, 15]. We take advantage of this to introduce stochasticity of reward and policy into our model by splitting the training set into subsets of the same sequence lengths. For each subset, we learn a unique reward/policy function. In the end, we come up with a reward/policy matrix, where for each matrix, the columns are the states and the rows are the sequence lengths whose probability distribution follows that of the sequence lengths.

With the above setup, we obtain the following matrices from the training set for each agent type:

1. \mathbf{R}_s (or \mathbf{R}): each entry is the reward (column) of each state that corresponds to each sequence length (row);
2. \mathbf{V} ($l \times N$): each entry is the value (column) of each state that corresponds to each sequence length (row);
3. Optimal policy matrix Π^* ($l \times N$): each entry is an optimal action $a \in A$ at each state (column) that corresponds to each sequence length (row).

From \mathbf{R} , we are able to derive the Gaussian distribution of reward $R(s)$ at each state $s \in S$ using the probability distribution of the sequence length (i.e., the rows).

An important consideration in our model is the agent’s **expected reward level**. This comes about from the observation that an agent may finish its trajectory even when there is sufficient budget to go on. Such behavior may come from an intrinsic expected reward level, such as a “personal goal”, having been met. Once such goal is met, the agent would just be happy to finish there and then and not go on to maximize the cumulative reward any further. In order to model such a personal goal, we make use of the value function. From Eqn. (2), the value function at state s is sum of the immediate reward $R_{\pi(s)}(s)$ and the future expected reward. We use this future expected reward to model agent i ’s expected reward level Q_i :

$$Q_i = V^\pi(s) - R_{\pi(s)}(s). \quad (7)$$

Since both $V^\pi(s)$ and $R_{\pi(s)}(s)$ are given (by \mathbf{V} and $R(s)$, respectively), we can derive Q_i for each agent i knowing its current state s and the current sequence length k . Furthermore, the optimal policy matrix Π^* is *stochastic* because, given a state s , each column vector of policies $\Pi^*[:, s]$ is distributed according to the Gaussian distribution of the sequence length. Algorithm 1 describes the proposed Adaptive⁵ MDP decision model for trajectory prediction.

Algorithm 1 follows the long-term optimal policy of an MDP because it makes use of the optimal (stochastic) policy function to make decision at each step. The policy function is long-term optimal as a result of solving the Bellman equation (2).

⁵ “Adaptive” is used to mean that the algorithm is adapted to stochastic rewards/policies and the budget constraint.

Algorithm 1 Adaptive MDP decision model for agent i

- 1: Given agent i ’s partial trajectory $\tilde{s}_i = \{(s, a)\}_i$ of current length k and i ’s current budget $B_i > 0$
 - 2: Let $s = \tilde{s}_i[k]$ be the current state
 - 3: Sample reward $R(s)$ from Gaussian distribution
 - 4: Retrieve current state’s value $\mathbf{V}[k, s]$
 - 5: Let $Q_i = \mathbf{V}[k, s] - R(s)$ be i ’s expected reward level
 - 6: Initialize i ’s future cumulative reward $U_i \leftarrow 0$
 - 7: Let $\hat{s}_i \leftarrow \emptyset$ be the predicted sequence
 - 8: **while** $U_i < Q_i$ and $B_i > 0$ **do**
 - 9: Sample an action a from policy $\Pi^*[k : l, s]$
 - 10: **while** $a \in \tilde{s}_i$ { a has been visited} **do**
 - 11: Repeat Step 9
 - 12: **end while**
 - 13: Sample next state s' from $P_a(s, s')$
 - 14: Update $k \leftarrow k + 1$; $s \leftarrow s'$
 - 15: Update $\hat{s}_i \leftarrow \hat{s}_i \cup (s, a)$; $\tilde{s}_i \leftarrow \tilde{s}_i \cup \hat{s}_i$
 - 16: Sample reward $R(s)$ from Gaussian distribution
 - 17: Let t_a be the travel time from current location to a
 - 18: Let Δ_a be the minimum duration to be spent at a
 - 19: Update $U_i \leftarrow U_i + R(s)$; $B_i \leftarrow B_i - (t_a + \Delta_a)$
 - 20: **end while**
 - 21: Return the sequence of actions in \hat{s}_i
-

6.2 Value Ratio

At each time step, the agent samples a random reward value r_j from the Gaussian distribution $R(a_j)$ of each of the *remaining* locations a_j . Given its current location, the agent heuristically maps itself to the nearest *cluster centroid* (refer to Sect. 5.3.2) as a “point of reference” and derives the distances d_j from the cluster centroid to each of the remaining locations. The agent then chooses to visit the location j^* that has the largest ratio r_j/d_j (i.e., the ratio of the immediate reward to its cost) and repeats until its budget runs out or there is no unvisited location left. This is the well-known best “bang-for-the-buck” greedy heuristic [3]. Algorithm 2 describes the model.

Algorithm 2 Value Ratio decision model for agent i

- 1: Given agent i ’s current location a_i , its current set of *unvisited* locations $\mathcal{G}_i \subseteq \mathcal{G}$ and the current budget $B_i > 0$
 - 2: Let $\hat{s}_i \leftarrow \emptyset$ be the predicted sequence of visits
 - 3: **while** $|\mathcal{G}_i| > 0$ and $B_i > 0$ **do**
 - 4: Sample reward r_j from Gaussian distribution for each $a_j \in \mathcal{G}_i$
 - 5: Let $C_{k^*} = \arg \min_k \text{distance}(a_i, C_k)$ ($1 \leq k \leq N$)
 - 6: Let $d_j = \text{distance}(a_j, C_{k^*})$, $\forall a_j \in \mathcal{G}_i$
 - 7: Select a_{j^*} where $j^* = \arg \max_j r_j/d_j$, $\forall a_j \in \mathcal{G}_i$
 - 8: Update $\hat{s}_i \leftarrow \hat{s}_i \cup \{a_{j^*}\}$; $\mathcal{G}_i \leftarrow \mathcal{G}_i \setminus \{a_{j^*}\}$
 - 9: Let t_{j^*} be the travel time from a_i to a_{j^*}
 - 10: Let Δ_{j^*} be the minimum duration to be spent at a_{j^*}
 - 11: Update $B_i \leftarrow B_i - (t_{j^*} + \Delta_{j^*})$; $a_i \leftarrow a_{j^*}$
 - 12: **end while**
 - 13: Return \hat{s}_i
-

7 Experiments

7.1 Dataset

We collaborated with a large theme park operator in a major Asian city to conduct experiments and collect demographic and behavioral

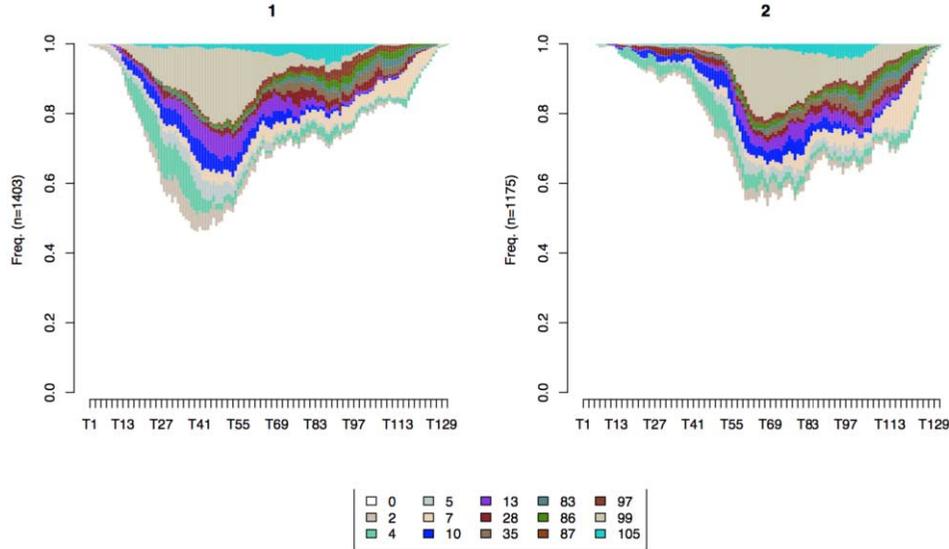


Figure 3: Visualization of the two clusters (agent types) of the training data in the experiments. Horizontal axes represent the timeline in discrete intervals of 5 minutes from 9 a.m. to 7 p.m. Vertical axes represent the probability of the visitors of each type being at each of the 14 attractions (or at some unknown location “0”). Attractions are represented by their color codes whose legend is shown at the bottom.

data from their visitors from January to April, 2014. The dataset contains the visitors’ trajectories tracked using RFID devices. In the experiments, visitors pay upfront a fixed amount in order to redeem up to 14 participating attractions (locations). Visitors can only redeem the attractions during the specified 10-hour period from 9 a.m. to 7 p.m. on a chosen day. Each attraction can only be visited once.

Our dataset \mathcal{D} contains trajectories of 3,867 unique and independent visitors together with their demographic features. The empirical distribution of the sequence length of these trajectories follows a typical bell-shaped characteristic of a Gaussian distribution.

7.2 Trajectory Clustering

We perform cross-validations⁶ on \mathcal{D} . For each fold, the training set \mathcal{S} is used for trajectory clustering and decision modeling. Our hierarchical clustering results in $K = 2$ clusters using the interval $\Delta_\tau = 5$ minutes (refer to Sect. 5.1) for all the agents. The value of K was chosen based on inspection of the hierarchical tree and empirical goodness of clustering via the silhouette coefficient (partitions of comparable sizes and good in-group cohesiveness).

Fig. 3 visualizes the 2 clusters using training data of one of the random folds. The horizontal axes represent the discretized timeline (by Δ_τ) from 9 a.m. to 7 p.m. for each cluster and the vertical axis represents the probability for each agent of each cluster to be at any one of the 14 attractions at any interval. (Note that even after 7 p.m., some activities can still be recorded in the park.) The attractions are identified by their numbers whose color codes are shown in the legend at the bottom of the figure. We denote “0” (white color) when we do not know for sure the location of an agent during a given time interval (i.e., he was not observed at any known attraction during the interval). We can see that, most of the time, visitors hang out in the park without checking into any specific attractions.

The trajectory clustering reveals that the main differences between the two agent types are their temporal behaviors. That is, agent type

1 tends to arrive earlier and has their peak of visiting activities earlier in the day (around 12–1 p.m.), and then (their visit frequency) sharply drops off. Whereas, agent type 2 tends to arrive much later and reaches their peak later (at round 3–4 p.m.), and then gradually declines. If budget is defined as the duration from the time of entry until the closing time (7 p.m.), then agent type 1 has, on average, 114 minutes more than agent type 2. As a result, we call agent type 1 the “early birds” and agent type 2 the “latecomers”. The two clusters have roughly comparable sizes with cluster 1 being 54.42% and cluster 2 being 45.58% of the set training \mathcal{S} .

7.3 Evaluation

For each cluster in \mathcal{S} , we learn the matrices \mathbf{R} , \mathbf{V} , and Π^* . The test set \mathcal{T} is used to validate the predicted trajectories. For each agent $i \in \mathcal{T}$, let l_i be i ’s final sequence length. We first predict i ’s type using its demographic features and first timestamp via a multinomial logistic model. Given i ’s partial trajectory of length k , we predict i ’s remaining trajectory while varying $k \in [2, l_i - 1]$. Let s_i^* and \hat{s}_i be i ’s actual and predicted remaining trajectory, respectively. We use the Levenshtein edit distance [6] to quantify the similarity between s_i^* and \hat{s}_i . Each match receives a fixed positive score and each mismatch incurs a negative penalty proportional to the distance (in kilometers) between the two locations.

The following baseline models are used for evaluation:

1. **HMM.** At each time step, predict agent i ’s current state s , generate an *unvisited* location based on the state’s multinomial probabilities p_s and repeat until B_i runs out. This is based on [20].
2. **Nearest neighbor.** At each time step, agent i redeems a remaining location that is nearest to its current location and repeats until B_i runs out.
3. **Random.** At each time step, i redeems a random unvisited location and repeats until B_i runs out.

⁶ Precisely, we performed 3-fold cross-validations to ensure a large enough training/test partition per fold.

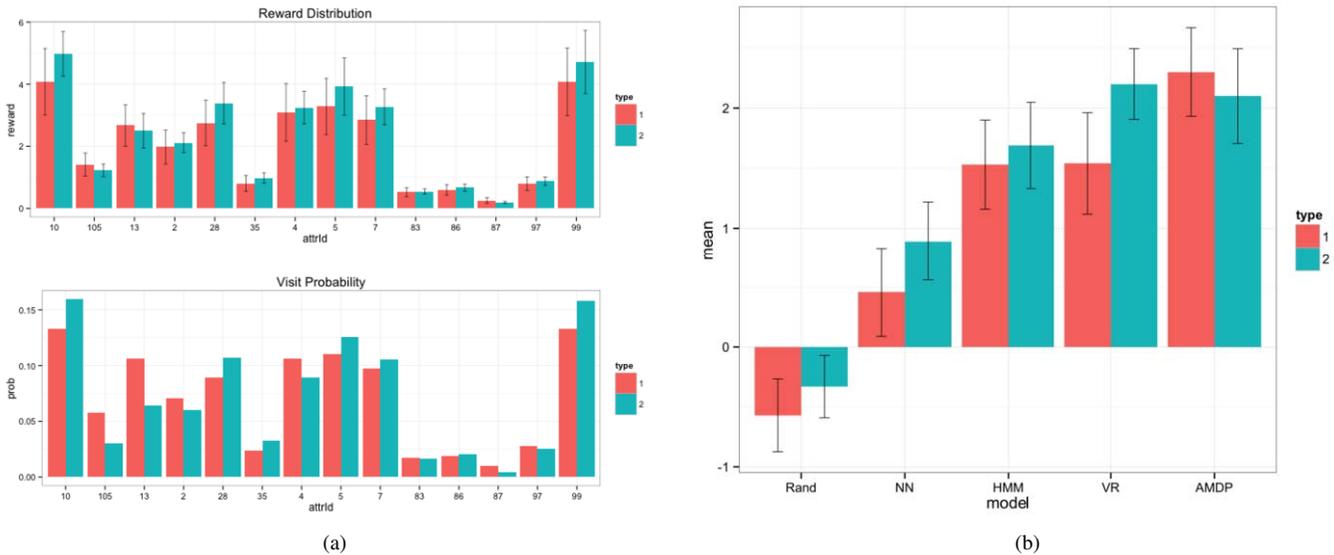


Figure 4: (a) Comparison between the estimated reward distribution for each attraction (“attrId”) (top panel) and the empirical visit probability for each attraction (bottom panel). (b) Similarity measures between the actual and predicted trajectories across different models: Random (“Rand”), Nearest Neighbor (“NN”), HMM, Value Ratio (“VR”) and Adaptive MDP (“AMDP”).

7.4 Results

Our experimental results are summarized in Fig. 4. In Fig. 4a, the mean reward per attraction learned from IRL and Eqn. (6) is plotted together with its 95% confidence interval (top panel). The figure shows that the mean rewards, in general, faithfully reflect their respective empirical probabilities of attraction visit for both agent types (i.e., their preferences – in the bottom panel).

It is noteworthy to observe in Fig. 4a that agent type 2 has, for the most part, higher (absolute) *immediate* reward per attraction (top panel) than agent type 1. This consequentially differentiates the underlying decision processes employed by the two agent types. Fig. 4b shows the distributions of the similarity measures (means and variances – represented by 95% confidence bars) across the models. Each distribution is computed from the cross-validation while varying the observed partial trajectory length $k \in [2, l_i - 1]$. A higher mean similarity implies a more accurate prediction, on average. These distributions (in Fig. 4b) are empirically verified to be Gaussian.

For agent type 1, Fig. 4b shows that the Adaptive MDP model has the most accurate prediction, on average. The Value Ratio and HMM model both have about the same second best average prediction score. The Random baseline model has the least accurate average prediction, which is quite reasonable, followed by the Nearest Neighbor model. For agent type 2, the figure shows that the Adaptive MDP model performs marginally worse than the Value Ratio model, even though it still fares much better than the other baselines. In other words, the Value Ratio model makes the most accurate prediction, on average, for this group of agents. This is a remarkable result that warrants further discussion.

7.5 Discussion

From trajectory clustering, we have discovered that agent type 1 are the early birds and agent type 2 are the latecomers. From the perspective of modeling, agent type 1 has a much larger budget (by 114 minutes, on average) than agent type 2. Larger budget means more

flexibility, more foresight and better long-term planning, which is what the Adaptive MDP model reflects: it embodies the long-term optimal policy of the corresponding reinforcement learning model. This indeed performs better than other short-sighted baselines.

On the other hand, a smaller budget, which agent type 2 has, translates into less flexibility and less time for careful planning, which ultimately results in more myopic and suboptimal decisions (i.e., resorting to greedy strategies). This is reflected in the experimental results, where the greedy and myopic Value Ratio model performs the best for agent type 2 (even though just marginally better than the Adaptive MDP). This myopic decision-making corroborates with the observations in Fig. 4a, where most of agent type 2’s immediate rewards are larger (in absolute terms) than agent type 1’s such that it sees less values in *delayed* (future) rewards and finds more incentives to act greedily [24]. This is also evidenced in Fig. 1, where type 2 has a much stronger tendency to visit attractions that are nearby to one another (i.e., maximizing the value ratio) than type 1.

8 Conclusion

In this paper, we address the problem of trajectory prediction using reinforcement learning to model the agent’s sequential decisions. By doing so, we have discovered from real-world trajectories how people make decisions: they make more optimal decisions when given enough time to do so. This is perhaps not surprising in retrospect, because it is reasonable that foresighted decisions and careful plans need time to coordinate, while myopic ones do not (as only the immediate rewards are considered). On the other hand, this also validates our framework’s ability to model real-world behaviors by finding out what makes reasonable sense in real life.

Our main shortcoming here is the simplistic handling of the budget constraint. We would like to see if handling it in more sophisticated ways would improve predictions. For example, for foresighted agents, we would like to experiment with decision models other than MDP in our future work. One of which is the adaptive stochastic knapsack [13], which is similar to a traditional knapsack model ex-

cept for the sequential decisions and stochastic reward of each item. Another shortcoming of this work is the simplistic Value Ratio model for myopic decision-making (type 2), which yields just slightly better prediction than the Adaptive MDP for agent type 2. Hence, for myopic agents, a more sophisticated decision model may be desirable to better model and predict their behaviors. One of such model for sequential decisions has been proposed in the operations research literature [23]. This is also worth investigating in the future work.

ACKNOWLEDGEMENTS

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Program Office, Media Development Authority, as well as its Corp Lab @ University scheme.

Siyuan Liu is additionally supported by the Basic Research Program of Shenzhen: JCYJ20140610152828686 and the Natural Science Foundation of China: 61572488.

REFERENCES

- [1] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y. Ng, ‘An application of reinforcement learning to aerobatic helicopter flight’, in *Advances in Neural Information Processing Systems*, eds., B. Scholkopf, J. Platt, and T. Hoffman, volume 19, Cambridge, MA, USA, (2007). MIT Press.
- [2] Pieter Abbeel and Andrew Y Ng, ‘Apprenticeship learning via inverse reinforcement learning’, in *Proceedings of the Twenty-first International Conference on Machine Learning*, p. 1. ACM, (2004).
- [3] Maria-Florina Balcan, Amit Daniely, Ruta Mehta, Ruth Urner, and Vijay V Vazirani, ‘Learning economic parameters from revealed preferences’, in *Web and Internet Economics*, 338–353, Springer, (2014).
- [4] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss, ‘A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains’, *The Annals of Mathematical Statistics*, **41**(1), 164–171, (1970).
- [5] R. Bellman, ‘A Markovian decision process’, *Journal of Mathematics and Mechanics*, **6**(4), 679–684, (April 1957).
- [6] Paul E Black, ‘Levenshtein distance’, *Algorithms and Theory of Computation Handbook*, (1999).
- [7] G David Forney Jr, ‘The Viterbi algorithm’, *Proceedings of the IEEE*, **61**(3), 268–278, (1973).
- [8] Chris Fraley and Adrian E Raftery, ‘Model-based clustering, discriminant analysis, and density estimation’, *Journal of the American Statistical Association*, **97**(458), 611–631, (2002).
- [9] Sébastien Gams, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez, ‘Next place prediction using mobility Markov chains’, in *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, p. 3. ACM, (2012).
- [10] Gerd Gigerenzer, Reinhard Selten, et al., ‘Rethinking rationality’, *Bounded rationality: The adaptive toolbox*, 1–12, (2001).
- [11] Valerie Guralnik and Karen Zita Haigh, ‘Learning models of human behaviour with sequential patterns’, in *Proceedings of the AAAI-02 Workshop on Automation as Caregiver*, pp. 24–30, (2002).
- [12] De-An Huang, Amir-massoud Farahmand, Kris M Kitani, and J Andrew Bagnell, ‘Approximate maxent inverse optimal control and its application for mental simulation of human interactions’, in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, (2015).
- [13] Taylan Ilhan, Seyed MR Iravani, and Mark S Daskin, ‘The adaptive knapsack problem with stochastic rewards’, *Operations Research*, **59**(1), 242–248, (2011).
- [14] John Krumm and Eric Horvitz, ‘Predestination: Inferring destinations from partial trajectories’, in *UbiComp 2006: Ubiquitous Computing*, 243–260, Springer, (2006).
- [15] Truc Viet Le, Siyuan Liu, and Hoong Chuin Lau, ‘Reinforcement learning framework for modeling spatial sequential decisions under uncertainty’, in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp. 1449–1450. International Foundation for Autonomous Agents and Multiagent Systems, (2016).
- [16] Truc Viet Le, Siyuan Liu, Hoong Chuin Lau, and Ramayya Krishnan, ‘A quantitative analysis of decision process in social groups using human trajectories’, in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 1425–1426. International Foundation for Autonomous Agents and Multiagent Systems, (2014).
- [17] Truc Viet Le, Siyuan Liu, Hoong Chuin Lau, and Ramayya Krishnan, ‘Predicting bundles of spatial locations from learning revealed preference data’, in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1121–1129. International Foundation for Autonomous Agents and Multiagent Systems, (2015).
- [18] Siyuan Liu, Qiang Qu, and Shuhui Wang, ‘Rationality analytics from trajectories’, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, **10**(1), 10, (2015).
- [19] Haiping Ma, Huanhuan Cao, Qiang Yang, Enhong Chen, and Jilei Tian, ‘A habit mining approach for discovering similar mobile users’, in *Proceedings of the 21st International Conference on World Wide Web*, pp. 231–240. ACM, (2012).
- [20] Wesley Mathew, Ruben Raposo, and Bruno Martins, ‘Predicting future locations with hidden Markov models’, in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 911–918. ACM, (2012).
- [21] Andrew Y Ng and Stuart Russell, ‘Algorithms for inverse reinforcement learning’, in *Proc. 17th International Conf. on Machine Learning*, (2000).
- [22] Stuart Russell, ‘Learning agents for uncertain environments (extended abstract)’, in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pp. 101–103, New York, NY, USA, (1998). ACM.
- [23] Matthew J Sobel and Wei Wei, ‘Myopic solutions of homogeneous sequential decision processes’, *Operations Research*, **58**(4-part-2), 1235–1246, (2010).
- [24] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, USA, 1998.
- [25] Shao Zhifei and Er Meng Joo, ‘A review of inverse reinforcement learning theory and recent advances’, *International Journal of Intelligent Computing and Cybernetics*, **5**(3), 293–311, (June 2012).
- [26] Brian D. Ziebart, Anind K. Dey, and J. Andrew Bagnell, ‘Probabilistic pointing target prediction via inverse optimal control’, in *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*, pp. 1–10, New York, NY, USA, (February 2012). ACM.
- [27] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey, ‘Maximum entropy inverse reinforcement learning.’, in *AAAI*, pp. 1433–1438, (2008).
- [28] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey, ‘Human behavior modeling with maximum entropy inverse optimal control.’, in *AAAI Spring Symposium: Human Behavior Modeling*, p. 92, (2009).

Person Re-Identification via Multiple Coarse-to-Fine Deep Metrics

Mingfu Xiong^{1,3} and Jun Chen^{1,2} and Zheng Wang^{1,3}
and Zhongyuan Wang^{2,3} and Ruimin Hu^{1,2,3} and Chao Liang^{1,2} and Daming Shi⁴

Abstract. Person re-identification, aiming to identify images of the same person from various cameras views in different places, has attracted a lot of research interests in the field of artificial intelligence and multimedia. As one of its popular research directions, the metric learning method plays an important role for seeking a proper metric space to generate accurate feature comparison. However, the existing metric learning methods mainly aim to learn an optimal distance metric function through a single metric, making them difficult to consider multiple similar relationships between the samples. To solve this problem, this paper proposes a coarse-to-fine deep metric learning method equipped with multiple different Stacked Auto-Encoder (SAE) networks and classification networks. In the perspective of the human's visual mechanism, the multiple different levels of deep neural networks simulate the information processing of the brain's visual system, which employs different patterns to recognize the character of objects. In addition, a weighted assignment mechanism is presented to handle the different measure manners for final recognition accuracy. The experimental results conducted on two public datasets, i.e., VIPeR and CUHK have shown the prospective performance of the proposed method.

1 Introduction

Person re-identification aims to judge whether two persons which come from different cameras views belong to the same person. Owing to its significance in tracking the escape route of suspects and daily life, it has been widely used in the criminal investigation and artificial intelligence [2]. Over the past decade, a large number of person re-identification methods have been proposed in the literatures [20, 1, 25, 23, 26, 19, 22, 16] and most of them have achieved satisfying performance. However, it is still a challenging problem because of various surveillance conditions, such as, view switching, lighting variations and image scaling (see Figure 1). Previous research on person re-identification can be generally classified into two categories: feature representation [25, 23, 13] and metric learning [26, 20, 9]. Since lighting and view changes can cause significant appearance variations, designing a set of discriminative and robust features is still a challenging problem [26, 21]. In order to boost the performance of person re-identification, increasing number of researches are devoted



Figure 1. The examples of aspects changes caused by different views, lighting conditions, scaling variations from public datasets CUHK [14], VIPeR [8], respectively. Each column shows two images of the same person from two different cameras.

ed to learn a proper distance function to compare two person image features [21, 11].

Most of the existing work focus on either feature representation or metric learning step, lacking a global consideration of above two steps. It is crucial to build an automatic connection among these components in the training process for the overall system performance. More recently, deep learning, which is based on an end-to-end network, has been presented to solve the problem in a unified framework. It has attracted a lot of research interests for its superb performance in person re-identification and other visual tasks [11, 10].

Generally speaking, deep learning aims to learn features and metrics in a unified hierarchical framework directly from raw data. It has also been used in metric learning [11, 24, 1]. Unlike most previous metric learning methods which usually seek a linear distance to project samples into another linear space, the deep metric learning methods try to compute the similarities of samples via multiple layer nonlinear transformations. However, most of them just try to seek a simplex manner to measure the similarities of the persons. Such a simplicity of the metric manner may cause the problem that other similarities relationship of samples cannot be well exploited. Figure 2 is a particular example, where the persons in the two pictures are similar in shape contour and clothes, but they are not the same one in fact. Therefore, the single metric learning methods may lose helpful discriminative information for similarity comparison.

To relieve the problems with these limitations, we propose a method with multiple coarse-to-fine SAE models (SAE networks and classification networks) for deep metric learning. In our algorithm, there exist several SAEs neural networks with different hidden layers for multi-scale metric learning and the similarities of multiple

¹ State Key Laboratory of Software Engineering, Wuhan University, China. email: {xmf2013, chenj, wangzwhu, hrm, cliang}@whu.edu.cn, wzy_hope@163.com

² Collaborative Innovation Center of Geospatial Technology, China.

³ National Engineering Research Center for Multimedia Software, Computer School of Wuhan University, China.

⁴ School of Science and Technology, Middlesex University London, London NW4 4BT, United Kingdom. email: d.shi@mdx.ac.uk



Figure 2. Examples of dissimilar pairs. From this figure, we can see that the persons in the same column are similar in color and contour. In fact, they are not the same person. So the important information that judges whether the samples belong to the same object may be lost via a single metric manner.

levels for person image pairs are obtained via different deep neural networks in a coarse-to-fine manner. Generally speaking, we judge two persons which are the same one or not just via the physical characteristic at first glance. Then the facial features and clothes can be compared. At last, more details will be observed for final validation. This process is the information handling of our visual system. In our work, there are different deep neural networks for metric learning and it includes many neural nodes for each network which simulates the neuron of brain. There are fewer nodes in shallow neural network and vice versa. These architectures are similar to the structure of the brain in the view of bionics. In this way, we can simulate the information processing of the brain’s visual system, which employs multiple different levels to recognize the character of objects. Besides, a weighted assignment mechanism is presented to handle these results which are from different SAEs networks.

The contribution of this paper can be summarized into two aspects:

Firstly, we propose a framework of multiple different SAEs networks and classification networks for metric learning to measure the similarities of the samples from coarse-to-fine.

Secondly, a weighted assignment mechanism is presented for integrating the results that come from previous different deep neural networks. The information processing mechanism of brain is simulated via this coarse-to-fine manner. Experimental results validate the effectiveness on two public person re-identification datasets.

2 Our Approach

2.1 Preliminaries

2.1.1 Person Re-identification Problem

As mentioned above, the purpose of person re-identification is to match the pedestrians observed in non-overlapping cameras via various visual methods. In other way, this problem can also be seen as a binary classification problem. For the convenience of following discussion, in our work, we consider a pair of cameras which are denoted as C_a and C_b , respectively. The persons in each camera are expressed as $\{p_a = p_a^1, p_a^2, \dots, p_a^n\}$ and $\{p_b = p_b^1, p_b^2, \dots, p_b^m\}$. The n and m denote the numbers of the person in each camera view. Let the label $y = 1$ if two pedestrian images (p_a^i, p_b^j) are matched, and $y=0$, otherwise. So a pair of person image is the object that we should consider in this paper. P_{ab}^l is the combination of two persons that from different cameras views, respectively.

2.1.2 The Basic Auto-Encoders

We recall the basic principles of the auto-encoder models, e.g. [3]. The classical auto-encoder tries to learn a function $h_{W,b}(x) \approx x$. In other words, the algorithm is trying to learn an approximation to the identify function, so as to the output \hat{x} that is similar to the input x . It is divided into two processes, that is “encoding” and “decoding”. In the former, it is using a deterministic function of $h = f_\theta = \sigma(Wx + b)$ with parameters $\theta = \{W, b\}$. And in the process of decoding, it is used to reconstruct the input by a reverse mapping of $f: h' = f_{\theta'} = \sigma(W'h + b')$ with $\theta' = \{W', b'\}$. The two parameter sets are usually constrained to be of the form $W' = W^T$, using the same weights for encoding the input and the latent representation y_i . A common method to train the model is the famous Back-Propagation Algorithm [18]. The cost function is described as below. For example, we just have a set of training samples: $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$. And the cost function for one training case is described as following.

$$J(W, b; x, y) = \frac{1}{2} \|h_{W,b}(x) - y\|^2 \quad (1)$$

In addition, cost function for the whole training set is described as formula (2).

$$J(W, b) = \frac{1}{m} \sum_{i=1}^m J(W, b; x, y) + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ij}^{(l)})^2 \quad (2)$$

In order to train the model, we just need to minimize $J(W, b)$. The process of training is not belong to this range.

2.2 Multiple Coarse-to-Fine Deep Metric Learning

The architecture of the multi-scale learning method is shown in Figure 3. It includes four layers to get the coarse-to-fine deep metric learning for person re-identification. The first layer is the monitored person images that come from two different camera views. We randomly combine the two person images together to form the original input for the second layer. Then the pretreatment is executed via subtracting the mean values and normalization for each sample pair. The images are transformed into gray images and the input of the SAE networks is formed. Then a softmax classifier is followed by each of the stacked auto-encoder to get a classification result. At last, we have utilized a weighted assignment mechanism to handle the classification results obtained from the former layer. And the multiple deep metric learning framework includes two networks: the SAEs network and classification network. The detail of each layer is described as below.

There are several different SAE models for metric learning in our algorithm. For each auto-encoder network, it has three layers: the input layer, hidden layer and the output layer. In many previous work, the auto-encoder networks were used for feature representation [17]. In this work, we have used it for metric learning. In details, each of the SAE network is following by a softmax classifier (See Figure 3). Each of them is trained via the back-propagation algorithm.

From Figure 3, we can see that the input of auto-encoder networks is the person image pairs, which is reprocessed before being input into the deep neural networks. The network parameters are trained from the first hidden layer. And the output of the first hidden layer is calculated via the parameters that were trained before. The output for the next hidden layers is counted through the same way. After that, the last hidden layer is followed by a softmax classifier. The output of the last hidden layer is used to train parameters for the

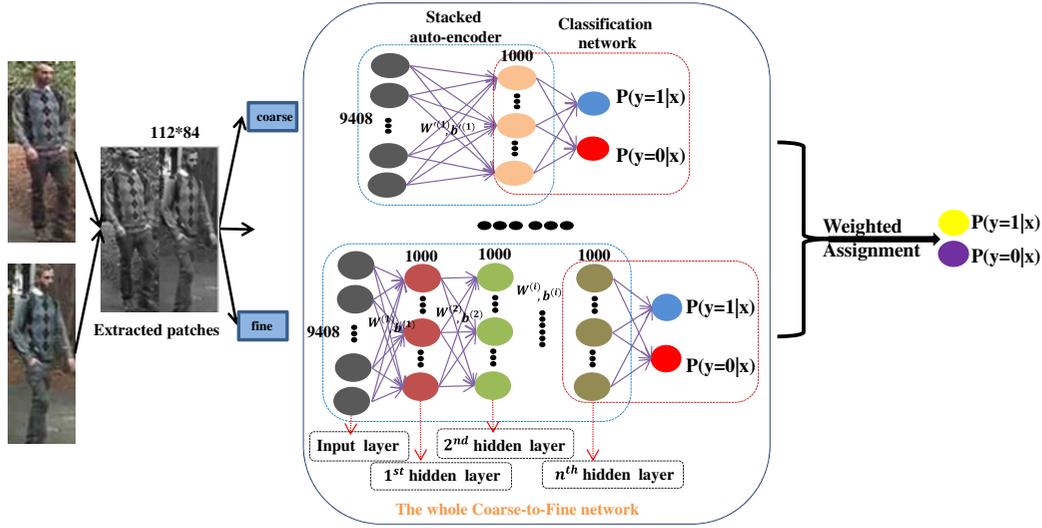


Figure 3. The Multiple Coarse-to-Fine Deep Metric Learning Framework. From left to right, there are four layers structured for last classification. For the first layer, we randomly select the two person images which come from different camera views. And the second, the combined image pairs are obtained from the previous layer that transformed into gray images. Then they are subtracted to the mean values and normalized into [0,1]. The coarse-to-fine SAEs structure is following in the third layer. This layer includes several SAEs which equipped with a softmax classifier. In other word, The whole network is composed of SAEs networks and classification networks. The parameter $\theta=\{W, b\}$ (W is the weight, and b is bias.) is to be trained. The output of the softmax classifier is the probability that the sample pair belongs to a certain class. And the weighted assignment mechanism is used for handle the classification results for the last layer.

softmax classifier. The cost function for the softmax classifier is described as formula (3). For example, considering the training set is $\{(x^{(1)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$. The m denotes the numbers of samples and $x^{(i)}$ represents the feature that is the output of the last hidden layer in this work. The y^i is the classification label for each sample.

$$J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}} \right] \quad (3)$$

In order to train the model, we just need to calculate the $J(\theta)$. The gradient decent method is used in the algorithm. And k is 2 for the person re-identification problem. The last result is classified into two classes.

In our model, there are several kinds of SAE networks and each of them has different configurations. So we can capture different levels metric results for each sample pair. A coarse metric learning is implemented via the shallow network and vice versa. So the coarse-to-fine metric manner is formed in this way. In our work, a pair of person images is generated to form the input of the SAE networks. But the final output is the probability that a person belongs to a class. Therefore, we can get multiple different results. These classified results are the sources that we can obtain the final recognition accuracy. And then handling these classified results is described as the following section.

2.3 Joint Learning for Weighted Assignment

As mentioned above, there are several kinds of SAE models for the persons binary classification. The output of each softmax classifier is a probability that the person pair belongs to a certain class. And the probability values that we can obtain are diversified. How to handle these results is remained to settle. In this work, we have utilized the weighted assignment mechanism to solve this problem. In our work, there are several SAE models for metric learning and the multiple

similarities are generated via these networks. As the characteristic capability of each metric is different. So the weighted assignment mechanism is presented to get the final result. And the process of jointing is described in Figure 4.

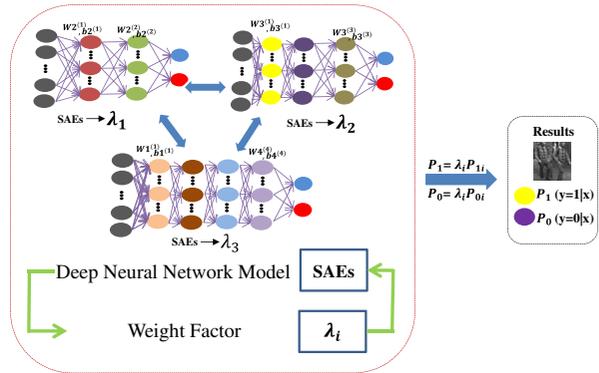


Figure 4. The process of the weights assignment mechanism. In our work, there are three kinds of SAE models for pedestrians classification. The weight factor λ_i is made via joining the deep networks. This process includes three parts: the first one is the process of these SAE models joint learning for three kinds of networks. The parameter $\theta=\{W, b\}$ should be trained. Then the weight factor is assigned for each SAE model. At last, the final result is gotten via the operation of weight assignment.

In details, the output of the softmax classifier is two classes represented by the probabilities. From Figure 3, we assume that there are three kinds of SAE models to classify for the person pair. The notations $P(y = 1|x)$ and $P(y = 0|x)$ denote the probabilities that the sample pair x belongs to the certain class. If the person pair is matched, the label $y = 1$, and $y = 0$ otherwise. Generally speaking, the two samples are similar, the other aspects of them are similar

too. We try to consider multiple aspects of the samples for judging whether they belong to the same object. For the person images pair (p_a^i, p_b^i) , the probability they matched is P_{ab}^i (P_{ab}^i is from the i -th SAE model. See Figure 3.). The weights assignment mechanism for each probability matrix distribution is represented as formula (4). After that, we reset the probabilities to get the last recognition result.

Algorithm 1 Weighted Assignment For Similar Probability

Input: N labeled a set of training samples (P_{ab}^i, y_i) where P_{ab}^i is the combination of two pedestrian images and $y_i \in \{1, -1\}$ denotes whether the two person belong to the same one.

A distribution over all the training samples: $D_1(i) = 1/N$ for $i=1, \dots, N$.

for $t=1, \dots, K$:

- Find the best localized feature λ_t for the current distribution D_t .

- Calculate the edge γ_t

$$\gamma_t = \sum_{i=1}^N D_t(i) h(x_i) y_i$$

- If $\gamma_t < 0$ break

- Set $\alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t}$

- Set $D_{t+1}(i) = \frac{1}{Z_t} D_t(i) \exp(-\alpha_t h(x_i) y_i)$, where z_t is a normalizing factor

- Add α_t, λ_t to the joint.

Output: The weight factor λ .

We assign different weights to each of the classification result to get a better representation. Generally, for the task of deep neural networks, more hidden layers lead to greater weights as well as better robust results. And the weights will be higher. In fact, these weights are learnt like the process in [6]: AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. Our approach is quite similar in these respects, however our object is domain specific (i.e. only applicable to comparing which class the pedestrian belongs to). The proposed probability assignment is a weighted ensemble of likelihood ratio tests, made by the Algorithm 1, a brief review of which that can be found below.

In training the weights are iteratively updated. The training set is $T = \{P_{ab}^1, P_{ab}^2, \dots, P_{ab}^N\}$. Initializing the weight distribution of training data is representing like formula (4). And $D_t(x)$ is the probability matrix which means the distribution for the combination of two persons. λ_t is the weight factor.

$$D_1 = (\lambda_{1,1}, \dots, \lambda_{1,i}, \dots, \lambda_{1,N}) \quad s.t. \quad \lambda_1 = \frac{1}{N}, \quad i = 1, 2, \dots, N. \quad (4)$$

In algorithm 1, the $h(x_i)$ is weak classifier, $x \rightarrow \{-1, +1\}$. The error of the update is γ_t and it generates in formula (5).

$$\gamma_t = P((x_i) \neq y_i) = \sum_k^N \lambda_{ti} I(h(x_i) \neq y_i) \quad (5)$$

The coefficient of $h(x_i)$ is calculated in formula (6)

$$\alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t} \quad (6)$$

The update of weight of probability distribution is representing in formula (7) (8). (t denotes the numbers of iteration.)

$$D_{t+1} = (\lambda_{t+1,1}, \lambda_{t+1,2}, \dots, \lambda_{t+1,N}) \quad (7)$$

$$\lambda_{t+1,i} = \frac{\lambda_{ti}}{Z_t} \exp(-\alpha_t h(x_i) y_i) \quad (8)$$

Z_t is a normalized factor and represents in formula (9). It makes the D_{t+1} become a probability distribution.

$$Z_m = \sum_1^N \lambda_{ti} \exp(-\alpha_t y_i h(x_i)) \quad (9)$$

In algorithm 1, there are several iterations for the joint learning. The similarity probability is searched for the best weights w.r.t the current distribution and made the joint. And the weight for each probability matrix (i.e. the output of each SAE model) is assigned as formula (10). N denotes the numbers of the SAE model. It is 3 in our work.

$$f(x) = \sum_{t=1}^N \lambda_t D_t(x) \quad (10)$$

3 Experimental Results

In this section, we evaluate our multiple coarse-to-fine deep metric learning algorithm on two person re-identification benchmarks: the VIPeR and CUHK. For each dataset, we would give out the experiment result and compare with other previous methods. The detailed experimentation is described as following. We introduce the coarse-to-fine metric learning method from three aspects. i.e. the physical character, profile feature and facial feature. So there are three SAE models simulating the human's visual system. Besides, we implement our algorithm using the Andrew Ng's deep learning framework and write the code for our own architecture. It is time-consuming for roughly 6 days for the deepest network on high-performance computing platform.

3.1 The Datasets

The widely used VIPeR dataset is collected by Gray and Tao [8] and contains 1264 outdoor images obtained from two views of 632 persons. Each pair is made up of images of the same person from two different cameras, under different viewpoints, poses and light conditions, respectively. All images are normalized to 128×48 pixels. Views changes are the matched image pairs containing a viewpoint change of 90 degree.

The CUHK02 Campus dataset [14] contains 1816 persons and five pairs of camera views (P1-P5, ten camera views). They have 971, 306, 107, 193 and 239 persons respectively. Each person has two images in each camera view. This dataset is used to evaluate the performance when camera views in test are different than those in training. In our experiment, we choose view pair P1 for evaluation. And this view includes 971 subjects, 1942 images. Each subjects has two images from 2 camera views. And the instances in the two datasets could be seen as Figures.5.

3.2 Experimental Methods

Evaluation Protocol. Re-identification models are commonly evaluated by the cumulative match characteristic (CMC) curve. This measure indicates how the matching performance of the algorithm improves as the number of returned image increases. Given an algorithm and a test set of images of people with labels, each image in the test set is compared against the remaining images under the given algorithmic model and the position of the correct match is recorded.



Figure 5. Some typical samples of the two public dataset. And each column shows two images of the same person from two different cameras with significant changes on view point and illumination condition. (a) VIPeR dataset contains significant difference between different views. (b) CUHK is similar to VIPeR, but more challenge as it contains more person pairs.

The CMC curve indicates for each rank the fraction of test samples which had that rank or better. A perfect CMC curve would reach the value 1 for rank 1. Specifically, let $P = \{p_1, \dots, p_{|P|}\}$ be a probe set, where $|P|$ is the size of P . And $G = \{g_1, \dots, g_n\}$ a gallery set. For each probe images $p_i \in P$, all gallery images $g_i \in G$ are ranked by comparing the distance between p_i and g_i in ascending order. The image of the same person p_i in the gallery set is denoted as g_{p_i} . And the index of which in the sorted gallery is denoted as $r(g_{p_i})$. The CMC value of rank k is defined as formula (5)

$$CMC_k = \frac{\sum_{i=1}^{|P|} 1(r(g_{p_i}) \leq k)}{|P|} \quad (11)$$

where $1(\bullet)$ is the indicator function.

Data Augmentation. In the training set, the matched sample pairs (positive samples) are several orders fewer than non-matched pairs (negative samples). If they are directly used to train the deep network, the model tends to predict all the inputs as being non-matched. The easiest and most common method to solve this problem is to artificially enlarge the positive samples and randomly reduce the negative samples using label-preserving transformations [4, 5]. In our work, we exploited data augmentation by extracting random patches from the previous image pairs like [12]. For example, in the VIPeR dataset, the resolution of the combined image pairs is $128 * 96$, and we chose the size of each patch is $112 * 84$. For the CUHK dataset, the original resolution for each image is $160 * 60$. We tried to shrink the images into $128 * 48$ first. After that, the compound mode of the person images is similar with VIPeR dataset. Then, we shrank the combined image into $112 * 84$ for each negative sample pair. So the positive and negative samples pairs can be balanced via this way.

Training Platform and Training Strategy. We test the run time of the process after feedback selection. The networks are trained on the High-Performance Computing platform (HPC), which is composed of Dawning Cluster, HP Cluster, HP SMP Mainframe, GPU Cluster and Storage System. And our algorithm is trained on the Dawning Cluster, which includes 93 computing nodes, 6 I/O nodes, 2 management nodes. For each node, there are 2 CPU with 12 cores with 2.2 GHz and the memory is 128 GB. It takes about 6 days to train the deepest network for CUHK. In addition, as the training and testing samples take up too much memory, our training algorithm adopts the mini-batch stochastic gradient descent proposed in [7]. The training data is divided into several mini-batches. And training errors are calculated upon each mini-batch in the softmax layer and

get Back-Propagation to the lower layers.

3.3 Experimental Results on VIPeR Dataset

In the first experiment, we evaluated our algorithm on the VIPeR dataset. We exploited 316 person image pairs for training and 316 person image pairs for testing. Similar to [12] positive and negative sample pairs were balanced in the procedure. It means that the dimension of the feature of each image pair was $9408 (112 * 84)$. Then, the feature acted as the input of the SAE networks. Besides, each sample belongs to a certain class. As described above, if the image pair is matched. The label is $y = 1$ and $y = 0$, otherwise.

In our multiple SAE models, there were three kinds of deep networks. The hidden layers were set 2, 3 and 4 for these SAE networks, respectively. And the hidden units of each network were set 1000, correspondingly. In addition, for the single auto-encoder network, the numbers of units for each hidden layer were the same. We exploited the SAE- K (The K denotes the numbers of the hidden layers.) to represent the configuration of each SAE network. In the training phase, the input of the SAE network was $9408 * 100000$ (we randomly selected 100000 samples which included positive and negative ones for training). There was a label for each one. At last, the output of the deep network was following by a softmax classifier. There were about 400 iterations for training the network architecture. In the testing phase, we exploited 316 samples pairs for predicting the accuracy. Three kinds of probability matrix were generated by three kinds of SAE networks. Then, the weighted assignment mechanism was used for making the final decision. After that, the probability was transformed into the recognition accuracy. The final performance would be confirmed through this way. After training an auto-encoder network, we would like to visualize the weights (filters) that learned by the algorithm and try to understand what has been learnt. Figure 6 shows some filters learned by the first hidden layer of our network. The filters of network have different texture patterns, which mean that they capture the information in a unified manner.

The experiment results and all the Cumulative Matching Characteristic (CMC) curves are shown in Figure 7. It shows not only the 3 results for 3 different networks, but also the combined results after balancing the similarities. From the figure, we conclude that the matching results of the 3 networks are different, and each of them is low. The CMC(1) for single SAE network is not very high. And the more hidden layers of the network, the higher of the recognition ac-

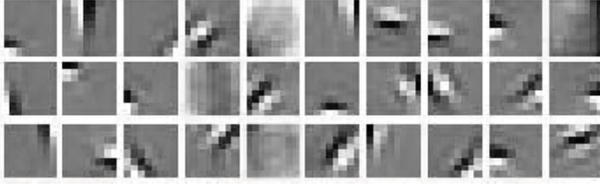


Figure 6. Visualization of Features: Visualization of some filters learned by a single auto-encoder network trained on VIPeR. The weights (called filters) of the first hidden layer for the auto-encoder network can be visualized.

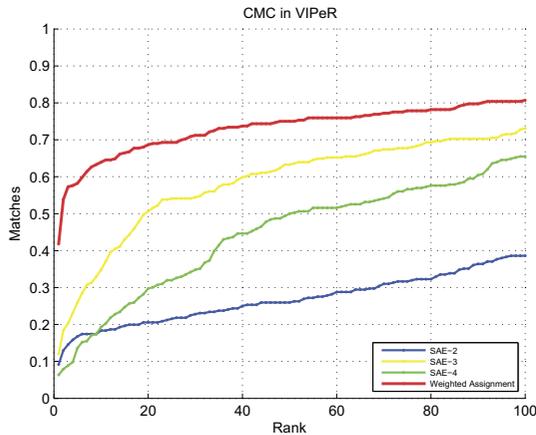


Figure 7. The recognition accuracies for different SAE models in VIPeR. The SAE- K denotes different hidden layers for the deep networks.

curacy for CMC(1). For the SAE-3, the accuracy is better than SAE-2. After exploiting the weighted assignment mechanism, the holistic recognition accuracy would be better than the single one and this phenomenon is consistent with our intuition.

As it is a classification problem in our work. There would be a final holistic accuracy for each SAE network and the classification network. For the test samples, each of them has a label which indicates the ground truth. It is a sign that was predicted by the softmax classifier. There are positive and negative samples in the test datasets. The holistic classification accuracy can manifest the performance of the model. The performances of the three deep models (SAE network and classification network) are shown in Figure 8. The accuracy indicates the correct classification for each test sample. If the accuracy is higher, the model is better to train. From the figure, the results are consistent with the CMC curves in Figure 7. The CMC(1) for SAE-4 is lower than SAE-2 in the figure. We think that there may exist too many connections and associated parameters between the adjacent layers in SAE-4. And they are difficult to tailor the performance.

Comparing with other metric learning methods, our algorithm has gotten the best recognition rate in CMC(1). The results can be seen as in Table 1. We compared with other six metric learning algorithms. The top two rows are the conventional metric learning methods. And the next four are the deep metric learning algorithms. We can see that our method enjoys the highest accuracy in CMC(1). When rank=10, our result is not very competitive. We guess that the results of the rank 10 may be led by the structure of SAE-4 which involves more hidden layers and associated parameters. Because the parameters are

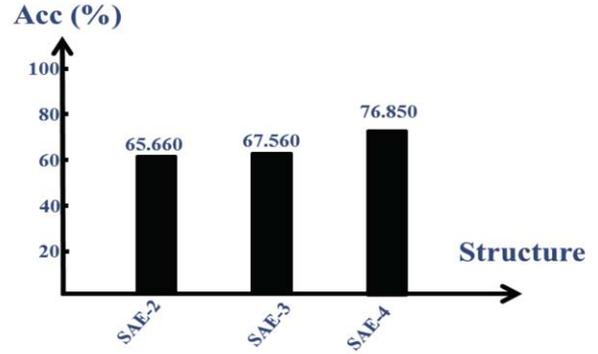


Figure 8. The overall classification performance in VIPeR for each deep model. Acc means the accuracy. From this figure, we can see that the coarse-to-fine classification mode is just like the character of the human brain which has many different levels of visual way. The SAE- K denotes the configurations of the SAEs.

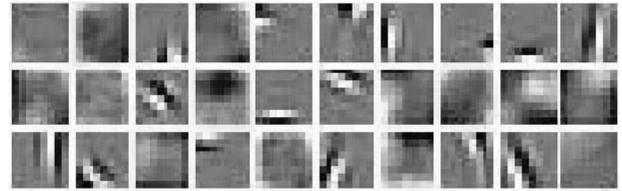


Figure 9. Some filters are learned on the CUHK.

over-fitting and the single result lower the holistic rank performance.

Table 1. Comparative results with the other metric learning algorithms on VIPeR.

Methods	Deep or Not	r=1	r=10
KISSME	Not	19.6%	62.2%
LMNN	Not	19.0%	58.1%
DML [24]	Yes	28.23%	73.45%
DDML [10]	Yes	29.56%	61.71%
DTML [11]	Yes	32.12%	65.92%
DCA [1]	Yes	34.81%	76.25%
Ours	Yes	41.77%	66.92%

3.4 Experimental Results on CUHK Dataset

In the second experiment, we evaluated our method on the CUHK dataset. The resolution of CUHK Campus is 60×160 . Before training, we scaled them to 48×128 first. This dataset included 970 persons, which was divided into 485 for training and 485 for testing. They were also randomly selected. For each person image, we also preprocessed it like the VIPeR dataset. And the compound mode for each person image was the same as the first experiment. Some of filters learning from the first hidden layer show in Figure 9. In this experiment, there were also three kinds of deep neural networks to train. The architectures of the SAE networks were set via the same way like the previous one. In the training phase, we preprocessed the combined image pairs like the way in the VIPeR dataset. And the size for the input sample was also $9408(112 \times 84)$. It was a label for each

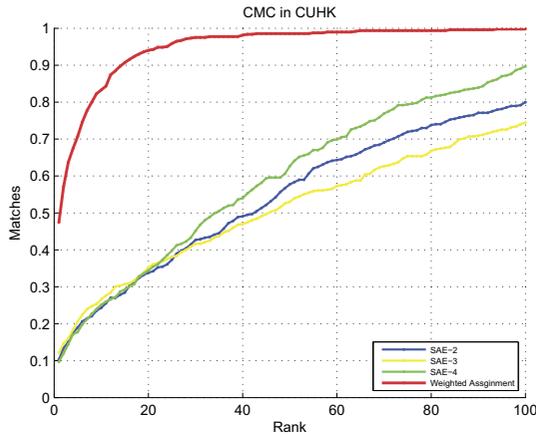


Figure 10. The recognition accuracies for different SAE networks in CUHK. The symbols denote the similar meaning like Figure 6.

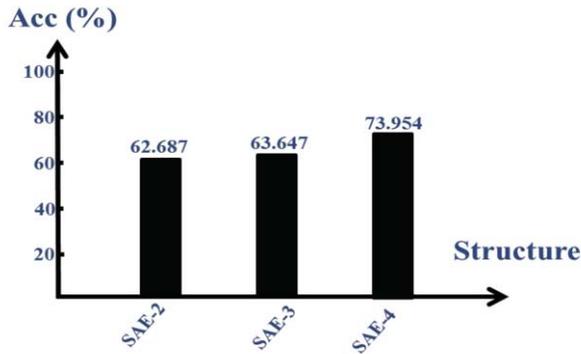


Figure 11. The overall classification performance in CUHK for each SAE network. The meaning of each notation in the figure is the same as in VIPeR.

one. The hidden layers for each SAE networks were set 2, 3 and 4, respectively. And the hidden units of each deep neural network were set 800, correspondingly. There were about 300 iterations for training the network architecture. At last, each of the auto-encoder networks was following by a softmax classifier. And the fine tuning was executed for the whole model via the back-Propagation algorithm. The output for each softmax classifier was the probability that a pair belongs to a certain class. A final accuracy was obtained via handling these probabilities. The recognition result can be seen as Figure 10.

Table 2. Comparative results with other metric learning algorithms on CUHK.

Methods	Deep or Not	r=1	r=10
KISSME	Not	29.40%	60.22%
LMNN	Not	21.17%	57.53%
FPNN [15]	Yes	27.87%	81.07%
DML [24]	Yes	16.17%	45.82%
Ours	Yes	47.42%	83.29%

From Figure 10, we can see that the single deep neural network for recognition performance is not very high. But the result is enhanced via exploiting the weighted assignment mechanism. This is

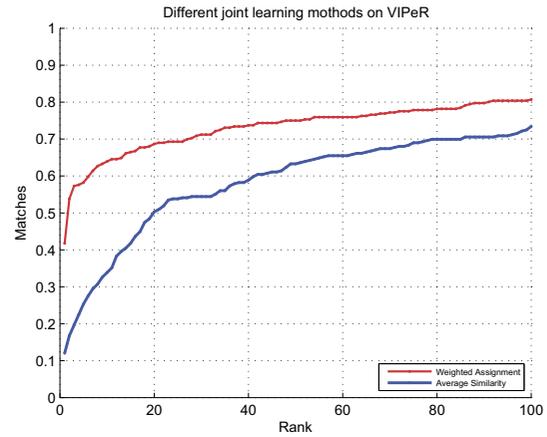


Figure 12. The different joint learning strategies on VIPeR. The weight assignment is better than average strategy.

in accordance with our intuition. The multi-scale metric learning is better than the single measure manner. The reason may be the same as the first experiment. And the result is not very precise. The holistic performance in this dataset is shown in Figure 11.

Comparing with other deep metric learning methods, our algorithm get the competitive result on the same dataset. The comparative results can be seen as Table 2. From it, we can see that our algorithm get the best accuracy comparing with other methods in CMC(1) and CMC(10). In addition, our model is simpler than other deep models for its architecture and training process comparing with [15, 1, 24]. Because they get involved into the structure modification, which take more time to train the deep networks. In our work, the deep models are just used to classification and the results are joining together to get the final result. The time complexity and space complexity are simpler comparing with previous methods.

3.5 Performance Verification on Joint Learning Strategies

In this section, we would compare the joint strategies for our previous experiments. Firstly, we exploited the average similarity strategy as the joint learning on the two common datasets. Comparing with the different strategies, the experiment result can be seen as Figure as 12. From the figure, we can see that the performance of weights assignment mechanism is better than average. In CUHK dataset, the performance for compared strategy is also different. The results can be seen as Figure 13. In fact, the average similarity strategy is the special case for the weights assignment mechanism. As the architecture of the neural network is different, the degree of contribution for recognition accuracy is also different. So there are different weight factor and the joint strategy is very important for the final result.

4 Conclusions and Future Work

In this work, we have presented a method which utilizes multiple coarse-to-fine auto-encoder models to address person re-identification problem under varied environmental changes. In our algorithm, we have trained several different SAE networks, with each followed by a softmax classifier. So that the brain's visual cortex can be simulated by our established deep neural networks with different

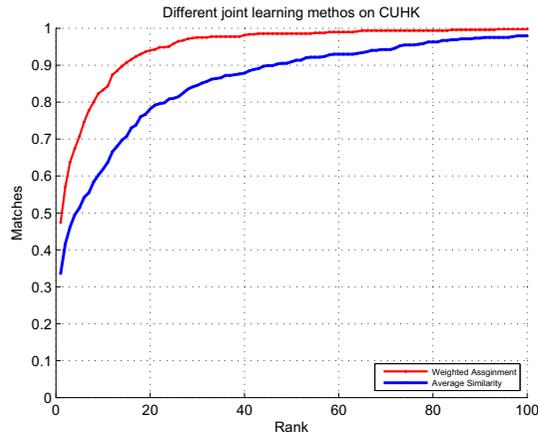


Figure 13. The different joint learning strategies on CUHK.

hidden layers. The preprocessed person image pairs via subtracting the mean value are used for network input and a couple of classification results are then produced. Finally, a weighted assignment mechanism is further used to boost recognition accuracy for the obtained classification results. Extensive experimental results on two public datasets have shown the superiority of our algorithm. Our established multiple coarse-to-fine deep metric learning approach can be extended to other visual applications, such as images classification, object detection and so on.

ACKNOWLEDGEMENTS

This research is supported by Development Program of China 863 Program (No. 2015AA016306), National Nature Science Foundation of China (No. 61231015, 61303114), The EU FP7 QUICK project under Grant Agreement No. PIRSES-GA-2013-612652*, Nature Science Foundation of Jiangsu Province (SBK2016040692), Specialized Research Fund for the Doctoral Program of Higher Education (No. 20130141120024), Nature Science Foundation of Hubei Province (2014CFB712) and National High Technology Research.

REFERENCES

- [1] Ejaz Ahmed, Michael Jones, and Tim K Marks, 'An improved deep learning architecture for person re-identification', *Differences*, **5**, 25, (2015).
- [2] Sola O Ajiboye, Philip Birch, Christopher Chatwin, and Rupert Young, 'Hierarchical video surveillance architecture: a chassis for video big data analytics and exploration', in *IS&T/SPIE Electronic Imaging*, pp. 94070K–94070K. International Society for Optics and Photonics, (2015).
- [3] Yoshua Bengio, 'Learning deep architectures for ai', *Foundations and trends® in Machine Learning*, **2**(1), 1–127, (2009).
- [4] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber, 'Multi-column deep neural networks for image classification', in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3642–3649. IEEE, (2012).
- [5] Dan C Ciresan, Ueli Meier, Jonathan Masci, Luca M Gambardella, and Jürgen Schmidhuber, 'High-performance neural networks for visual object classification', *arXiv preprint arXiv:1102.0183*, (2011).
- [6] Piotr Dollár, Zhuowen Tu, Hai Tao, and Serge Belongie, 'Feature mining for image classification', in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8. IEEE, (2007).
- [7] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio, 'Maxout networks', *arXiv preprint arXiv:1302.4389*, (2013).
- [8] Douglas Gray, Shane Brennan, and Hai Tao, 'Evaluating appearance models for recognition, reacquisition, and tracking', in *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, volume 3. Citeseer, (2007).
- [9] Martin Hirzer, Peter M Roth, and Horst Bischof, 'Person re-identification by efficient impostor-based metric learning', in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pp. 203–208. IEEE, (2012).
- [10] Junlin Hu, Jiwen Lu, and Yap-Peng Tan, 'Discriminative deep metric learning for face verification in the wild', in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1875–1882. IEEE, (2014).
- [11] Junlin Hu, Jiwen Lu, and Yap-Peng Tan, 'Deep transfer metric learning', in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 325–333. IEEE, (2015).
- [12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, 'Imagenet classification with deep convolutional neural networks', in *Advances in neural information processing systems*, pp. 1097–1105, (2012).
- [13] Igor Kviatkovsky, Amit Adam, and Ehud Rivlin, 'Color invariants for person reidentification', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **35**(7), 1622–1634, (2013).
- [14] Wei Li and Xiaogang Wang, 'Locally aligned feature transforms across views', in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3594–3601. IEEE, (2013).
- [15] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang, 'Deepreid: Deep filter pairing neural network for person re-identification', in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 152–159. IEEE, (2014).
- [16] Chao Liang, Binyue Huang, Ruimin Hu, Chunjie Zhang, Xiaoyuan Jing, and Jing Xiao, 'A unsupervised person re-identification method using model based representation and ranking', in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 771–774. ACM, (2015).
- [17] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio, 'Contractive auto-encoders: Explicit invariance during feature extraction', in *International Conference on Machine Learning (ICML)*, pp. 833–840, (2011).
- [18] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, 'Learning representations by back-propagating errors', *Cognitive modeling*, **5**, 3, (1988).
- [19] Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang, 'Person re-identification by discriminative selection in video ranking', (2016).
- [20] Yimin Wang, Ruimin Hu, Chao Liang, Chunjie Zhang, and Qingming Leng, 'Camera compensation using a feature projection matrix for person reidentification', *Circuits and Systems for Video Technology, IEEE Transactions on*, **24**(8), 1350–1361, (2014).
- [21] Zheng Wang, Ruimin Hu, Chao Liang, Yi Yu, Junjun Jiang, Mang Ye, Jun Chen, and Qingming Leng, 'Zero-shot person re-identification via a cross-view consistency', *IEEE Transactions on Multimedia*, **18**(2), 260–272, (2016).
- [22] Zheng Wang, Ruimin Hu, Yi Yu, Chao Liang, and Wenxin Huang, 'Multi-level fusion for person re-identification with incomplete marks', in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1267–1270. ACM, (2015).
- [23] Mang Ye, Chao Liang, Zheng Wang, Qingming Leng, and Jun Chen, 'Ranking optimization for person re-identification via similarity and dissimilarity', in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1239–1242. ACM, (2015).
- [24] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li, 'Deep metric learning for person re-identification', in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pp. 34–39. IEEE, (2014).
- [25] Rui Zhao, Wanli Ouyang, and Xiaogang Wang, 'Unsupervised saliency learning for person re-identification', in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3586–3593. IEEE, (2013).
- [26] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang, 'Person re-identification by probabilistic relative distance comparison', in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 649–656. IEEE, (2011).

How Hard Is Bribery with Distance Restrictions?

Yongjie Yang^{1,2} and Yash Raj Shrestha³ and Jiong Guo¹

Abstract. We study the complexity of the bribery problem with distance restrictions. In particular, in the bribery problem, we are given an election and a distinguished candidate p , and are asked whether we can make p win/not win the election by bribing at most k voters to recast their votes. In the bribery problem with distance restrictions, we require that the votes recast by the bribed voters are close to their original votes. To measure the closeness between two votes, we adopt the prevalent Kendall-Tau distance and the Hamming distance. We achieve a wide range of complexity results for this problem under a variety of voting correspondences, including the Borda, Condorcet, Copeland $^\alpha$ for every $0 \leq \alpha \leq 1$ and Maximin.

1 Introduction

Voting is a common method for preference aggregation and collective decision-making, and has applications in many areas such as political elections, multi-agent systems, web spam reduction and pattern recognition [14, 15, 32, 37]. In real-world applications, there exist many potential factors that may affect the result of voting. For instance, a strategic individual may alter some of the already submitted votes, or the votes that the voters intend to submit. An example of scenario is when a candidate attempts to change the preferences of voters by running a campaign, or in more extreme cases where this strategy involves paying voters to change their votes, or bribing election officials to get access to already submitted votes in order to modify them. A prominent method to address such issues concerning strategic behavior is to use complexity as a barrier [23, 31, 39]. The key point is that if it is computationally difficult for the strategic individual to figure out how to successfully change the result, he may refrain from attacking the voting.

In this paper, we study a voting model in which an external agent attempts in switching the voters' preferences in order to make a distinguished candidate win the election (*constructive*), or lose the election (*destructive*). The external agent's capacity is bounded by a budget constraint. We observe that, while the voter is willing to recast a new vote persuaded by an external agent, he may nevertheless prefer to submit a preference that deviates as little as possible from his true preference. Indeed, if voting is public, he may be worried that switching his preference completely may harm his reputation, yet he will not be caught out if his final preference is sufficiently similar to his true preference. We call this model *distance restricted bribery*. To quantify the amount of deviation of the new recast vote and the original vote of a bribed voter, we use two *distance* measures. Particularly, we consider what are arguably the most prominent distances on votes, namely, the *Hamming distance* (see, e.g., [8, 16, 34, 35, 40] for discussions of Hamming distance in

the context of voting) and *Kendall-Tau distance* (KT-distance for short. See [2, 5, 6, 9] for further discussions). The definitions of these two distances are in Section 2. We study the complexity of the voting model for various voting systems, including the Borda, Condorcet, Maximin and Copeland $^\alpha$. We obtain a broad range of results showing that the complexity of bribery depends closely on the settings. A primary conclusion from our results is that the distance restricted bribery problem remains NP-hard for some voting systems even when the distance is bounded by a very small constant. On the other hand, there exist voting systems for which the distance restricted bribery problem is polynomial-time solvable, when the distance is bounded by 1 or 2, and voting systems for which the distance restricted bribery problem is polynomial-time solvable regardless of the values of the distance bound. In particular, we achieve several dichotomy results with respect to the values of the distance bound. For instance, for Condorcet, the constructive restricted bribery problem with KT-distance restriction is polynomial time solvable if the distance is bounded by at most 2; and NP-hard otherwise. See Table 1 for further details on our results. Due to space limitation, several proofs are deferred to a full version of the paper.

Our model is closely related to the bribery problem which has been widely studied in computational social choice. Faliszewski, Hemaspaandra and Hemaspaandra [22] introduced the *bribery* problem, where one is to decide whether a distinguished candidate can become a winner (constructive) or be prevented from being a winner (destructive) by recasting at most \mathcal{R} (a given integer) votes. Clearly, the bribery problem studied in [22] can be considered as a distance restricted bribery problem with the distance bound being considerably large (depends on which distance concept we adopt). The complexity of the bribery problem proposed in [22] has been extensively studied in the literature for various voting systems. In particular, it is known that, for Borda and Condorcet, the constructive bribery problem is NP-hard, while the destructive counterpart turned out to be polynomial-time solvable [11, 24]. For Maximin and Copeland $^\alpha$, both the constructive and the destructive bribery problems are NP-hard [24, 26]. Our study clearly complements these complexity results. Of particular interest is that our study shed significant light on the complexity border between polynomial-time solvability and NP-hardness of the bribery problem, with respect to the distance bound. Recently, exploring the complexity border for various strategic voting problems, with respect to diverse structural parameters, has received a considerable amount of attention of researchers from both theoretical computer science and computational social choice communities [10, 18, 25, 43, 44, 45, 46]. The reason is that in many real-world applications, the votes of the voters are subject to some natural combinatorial restrictions.

Our model is also related to some other variants of the bribery problem. In [22], the authors also considered the $\$$ bribery version where each voter has a price to change his vote. Later,

¹ Shandong University, Jinan, China, email:jguo@sdu.edu.cn

² Saarland University, Germany, email:yongjie@mmci.uni-saarland.de

³ ETH Zürich, Zürich, Switzerland, email:yshrestha@ethz.ch

	General		Kendall-Tau distance							Hamming distance			
	Const	Dest	Const (ℓ)				Dest (ℓ)			Const (ℓ)		Dest (ℓ)	
			1	2	3	≥ 4	1	≥ 2	3	≥ 4	2	≥ 3	
Borda	NP-h \diamond	P	P			NP-h (Thm. 2)	P	P	P	P		P	P
Condorcet	NP-h \clubsuit	P \clubsuit	P	P		NP-h (Thm. 4)	P	P	P	P	NP-h (Thm. 10)	P	P
Copeland $^\alpha$ $0 \leq \alpha \leq 1$	NP-h \heartsuit	NP-h \heartsuit				NP-h (Thm. 5)				NP-h (Thm. 5) UNI: $\ell \geq 5$	NP-h (Thm. 9)		
Maximin	NP-h \clubsuit	NP-h \clubsuit				NP-h (Thm. 7)	P			NP-h	NP-h (Thm. 11)		

Table 1. A summary of the complexity of the bribery problems. Here, the general case refers to the bribery problem studied in [22]. In the table, “Const” stands for “Constructive” and “Dest” stands for “Destructive”. Moreover, “ ℓ ” is the distance upper bound. Furthermore, “P” stands for “polynomial-time solvable” and “NP-h” stands for “NP-hard”. Unless stated otherwise, all results shown in this table apply to both the unique-winner model and the nonunique-winner model. The complexity of the problems whose distance bound ℓ is not shown in the table remains open. The polynomial-time solvability results for the Kendall-Tau distance restriction are from Theorem 1, and the polynomial-time solvability results for the Hamming distance restriction are from Theorem 8. The result marked by \diamond is from [11], by \clubsuit from [24], and by \heartsuit from [26].

Faliszewski [21] proposed a new notion of bribery, called *nonuniform bribery*, where a voter’s price may depend on the nature of changes he is asked to implement. A similar notion called *microbribery* was considered in [26]. Elkind, Faliszewski and Slinko [17] introduced the framework of *swap bribery* where the briber can ask a voter to perform a sequence of swaps; each swap changes the relative order of two candidates that are currently adjacent in this voter’s preference list. Moreover, each swap may have a different price; and the price of a bribery is the sum of the prices of all swaps that it involves. In the same paper [17], the authors also studied the *shift bribery* problem, which is a restricted variant of swap bribery. In particular, in the shift bribery problem, only swaps involving the distinguished candidate are allowed. Recently, Pini, Rossi and Venable [42] investigated the complexity of bribery in voting with soft constraints, where each candidate is an element of the Cartesian product of the domains of some variables, and voters express their preferences over the candidates via soft constraints. Mattei et al. [38] studied the complexity of bribery in CP-nets.

In addition, our study is related to Obraztsova and Elkind’s work [41] where a manipulator aims to make a distinguished candidate win or lose the election by casting an untruthful vote. Here, the untruthful vote should be as close as possible to the truthful vote of the manipulator. They examined this problem for several voting systems with the adoption of three prominent distances, namely, the KT-distance, the footrule distance, and the maximum displacement distance. Our model differs from theirs in the following aspects. First, in our settings, at most \mathcal{R} voters might be bribed, however, they considered only one such voter. Second, their problems ask the manipulator to cast an untruthful vote which is as close as possible to the truthful vote. We mainly focus on the settings where the bribed voters must cast their votes which have a small constant discrepancy from their original votes.

2 Preliminaries

Voting system. A *voting system* can be specified by a set \mathcal{C} of *candidates*, a multiset $\Pi_{\mathcal{V}} = \{\pi_{v_1}, \pi_{v_2}, \dots, \pi_{v_n}\}$ of *votes* cast by a corresponding set $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ of *voters* (π_{v_i} is cast by v_i), and a *voting correspondence* τ which maps the *election* $\mathcal{E} = (\mathcal{C}, \Pi_{\mathcal{V}}, \mathcal{V})$ to a subset of candidates $\tau(\mathcal{E})$, the *winners*. We often discard \mathcal{V} from the above notation for election \mathcal{E} since $\Pi_{\mathcal{V}}$ is sufficient to determine the winners. If there is only one winner, we call it a *unique winner*; otherwise we call them *co-winners*. Moreover, each vote $\pi_v \in \Pi_{\mathcal{V}}$ is defined as a linear order over

the candidates. Throughout this paper, we interchangeably use the terms “vote” and “voter”. The linear order of a vote is also called the *preference* of the vote over the candidates. For convenience, we use \succ_v to denote the preference of the vote cast by the voter v . Therefore, for a voter v who prefers the candidate a to b to c , the vote will be written as $\pi_v : a \succ_v b \succ_v c$. In context where \succ_v is clearly known to be whose preference, we drop v from \succ_v . We say a is *ranked above* b in a vote π_v if $a \succ_v b$. The *position* of a candidate $a \in \mathcal{C}$ in a vote π_v , denoted as $pos_{\succ_v}(a)$ (or simply $pos_v(a)$), is defined as $|\{b \in \mathcal{C} \mid b \succ_v a\}| + 1$, i.e., the number of candidates ranked above a in the vote plus 1.

For two candidates c and c' in an election $\mathcal{E} = (\mathcal{C}, \Pi_{\mathcal{V}})$, let $N_{\mathcal{E}}(c, c')$ denote the number of votes which prefer c to c' . We drop the index \mathcal{E} when it is clear from context. If $N_{\mathcal{E}}(c, c') > N_{\mathcal{E}}(c', c)$, we say c *beats* c' by $N_{\mathcal{E}}(c, c')$ in \mathcal{E} ; otherwise if $N_{\mathcal{E}}(c, c') = N_{\mathcal{E}}(c', c)$ we say c *ties* c' in \mathcal{E} .

Voting Correspondences. We mainly study the following voting correspondences.

Borda: Every voter gives 0 points to his last-ranked candidate, 1 point to his second-last ranked candidate and so on. A candidate with the highest score is a winner.

Copeland $^\alpha$ ($0 \leq \alpha \leq 1$): For a candidate c , let $B(c)$ be the set of candidates who are beaten by c , and $T(c)$ the set of candidates who tie with c . The Copeland $^\alpha$ score of c is then defined as $|B(c)| + \alpha \cdot |T(c)|$. A Copeland $^\alpha$ winner is a candidate with the highest score.

Maximin: For a candidate c , the Maximin score of c is defined as $\min_{c' \in \mathcal{C} \setminus \{c\}} N(c, c')$. A Maximin winner is a candidate with the highest Maximin score.

Condorcet: A Condorcet winner is a candidate with Copeland 0 score $m - 1$, and a weak Condorcet winner is a candidate with Copeland 1 score $m - 1$, where m denotes the number of candidates. It is known that Condorcet winner (weak Condorcet winner) may not exist in an election. However, if there is a Condorcet winner in an election, it must be unique. Given an election, the voting correspondence returns all weak Condorcet winners if there is a weak Condorcet winner; otherwise, it returns the empty set.

Distance. We mainly consider the Hamming distance and the KT-distance in this paper. The Hamming distance, named after Richard Hamming, was initially defined on strings [28]. In particular,

the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. For example, the Hamming distance between the string “ $a\ 1\ b\ b$ ” and the string “ $a\ b\ 1\ b$ ” is two since there are two positions (the second and the third positions) where the symbols are different. In the context of Hamming distance in this paper, we regard each vote as a string with each element being (the name of) a candidate. For example, the vote defined as $a \succ b \succ c \succ d$ will be considered as the string “ $a\ b\ c\ d$ ”. Hence, the *Hamming distance* between every two votes with preferences \succ_1, \succ_2 , denoted as $D_{HAM}(\succ_1, \succ_2)$, is the Hamming distance between the two strings corresponding to the two votes.

The KT-distance was coined by Maurice Kendall [33]. In a formal way, the *KT-distance* between two votes with preferences \succ_1 and \succ_2 , respectively, denoted as $D_{KT}(\succ_1, \succ_2)$, is equal to $|\{(a, b) | a, b \in \mathcal{C}, a \succ_1 b \text{ and } b \succ_2 a\}|$. Equivalently, the KT-distance between two votes can be defined as the minimum number of swaps of adjacent candidates needed to transform one into the other [4]. In addition, the KT-distance also turns out to be equal to the number of exchanges needed in a bubble sort (see [1] for an introduction to bubble sort) to convert one full ranking to the other [19]. Due to this fact, the KT-distance is also referred to as *bubble-sort distance* in the literature [7, 12, 19, 20].

Problem Definitions. We mainly study the following problems for different voting correspondences. In what follows, let τ be a voting correspondence and “DIST” a distance function. In this paper, “DIST” can be “KT” for the KT-distance or “HAM” for the Hamming distance. Moreover, ℓ is a positive integer. For two votes with preferences \succ_1, \succ_2 and a distance “DIST”, we say these two votes are DIST(ℓ)-close if $D_{DIST}(\succ_1, \succ_2) \leq \ell$. For each problem, we study both the *unique-winner model* and the *nonunique-winner model*. In the unique-winner model (resp. nonunique-winner model) for τ not being Condorcet, a candidate $c \in \mathcal{C}$ wins an election $\mathcal{E} = (\mathcal{C}, \Pi_V)$ if and only if $\tau(\mathcal{E}) = \{c\}$ (resp. $c \in \tau(\mathcal{E})$), i.e., c is the unique winner (resp. c is either the unique winner or a co-winner). For Condorcet, a candidate c wins an election (\mathcal{C}, Π_V) in the unique-winner model (resp. nonunique-winner model) if and only if c is the Condorcet winner (resp. a weak Condorcet winner). In the following, let “MOD” be either “UNI” standing for “unique-winner model”, or “NON” standing for “nonunique-winner model”.

Constructive Distance Restricted Bribery under τ
(C-DIST(ℓ)- τ -MOD)

Input: An election (\mathcal{C}, Π_V) , a distinguished candidate $p \in \mathcal{C}$, and a positive integer $\mathcal{R} \leq |\Pi_V|$. Here, p does not win the election (\mathcal{C}, Π_V) under τ .

Question: Is it possible to make p win the election by replacing (recasting) at most \mathcal{R} votes, under τ ? Here, a vote can only be replaced with a DIST(ℓ)-close vote.

Destructive Distance Restricted Bribery under τ
(D-DIST(ℓ)- τ -MOD)

Input: An election (\mathcal{C}, Π_V) , a distinguished candidate $p \in \mathcal{C}$, and a positive integer $\mathcal{R} \leq |\Pi_V|$. Here, p wins (\mathcal{C}, Π_V) under τ .

Question: Is it possible to prevent p from winning the election by replacing (recasting) at most \mathcal{R} votes, under τ ? Here, a vote can only be replaced with a DIST(ℓ)-close vote.

We give either polynomial-time algorithms or NP-hardness reductions for the above problems. Our hardness proofs are based on reductions from the X3C problem and the X4C problem defined as follows. Let ℓ be 3 or 4.

Exact ℓ -Set Cover (X ℓ C)

Input: A universal set $U = \{c_1, c_2, \dots, c_{\ell \cdot \kappa}\}$ and a collection $S = \{s_1, s_2, \dots, s_n\}$ of ℓ -subsets of U .

Question: Is there an $S' \subseteq S$ such that $|S'| = \kappa$ and each $c_i \in U$ appears in exactly one set of S' ?

It is known that both the X3C and the X4C problems are NP-hard [3, 27]. In particular, the NP-hardness of both problems holds even when each element $c_i \in U$ occurs in exactly 3 subsets of S [3, 27]. Notice that under this assumption, $n = 3\kappa$ in X3C.

The words “promote” and “degrade” are often used in NP-hardness reductions and description of polynomial-time algorithms with specific meanings in this paper. In particular, for a vote π and a candidate c , *promoting* the candidate c by ℓ positions for some $\ell < \text{pos}_\pi(c)$ means recasting the vote π as follows: (1) rank c in the $(\text{pos}_\pi(c) - \ell)$ -th position; (2) rank every candidate c' with $\text{pos}_\pi(c) > \text{pos}_\pi(c') \geq \text{pos}_\pi(c) - \ell$ in the $(\text{pos}_\pi(c') + 1)$ -th position; and (3) rank all the remaining candidates in their original positions. *Degrading* the candidate c by ℓ positions means recasting the vote π as follows: (1) rank c in the $(\text{pos}_\pi(c) + \ell)$ -th position of π ; (2) rank every candidate c' with $\text{pos}_\pi(c) < \text{pos}_\pi(c') \leq \text{pos}_\pi(c) + \ell$ in the $(\text{pos}_\pi(c') - 1)$ -th position; and (3) rank all the remaining candidates in their original positions. See Figure 1 for an illustration.

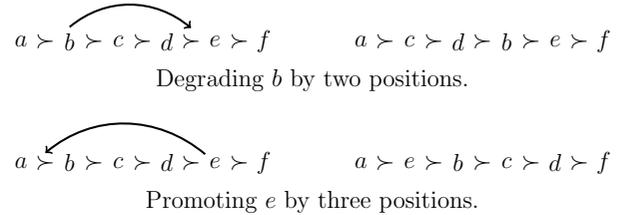


Figure 1. Illustrations of promoting and degrading candidates.

3 Kendall-Tau Distance Restricted Bribery

In this section, we investigate the bribery problem with KT-distance restrictions. In the following, we summarize our results in several theorems. We begin with some polynomial-time solvability results.

Theorem 1 *The following problems are polynomial-time solvable: C-KT(1)-Borda-UNI/NON, C-KT(ℓ)-Condorcet-UNI/NON for $\ell = 1, 2$, D-KT(ℓ)-Borda-UNI/NON for every possible ℓ , D-KT(ℓ)-Condorcet-UNI/NON for every possible ℓ , D-KT(1)-Maximin-UNI/NON.*

PROOF. Due to space limitation, we give only the polynomial-time algorithms for the C-KT(2)-Condorcet-UNI problem and the D-KT(1)-Maximin-UNI problem. Let $((\mathcal{C}, \Pi_V), p \in \mathcal{C}, \mathcal{R})$ be a given instance. Let m be the number of candidates and n the number of votes, i.e., $m = |\mathcal{C}|$ and $n = |\Pi_V|$.

C-KT(2)-Condorcet-UNI. We reduce the problem to the SIMPLE B-EDGE COVER OF MULTIGRAPHS problem which is polynomial-time solvable [36].

SIMPLE B-EDGE COVER OF MULTIGRAPHS (B-ECM)

Input: An undirected multigraph $G = (U, E)$ where U is the set of vertices and E is the set of edges, a function $f : U \rightarrow \mathbb{Z}^+$ and a positive integer κ .

Question: Does there exist an $E' \subseteq E$ such that $|E'| \leq \kappa$ and every vertex $u \in U$ is incident to at least $f(u)$ edges in E' ?

Now we show how to reduce the C-KT(2)-Condorcet-UNI problem to the B-ECM problem. For each candidate $c \in C \setminus \{p\}$ which is not beaten by p , we create a vertex. For simplicity, we still use c to denote the corresponding vertex. We define \overleftarrow{p}_v for a vote π_v where p is not ranked in the top as follows: if p is not ranked in the second highest position in π_v , then \overleftarrow{p}_v is the set consisting of the two candidates immediately ranked above p in π_v ; if p is ranked in the second-highest position in π_v , then \overleftarrow{p}_v is the set consisting of the candidate ranked in the highest position in π_v . For example, for a vote π_v with preference $a \succ b \succ c \succ p \succ d$, $\overleftarrow{p}_v = \{b, c\}$, while for a vote π_u with preference $a \succ p \succ c \succ b \succ d$, $\overleftarrow{p}_u = \{a\}$. The edges are created according to the votes. Precisely, for each vote π_v with $|\overleftarrow{p}_v| = 2$, if both candidates of $\overleftarrow{p}_v = \{c, c'\}$ are not beaten by p , we create an edge between c and c' . On the other hand, if only one of \overleftarrow{p}_v is not beaten by p , we introduce a new degree-1 vertex adjacent to the vertex in \overleftarrow{p}_v that is not beaten by p . For each vote π_v with $|\overleftarrow{p}_v| = 1$, if the candidate in \overleftarrow{p}_v is not beaten by p , we introduce a new degree-1 vertex adjacent to the candidate in \overleftarrow{p}_v . Now we come to the capacities of the vertices. Each vertex corresponding to a candidate c has a capacity $f(c) = (N(c, p) - N(p, c))/2 + 1$ whenever $N(c, p) - N(p, c) \equiv 0 \pmod{2}$, and has a capacity $f(c) = (N(c, p) + 1 - N(p, c))/2$ otherwise. Moreover, each newly introduced degree-1 vertex has capacity 0. The value of the capacity $f(c)$ indicates the minimum number of votes which rank c above p , that are needed to be replaced with votes which rank p above c in order to make p beat c . Finally, we set $\kappa = \mathcal{R}$.

Now we get an instance of the B-ECM problem. Moreover, given a solution E' of the instance of the B-ECM problem, we can get a solution for C-KT(2)-Condorcet-UNI in polynomial time. In particular, for each edge $(c, c') \in E'$, if none of $\{c, c'\}$ is a newly introduced degree-1 vertex, we recast the corresponding vote by promoting p by two positions; otherwise, we recast the corresponding vote by promoting p by one position.

D-KT(1)-Maximin-UNI: The algorithm first carries out a polynomial number of guesses. In particular, the algorithm guesses a candidate p' which prevents p from being the unique winner, an integer s which plays the role as an upper bound of the Maximin score of p in the final election and a lower bound of the Maximin score of p' in the final election, and a candidate q with $N(p, q) \leq s$ in the final election. These lead to at most $(m-1)^2 \times n$ subinstances where m is the number of candidates and n the number of votes. To make it clear, we give the formal definition of the subproblem.

Sub-D-KT(1)-Maximin-UNI

Input: An election $\mathcal{E} = (C, \Pi_V)$, a distinguished candidate $p \in C$, two further candidates $p', q \in C \setminus \{p\}$ (it may be that $p' = q$), and two integers $0 \leq s, \mathcal{R} \leq |\Pi_V|$.

Question: Is there a submultiset $\Pi_{\mathcal{T}} \subseteq \Pi_V$ of votes such that

- (1) $\Pi_{\mathcal{T}}$ contains at most \mathcal{R} votes; and
- (2) we can replace every vote $\pi_v \in \Pi_{\mathcal{T}}$ with a new vote obtained from π_v by swapping two consecutively ranked candidates so that $N(p, q) \leq s$ in the final election, and the Maximin score of p' is at least s in the final election?

Now we focus on solving the subproblem. Let Π_p be the multiset of votes which rank p immediately above q . Let $A = \{c \in C \setminus \{p'\} \mid N_{\mathcal{E}}(p', c) < s\}$. For each $c \in A$, let $\Pi_c \subseteq \Pi_V \setminus \Pi_p$ be the multiset of votes that rank c immediately above p' . Clearly, for every distinct two candidates $c, c' \in A$, $\Pi_c \cap \Pi_{c'} = \emptyset$. Moreover, for every $c \in A$, let $f(c) = s - N_{\mathcal{E}}(p', c)$. The algorithm works as follows. For each $c \in A$, arbitrarily choose $\min\{f(c), |\Pi_c|\}$ votes in Π_c , and replace each of them with a new vote obtained from the original vote by swapping c and p' ; then, set $f(c) := f(c) - \min\{f(c), |\Pi_c|\}$ and $\mathcal{R} := \mathcal{R} - \min\{f(c), |\Pi_c|\}$. If $\mathcal{R} < 0$ after doing so, we cannot make p' have a Maximin score at least s by replaying at most \mathcal{R} votes; and thus, the algorithm returns “No”. Otherwise, let $B = \{c \in A \mid f(c) > 0\}$. Then, for each $c \in B$, let $\bar{\Pi}_c$ be the multiset of votes in Π_p that rank c immediately above p' . If $|\bar{\Pi}_c| < f(c)$, the given instance is a No-instance (since we cannot make p' have a Maximin score at least s in the final election); and thus, we return “No”. Otherwise, we arbitrarily choose $\min\{f(c), |\bar{\Pi}_c|\}$ votes in $\bar{\Pi}_c$, and (1) replace each of them with a new vote obtained from the original vote by swapping c and p' ; (2) remove them from the multiset Π_p ; and (3) set $\mathcal{R} := \mathcal{R} - \min\{f(c), |\bar{\Pi}_c|\}$. If $\mathcal{R} < 0$ or $\min\{|\Pi_p|, \mathcal{R}\} < N_{\mathcal{E}}(p, q) - s$ after doing so, we return “No”. Otherwise, we return “Yes” since we can get a solution by replacing arbitrary $N_{\mathcal{E}}(p, q) - s$ votes in Π_p by new votes obtained from the original votes by swapping p and q . \square

Now we discuss the NP-hardness results. We begin with constructive distance restricted bribery for Borda. We have seen from Theorem 1 that the destructive counterpart turned out to be polynomial-time solvable for every possible value of ℓ . The following theorem shows, however, that constructive distance restricted bribery for Borda is NP-hard, even when the distance is bounded by a small constant. We first define some useful notations.

For an order $X = (x_1, x_2, \dots, x_i)$ over a set $\{x_1, x_2, \dots, x_i\}$, we denote by \overleftarrow{X} the reverse order of X , i.e., $\overleftarrow{X} = (x_i, \dots, x_2, x_1)$. For a subset $Y \subseteq \{x_1, x_2, \dots, x_i\}$, $X \setminus Y$ is the order obtained from X by deleting all the elements in Y . For example, for $X = (1, 4, 3, 8, 5)$ and $Y = \{4, 8\}$, $X \setminus Y = (1, 3, 5)$. For two subsets X and Y of candidates such that $X \cap Y = \emptyset$, and a vote with preference \succ , $X \succ Y$ means that every candidate in X is ranked above every candidate in Y in the vote.

Theorem 2 *C-KT(ℓ)-Borda-UNI/NON are NP-hard for all $\ell \geq 3$.*

PROOF. We give only proofs for the case $\ell = 3$ here. We first consider C-KT(3)-Borda-NON. The reduction is from X3C. Given an instance $\mathcal{F} = (U = \{c_1, c_2, \dots, c_{3\kappa}\}, S = \{s_1, s_2, \dots, s_m\})$ of X3C, we create an instance \mathcal{E} for C-KT(3)-Borda-NON as follows.

Candidates: For each $c \in U$, we create a corresponding candidate. For convenience, we still use c to denote the corresponding candidate. In addition, we create a set $Y = \{y_1, y_2, \dots, y_{6m-6}\}$ of $6m-6$ dummy candidates, each of which has a considerably less Borda score than that of any other candidate not in Y . For ease of exposition, we partition the dummy candidates into subsets Z_1, Z_2, \dots, Z_m . To be precise, for each $i = 1, 2, \dots, m-2$, $Z_i = \{y_{6i-5}, y_{6i-4}, y_{6i-3}, y_{6i-2}, y_{6i-1}, y_{6i}\}$. Moreover, $Z_{m-1} = \{y_{6m-11}, y_{6m-10}, y_{6m-9}\}$ and $Z_m = \{y_{6m-8}, y_{6m-7}, y_{6m-6}\}$. Finally, we create a distinguished candidate p .

Votes: We create $2m+2$ votes in total. In the following, we do not distinguish between the terms “set” and “order”. In particular, U is considered as an order $(c_1, c_2, \dots, c_{3\kappa})$, and every $s = \{c_x, c_y, c_z\} \in S$ is considered as an order (c_x, c_y, c_z) with $x < y < z$. Moreover, each Z_i where $i \in \{1, \dots, m\}$ is considered as an arbitrary but fixed order of its elements.

For each $s_j \in S$ with $j = 1, 2, \dots, m-2$, we create two votes as follows.

$$\pi_{s_j} : s_j \succ p \succ Z_j \succ U \setminus s_j \succ Y \setminus Z_j$$

$$\pi'_{s_j} : \overleftarrow{U \setminus s_j} \succ Z_j \succ p \succ \overleftarrow{s_j} \succ Y \setminus Z_j$$

Note that with the above $2(m-2)$ votes, all candidates in $U \cup \{p\}$ have the same Borda score. The following four votes are created according to the last two 3-subsets $s_{m-1}, s_m \in S$.

$$\pi_{s_{m-1}} : s_{m-1} \succ p \succ Z_m \cup Z_{m-1} \succ U \setminus s_{m-1} \succ Y \setminus (Z_m \cup Z_{m-1})$$

$$\pi'_{s_{m-1}} : \overleftarrow{U \setminus s_{m-1}} \succ Z_{m-1} \succ p \succ Z_m \succ \overleftarrow{s_{m-1}} \succ Y \setminus (Z_m \cup Z_{m-1})$$

$$\pi_{s_m} : s_m \succ p \succ Z_m \cup Z_{m-1} \succ U \setminus s_m \succ Y \setminus (Z_m \cup Z_{m-1})$$

$$\pi'_{s_m} : \overleftarrow{U \setminus s_m} \succ Z_{m-1} \succ p \succ Z_m \succ \overleftarrow{s_m} \succ Y \setminus (Z_m \cup Z_{m-1})$$

With the above four votes and the previously created $2(m-2)$ votes, p has exactly 6 more points than every candidate $c \in U$.

Finally, we have two votes with preferences $U \succ Z_m \succ p \succ Y \setminus Z_m$; and $\overleftarrow{U} \succ Z_m \succ p \succ Y \setminus Z_m$, respectively.

With all the $2m+2$ votes created as above, p has exactly $3\kappa+1$ less points than every candidate $c \in U$, and all candidates in U have the same Borda score.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

In the following, let $A = \{\pi'_{s_j} \mid s_j \in S\}$ and B the set of the last two created votes. Now we discuss the correctness of the reduction.

(\Rightarrow): Suppose that \mathcal{F} is a Yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_{s_j} \mid s_j \in S'\}$ be the multiset of the votes of the first type corresponding to S' . Every vote π_{s_j} in $\Pi_{S'}$ ranks the three candidates in s_j above p . Consider the election \mathcal{E}' obtained from the original election \mathcal{E} by replacing each $\pi_{s_j} \in \Pi_{S'}$ with a vote obtained from π_{s_j} by promoting p to the highest position. Precisely, for each $\pi_{s_j} \in \Pi_{S'}$ defined as $s_j \succ p \succ Z_j \succ U \setminus s_j \succ Y \setminus Z_j$, we replace it with a vote defined as $p \succ s_j \succ Z_j \succ U \setminus s_j \succ Y \setminus Z_j$. Clearly, each replacement increases the score of p by 3, and decreases the score of every candidate in s_j by 1. Since there are exactly κ votes in $\Pi_{S'}$, the score of p is finally increased by 3κ . Since S' is an exact 3-set cover, for every $c \in U$, there is only one vote in $\Pi_{S'}$ which ranks c above p . Therefore, all replacements decrease the score of each candidate in U by 1. Since p has exactly $3\kappa+1$ less points than every candidate $c \in U$ in the original election \mathcal{E} , p has exactly the same score as every candidate $c \in U$ in the final election \mathcal{E}' . Therefore, p becomes a winner in \mathcal{E}' .

(\Leftarrow): Suppose that \mathcal{E} is a Yes-instance and $\Pi_{S'}$ is the multiset of votes which are replaced. We assume that $\Pi_{S'}$ does not contain any vote in $A \cup B$. This assumption is sound due to the following lemma.

Lemma 3 *If \mathcal{E} is a Yes-instance, there must be a solution wherein no vote in $A \cup B$ is replaced.*

PROOF. We prove the lemma by showing that it is always better to replace a vote not in $A \cup B$ than to replace a vote in $A \cup B$. Suppose that π is a vote in $A \cup B$ that is replaced. Observe that promoting p is always better than degrading candidates in U , since promoting p by one position decreases the score gap between every candidate in U and p by one, while degrading some candidate $c \in U$ by one position only decreases the score gap between c and p by one (sometimes even increases the score gap between some other candidate $c' \in U$ and p). Moreover, the amount of points that can be decreased in the score gap between every candidate in U and p by promoting p in π , can be also achieved by promoting p in any vote that is not in $A \cup B$. In fact, since in every vote in $A \cup B$ there are at least three dummy candidates ranked below some candidates in U but ranked above p , replacing votes which are not in $A \cup B$ can always

do better: replacing a vote $\pi_s \notin A \cup B$ where $s = \{c_x, c_y, c_z\}$ with preference $c_x \succ c_y \succ c_z \succ p \dots$ with a vote with preference $p \succ c_x \succ c_y \succ c_z \dots$ does not only decrease the score gap between every candidate in $U \setminus s$ and p by 3, but better yet, decreases the score gap between every candidate in s and p by 4. \square

Due to the above analysis, we assume that $\Pi_{S'}$ contains only the votes in $\{\pi_{s_j} \mid s_j \in S\}$, where S is the collection of 3-subsets in \mathcal{F} . Let $S' = \{s_j \mid \pi_{s_j} \in \Pi_{S'}\}$ be the subcollection corresponding to $\Pi_{S'}$. First observe that for any vote $\pi_s \in \Pi_{S'}$ where $s \in S$, promoting p by three positions is always better than any other combinations: by doing so, the score gap between every candidate in U and p is decreased by at least 3 (for candidates in s , the score gaps are decreased by 4). Therefore, we can assume that in the solution, every vote in $\Pi_{S'}$ is replaced with a new vote obtained from the original vote by promoting p by three positions. Since p has $3\kappa+1$ less points than every candidate in U in the original election \mathcal{E} , and we can replace at most κ votes, every candidate in U must be degraded by one position at least once. This implies that for every $c \in U$, there must be a vote $\pi_s \in \Pi_{S'}$ with $c \in s$, further implying that S' is an exact 3-set cover of \mathcal{F} .

The reduction for C-KT(3)-Borda-UNI is similar to the above reduction, with the difference in the last created vote. Precisely, we remove the last vote created in the reduction for C-KT(3)-Borda-NON, and create a vote with preference $\overleftarrow{U} \succ Z_m \cup \{y_{6m-12}\} \succ p \succ Y \setminus Z_m \cup \{y_{6m-12}\}$.

By ranking the candidate y_{6m-12} between Z_m and p , the score gap between every candidate in U and p decreases to 3κ , one point less than that in the reduction for C-KT(3)-Borda-NON. \square

Now we come to Condorcet. The C-KT(ℓ)-Condorcet-UNI problem is related to the Dodgson voting [13], where each candidate has a Dodgson score defined as the minimum number of swaps of adjacent candidates needed to make the candidate the Condorcet winner. Calculating the Dodgson score of a candidate is NP-hard [29, 30]. Recall that the KT-distance between two votes is equal to the minimum number of swaps of adjacent candidates needed to transform one into the other. Therefore, if a candidate can become the Condorcet winner by recasting at most \mathcal{R} votes with respect to KT-distance upper bound ℓ , then the Dodgson score of the candidate is at most $\mathcal{R} \cdot \ell$. In Theorem 1, we have shown that both C-KT(1)-Condorcet-UNI/NON and C-KT(2)-Condorcet-UNI/NON are polynomial-time solvable. In the following, we show that the polynomial-time solvability does not hold for C-KT(ℓ)-Condorcet-UNI/NON for every $\ell \geq 3$. Recall that in the general case, the constructive bribery for Condorcet is NP-hard [26, 30].

Hereinafter, we assume that in both the X3C problem and the X4C problem, each $c_i \in U$ occurs in exactly three subsets of S .

Theorem 4 *C-KT(ℓ)-Condorcet-UNI/NON are NP-hard for every $\ell \geq 3$.*

PROOF. We first consider C-KT(3)-Condorcet-UNI. The reduction is from the X3C problem. Given an instance $\mathcal{F} = (U = \{c_1, c_2, \dots, c_{3\kappa}\}, S = \{s_1, s_2, \dots, s_{3\kappa}\})$ of X3C, we create an instance \mathcal{E} for C-KT(3)-Condorcet-UNI as follows.

Candidates: For each $c \in U$, we create a corresponding candidate. For simplicity, we still use the same notation c to denote this candidate. In addition, we have a distinguished candidate p and a set $Y = \{y_1, y_2, y_3\}$ of three dummy candidates.

Votes: For each $s = \{c_x, c_y, c_z\} \in S$, we create a vote π_s defined as $s \succ p \succ U \setminus s \succ Y$. Here, the candidates in s , in $U \setminus s$ and in Y are

ranked according to the increasing order of the indices, respectively. In addition, we create $3\kappa - 5$ votes defined as $U \succ Y \succ p$. Here, the candidates in U and in Y are ranked according to the increasing order of the indices, respectively. In total, we have $6\kappa - 5$ votes.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

Now we discuss the correctness. Observe that c_1 is the current Condorcet winner, and no candidate in Y can become the Condorcet winner by replacing at most κ votes with respect to the distance restriction.

(\Rightarrow ;) Suppose that \mathcal{F} is a Yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_{s_j} \mid s_j \in S'\}$ be the multiset of votes corresponding to S' . Consider replacing each vote $\pi_s \in \Pi_{S'}$ by another vote obtained from π_s by promoting p to the highest position, that is, replacing each vote $\pi_s \in \Pi_{S'}$ defined as $s \succ p \succ U \setminus s \succ Y$ with a vote defined as $p \succ s \succ U \setminus s \succ Y$. Since s is a 3-subset, the KT-distance between the original vote and the new vote is 3. Since S' is an exact 3-set cover, for every $c \in U$ there is exactly one vote in $\Pi_{S'}$ which ranks c above p (and p is ranked above c after the replacement). Therefore, after κ replacements as discussed above, for every $c \in U$, there are exactly $3\kappa - 2$ votes ranking p above c , implying that p is the Condorcet winner in the final election.

(\Leftarrow ;) Suppose that \mathcal{E} is a Yes-instance and $\Pi_{S'}$ is the multiset of votes which are replaced. Since $|Y| = 3$ and each vote can be replaced only with a vote which has KT-distance at most 3 from it, replacing any of the last $3\kappa - 5$ votes is not helpful in improving the winning status of p (In other words, replacing a vote in the last $3\kappa - 5$ votes is not helpful for p to beat any candidate in U , since the dummy candidates in Y are ranked between U and p ; and thus, according to the distance restriction, p cannot be ranked above any candidate in U via a replacement of a vote in the last $3\kappa - 5$ votes.). Therefore, we know that $\Pi_{S'}$ contains only votes corresponding to S . Let $S' = \{s \in S \mid \pi_s \in \Pi_{S'}\}$ be the subcollection of S corresponding to $\Pi_{S'}$. In order to make p the Condorcet winner, for every $c \in U$ there must be at least one vote, corresponding to some s with $c \in s$, which is replaced with a vote ranking p above c . This implies that S' is an exact 3-set cover of \mathcal{F} .

The above reduction directly applies to C-KT(3)-Condorcet-NON. The NP-hardness of C-KT(4)-Condorcet-UNI/NON can be proved via reductions from the X4C problem. The reductions are analogous to the ones for C-KT(3)-Condorcet-UNI/NON (we need to create one more dummy candidate y_4 and add it to Y). The NP-hardness of C-KT(ℓ)-Condorcet-UNI/NON for every $\ell \geq 5$ is implied by the NP-hardness reductions in Theorem 3.2 in [26]. \square

Now we come to Copeland $^\alpha$. It is known that both the constructive and the destructive bribery problems without distance restrictions for Copeland $^\alpha$ are NP-hard [26], for both the unique-winner model and the nonunique-winner model. We show that these NP-hardness results hold for the distance restricted bribery problem, even when the distance bound is demanded to be a very small constant.

Theorem 5 *The C-KT(ℓ)-Copeland $^\alpha$ -UNI/NON problem, the D-KT(ℓ)-Copeland $^\alpha$ -NON problem for every $\ell \geq 3$, and the D-KT(ℓ)-Copeland $^\alpha$ -UNI problem for every $\ell \geq 5$ are NP-hard. All these results hold for every $0 \leq \alpha \leq 1$.*

PROOF. We first consider the C-KT(3)-Copeland $^\alpha$ -UNI problem. The reduction is from X3C. Let $\mathcal{F} = (U = \{c_1, c_2, \dots, c_{3\kappa}\}, S = \{s_1, s_2, \dots, s_{3\kappa}\})$ be an instance of X3C. We create an instance \mathcal{E} for C-KT(3)-Copeland $^\alpha$ -UNI as follows.

Candidates: We have $|U| + 8$ candidates in total. In particular, for each $c_i \in U$, we create a candidate. For simplicity, we still

use c_i to denote this candidate. In addition, we have 8 candidates $p, y, z_1, z_2, z_3, z'_1, z'_2, z'_3$, where p is the distinguished candidate. For ease of exposition, let $Z = \{z_1, z_2, z_3\}$ and $Z' = \{z'_1, z'_2, z'_3\}$.

Votes: Let $n = |S| = 3\kappa$. We create $2n + 1$ votes in total. In particular, for each $s = \{c_i, c_j, c_k\} \in S$, we create one vote π_s with preference $y \succ Z' \succ s \succ p \succ U \setminus s \succ Z$. Here, the candidates in $Z, Z', s, U \setminus s$ are ranked according to the increasing order of the indices, respectively. In addition, we create $n - 2$ votes each with preference $U \succ Z \succ p \succ y \succ Z'$. Finally, we create 3 votes each with preference $p \succ y \succ Z' \succ U \succ Z$. In the above $n + 1$ votes, the candidates in U, Z and Z' are ranked according to the increasing order of the indices, respectively. It is easy to verify that the candidate y is the current (unique) winner.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

Now we prove that \mathcal{F} is a Yes-instance if and only if \mathcal{E} is a Yes-instance.

(\Rightarrow ;) Suppose that \mathcal{F} is a Yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_s \mid s \in S'\}$ be the set of votes corresponding to S' . Consider the election after replacing all votes in $\Pi_{S'}$ in the following way: each $\pi_s \in \Pi_{S'}$ with $s \in S'$ is replaced with a vote defined as $y \succ Z' \succ p \succ s \succ U \setminus s \succ Z$. Clearly, the KT-distance between these two votes is 3. Since S' is an exact 3-set cover, for each $c_i \in U$ there is exactly one vote $\pi_s \in \Pi_{S'}$ with $c_i \in s$. Due to the construction, c_i is ranked above p in π_s , while ranked below p in the new vote which replaces π_s . Therefore, after κ replacements as discussed above, for every $c_i \in U$ there are $n + 1$ votes which rank p above c_i , implying that p beats every candidate $c_i \in U$, further implying that p is the unique Copeland $^\alpha$ winner (holds for every $0 \leq \alpha \leq 1$).

(\Leftarrow ;) Suppose that \mathcal{E} is a Yes-instance and $\Pi_{S'}$ is the multiset of votes which are replaced. Let \mathcal{E}' be the final election obtained from \mathcal{E} by replacing the votes in $\Pi_{S'}$ with κ many new votes (we discuss later what are the new votes). Observe that the candidate y beats every other candidate except p in \mathcal{E} . A deeper observation is that y still beats these candidates in the final election \mathcal{E}' .

Lemma 6 *The candidate y beats everyone in $U \cup Z \cup Z'$ in \mathcal{E}' .*

PROOF. Clearly, y beats every candidate in Z' in the final election \mathcal{E}' since all votes rank y above Z' . Now we consider the candidates in $U \cup Z$. Observe first that every vote in \mathcal{E} either ranks y above all candidates in $U \cup Z$, or ranks all candidates in $U \cup Z$ above y . Moreover, the votes that rank y above all candidates in $U \cup Z$ are those corresponding to S , and the last three created votes. However, in these votes, the candidates in Z' ($|Z'| = 3$) are ranked between y and every candidate in $U \cup Z$; thus, we cannot replace a vote which ranks y above a candidate $a \in U \cup Z$ by a KT(3)-close vote which, however, ranks a above y . Therefore, the votes which rank y above a candidate $a \in U \cup Z$ will still rank y above a in the final election \mathcal{E}' . The lemma follows. \square

Due to the above lemma and the fact that p is the unique winner in the final election \mathcal{E}' , we know that p beats every other candidate in \mathcal{E}' . Observe that in the original election \mathcal{E} , p is beaten by every candidate in U . Then, due to the distance restriction, $\Pi_{S'}$ must be from the votes corresponding to S . Let $S' = \{s \mid \pi_s \in \Pi_{S'}\}$ be the subcollection of 3-subsets corresponding to $\Pi_{S'}$. Since p beats all candidates in U in the final election \mathcal{E}' and we can replace at most $\mathcal{R} = \kappa$ votes, for each $c_i \in U$ there must be a vote $\pi_s \in \Pi_{S'}$ with $c_i \in s$, which is replaced with a new vote obtained from π_s by promoting p by three positions. Since $|S| = 3\kappa$, it follows that S' is an exact 3-set cover.

The above reduction applies to D-KT(3)-Copeland $^\alpha$ -NON if we set y as the distinguished candidate.

Now we consider the C-KT(3)-Copeland $^\alpha$ -NON problem. The above reduction does not apply to this case, since p does not need to beat every other candidate in the final election to become a winner (p could also become a winner even there is no exact 3-set cover). In order to overcome this situation, we introduce a new dummy candidate y' which beats p , but is beaten by y in the original election. Precisely, we adopt the votes constructed as above, and rank y' in the votes as follows: (1) rank y' immediately after y in all votes corresponding to S and all the $n - 2$ votes following; and (2) rank y' above p in all the three votes created in the last.

The NP-hardness of the C-KT(4)-Copeland $^\alpha$ -UNI/NON problem and the D-KT(4)-Copeland $^\alpha$ -NON problem can be proved via reductions from the X4C problem analogously. The NP-hardness of the C-KT(ℓ)-Copeland $^\alpha$ -UNI/NON problem and the D-KT(ℓ)-Copeland $^\alpha$ -UNI/NON problem for every $\ell \geq 5$ is implied by the NP-hardness reductions in Theorem 3.2 in [26]. \square

We have just investigated the NP-hardness of the C-KT(ℓ)-Copeland $^\alpha$ -UNI/NON and the D-KT(ℓ)-Copeland $^\alpha$ -NON problems for every $\ell \geq 3$, and the D-KT(ℓ)-Copeland $^\alpha$ -UNI problem for every $\ell \geq 5$, but left the complexity of the D-KT(ℓ)-Copeland $^\alpha$ -UNI problem for each integer $\ell \in \{3, 4\}$ open. We cannot straightforwardly adopt the reductions for the C-KT(3, 4)-Copeland $^\alpha$ -NON problem to prove the NP-hardness of the D-KT(3, 4)-Copeland $^\alpha$ -UNI problem, since both candidates y and y' win the election, and thus, no candidate is valid to be the distinguished candidate.

Now we investigate the complexity of the distance restricted bribery problems for Maximin. It is known that both the constructive and the destructive bribery problems for Maximin without distance restrictions are NP-hard [24]. We prove that the NP-hardness holds even when each bribed voter only wants to recast a new vote which has a small constant discrepancy from his original vote.

Theorem 7 *The D-KT(ℓ)-Maximin-UNI/NON problem and the C-KT(ℓ)-Maximin-UNI/NON problem are NP-hard for every $\ell \geq 4$.*

4 Hamming Distance Restricted Bribery

In this section, we study the bribery problem with Hamming distance restrictions. It should be noted that the Hamming distance between every two votes is at least 2. We begin with several polynomial-time solvability results.

Theorem 8 *The D-HAM(ℓ)-Condorcet-UNI/NON problem and the D-HAM(ℓ)-Borda-UNI/NON problem are polynomial-time solvable for every positive integer ℓ .*

PROOF. We prove the theorem by deriving polynomial-time algorithms for the problems stated in the theorem. We only describe the algorithms for the unique-winner model in detail. The algorithms for the nonunique-winner model are similar. Let m be the number of candidates, n the number of votes, \mathcal{R} the number of votes that can be replaced, and p the distinguished candidate.

D-HAM(ℓ)-Condorcet. We first consider D-HAM(2)-Condorcet. The algorithm first guesses a candidate p' which is not beaten by p in the final election. This leads to $m - 1$ subinstances, each asking whether we can make p' not be beaten by p by replacing $\mathcal{R}' \leq \mathcal{R}$ votes with \mathcal{R}' many HAM(2)-close votes. To solve each subinstance,

we need only to arbitrarily choose up to \mathcal{R} votes which rank p above p' , and replace each of them with a new vote obtained from the original vote by swapping p and p' . After this, if p' is not beaten by p , the subinstance is a Yes-instance; otherwise, the subinstance is a No-instance. It is clear that the original instance is a Yes-instance if and only if at least one of the subinstances is a Yes-instance. The above algorithm directly applies to D-HAM(ℓ)-Condorcet for every possible $\ell \geq 2$.

D-HAM(2)-Borda. The algorithm first guesses a candidate p' which prevents p from being the unique-winner in the final election. This leads to $m - 1$ subinstances, each asking whether we can make p' have an equal or greater Borda score than that of p by replacing $\mathcal{R}' \leq \mathcal{R}$ votes with \mathcal{R}' many HAM(2)-close votes. To solve each subinstance, we order all votes π_v according to a nonincreasing order of $\max\{pos_v(p') - 1, m - pos_v(p), 2 \cdot (pos_v(p') - pos_v(p))\}$. Let Π be the multiset of the first $\min\{n, \mathcal{R}'\}$ votes according to this order. Then, we replace every vote in Π in the following way. For each $\pi_v \in \Pi$, if $pos_v(p') - 1 \geq m - pos_v(p)$ and $pos_v(p') - 1 \geq 2 \cdot (pos_v(p') - pos_v(p))$, then replace π_v with a new vote obtained from π_v by swapping p' and the first ranked candidate in π_v ; otherwise, if $m - pos_v(p) \geq pos_v(p') - 1$ and $m - pos_v(p) \geq 2 \cdot (pos_v(p') - pos_v(p))$, replace π_v with a vote obtained from π_v by swapping p and the last ranked candidate in π_v ; finally, if $2 \cdot (pos_v(p') - pos_v(p)) \geq pos_v(p') - 1$ and $2 \cdot (pos_v(p') - pos_v(p)) \geq m - pos_v(p)$, replace π_v with a vote obtained from π_v by swapping p and p' . After doing this for every vote in Π , if p' has an equal or greater Borda score than that of p , the subinstance is a Yes-instance; otherwise, the subinstance is a No-instance. It is clear that the original instance is a Yes-instance if and only if at least one of the subinstances is a Yes-instance.

D-HAM(3)-Borda. The algorithm carries out $m - 1$ guesses as in the above algorithm for D-HAM(2)-Borda. So, we need only to focus on the subinstances. Notice that to prevent p from being the unique winner, we have more choices to do in this case than in D-HAM(2)-Borda. In particular, for a vote π_v , to decrease the score gap between p and the guessed candidate p' , we can either place p' in the first position, p in the $\max\{pos_v(p), pos_v(p')\}$ -th position, the first ranked candidate in π_v in the $\min\{pos_v(p), pos_v(p')\}$ -th position, or we can place p in the last position, p' in the $\min\{pos_v(p), pos_v(p')\}$ -th position, the last ranked candidate in π_v in the $\max\{pos_v(p), pos_v(p')\}$ -th position. Let $sga(\pi_v)$ be the amount of the decrease of the score gap between p and p' if we perform the first operation, and $sgb(\pi_v)$ the amount of decrease of the score gap between p and p' if we perform the second operation. Then, to solve each subinstance, we order the votes according to a nonincreasing order of $\max\{sga(\pi_v), sgb(\pi_v)\}$ and recast the votes accordingly, as discussed above.

D-HAM(4)-Borda. Similar to the above algorithms, the algorithm for D-HAM(4)-Borda first carries out $m - 1$ guesses, leading to $m - 1$ subinstances. Now we restrict our attention to these subinstances. For each subinstance, we partition the votes into two multisubsets Π_1 and Π_2 , where Π_1 includes all votes that rank p above p' , and Π_2 includes all votes that rank p' above p . Then, we order the votes in Π_1 according to a nonincreasing order of $pos(p') - pos(p)$. Moreover, we choose the first $\min\{|\Pi_1|, \mathcal{R}'\}$ votes, and replace each of them with a vote obtained from the original vote by swapping p' and the first ranked candidate, and swapping p and the last ranked candidate. After doing so, if p' has an equal or greater score than that of p , we return "Yes". Otherwise, if p' still has a less score than that of p , we distinguish between two cases. If $|\Pi_1| \geq \mathcal{R}'$, we return "No"

immediately. In the case that $|\Pi_1| < \mathcal{R}$, we order the votes in Π_2 according to a nondecreasing order of $\text{pos}(p) - \text{pos}(p')$. Then, we choose the first $\mathcal{R} - |\Pi_1|$ votes, and replace each of them with a vote obtained from the original vote by swapping p' and the first ranked candidate, and swapping p with the last ranked candidate. After doing this, if p' has an equal or greater score than that of p , the subinstance is a Yes-instance; otherwise it is a No-instance.

D-HAM(ℓ)-Borda. The algorithm for D-HAM(ℓ)-Borda with $\ell > 4$ is exactly the same as for D-HAM(4)-Borda. \square

Now we show our hardness results. We begin with the distance restricted bribery problem for Copeland $^\alpha$.

Theorem 9 *The C-HAM(2)-Copeland $^\alpha$ -UNI/NON problem and the D-HAM(2)-Copeland $^\alpha$ -UNI/NON problem are NP-hard for every $0 \leq \alpha \leq 1$.*

PROOF. Due to space limitation, we give only the NP-hardness proof for C-HAM(2)-Copeland $^\alpha$ -UNI. Our reduction is from the X3C problem. Let $\mathcal{F} = (U = \{c_1, c_2, \dots, c_{3\kappa}\}, S = \{s_1, s_2, \dots, s_{3\kappa}\})$ be a given instance of the X3C problem. We create an instance \mathcal{E} for C-HAM(2)-Copeland $^\alpha$ -UNI as follows.

Candidates: We create $3\kappa + 2$ candidates in total. In particular, for each element $c_i \in U$, we create one candidate. For convenience, we still use c_i to denote the corresponding candidate. In addition, we have two candidates p and q with p being the distinguished candidate.

Votes: For each $s \in S$, we create a vote π_s with preference $q \succ U \setminus s \succ p \succ s$. In addition, we create $\kappa - 1$ votes with preference $p \succ q \succ U$, and two votes with preference $U \succ p \succ q$. In total, we have $4\kappa + 1$ votes. The comparisons between every two candidates are summarized in Table 2. It is easy to verify that q beats every other candidate; and thus, q is the current unique winner.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

	q	p	c_j
q	-	3κ	$4\kappa - 1$
p	$\kappa + 1$	-	$\kappa + 2$
c_i	2	$3\kappa - 1$	\dots

Table 2. Comparisons between every two candidates in the NP-hardness reduction for C-HAM(2)-Copeland $^\alpha$ -UNI in Theorem 9. The comparisons between c_i and c_j for $i \neq j$ do not play any role in the correctness argument.

Now we prove the correctness of the reduction.

(\Rightarrow): Suppose that \mathcal{F} is a Yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_s \mid s \in S'\}$ be the set of votes corresponding to S' . Consider the final election \mathcal{E}' obtained from \mathcal{E} by replacing each vote π_s where $s \in S$ with a vote obtained from π_s by swapping p and q . More precisely, each $\pi_s \in \Pi_{S'}$ with preference $q \succ U \setminus s \succ p \succ s$ is replaced with a vote with preference $p \succ U \setminus s \succ q \succ s$. Clearly, the Hamming distance between these two votes is two. Moreover, we have that $N_{\mathcal{E}'}(p, q) = 2\kappa + 1$. Now we consider the comparison between p and every $c_i \in U$. Since S' is an exact 3-set cover, for every c_i there are exactly $\kappa - 1$ votes $\pi_s \in \Pi_{S'}$ with $c_i \notin s$. All these votes rank c_i above p in \mathcal{E} . However, these votes are replaced with $\kappa - 1$ votes which rank p above c_i as discussed above, in the final election \mathcal{E}' . Therefore, for every $c_i \in U$, there are $(\kappa + 2) + (\kappa - 1) = 2\kappa + 1$ votes which rank p above c_i , implying that p beats every $c_i \in U$ in \mathcal{E}' . Therefore, p becomes the unique winner in \mathcal{E}' .

(\Leftarrow): Suppose that \mathcal{E} is a Yes-instance. Let \mathcal{E}' be the final election obtained from \mathcal{E} by replacing at most κ votes. Since $N_{\mathcal{E}}(q, c_i) =$

$4\kappa - 1$, we know that q beats every candidate $c_i \in U$ in \mathcal{E}' . As a result, q is beaten by p in \mathcal{E}' , since otherwise, q would beat every other candidate in the final election \mathcal{E}' , and thus, remains the unique winner. Moreover, since p is the unique winner in \mathcal{E}' , p must beat every other candidate in the final election \mathcal{E}' . Since $N_{\mathcal{E}}(p, q) = \kappa + 1$, in order to make p beat q , there has to be κ votes ranking q above p that are replaced by κ new votes ranking p above q . Due to this, we know that the replaced votes are from the votes corresponding to S , since any other vote has already ranked p above q . Let $\Pi_{S'}$ be the replaced votes, and $S' = \{s \mid \pi_s \in \Pi_{S'}\}$ the subcollection of 3-subsets corresponding to $\Pi_{S'}$. As discussed above, p beats every candidate $c_i \in U$ in \mathcal{E}' . Since $N_{\mathcal{E}}(p, c_i) = \kappa + 2$, for every $c_i \in U$, there must be at least $\kappa - 1$ votes in $\Pi_{S'}$ ranking c_i above p that are replaced by $\kappa - 1$ votes ranking p above c_i . This happens only if S' is an exact 3-set cover. \square

For Condorcet and Maximin, we have the following results.

Theorem 10 *C-HAM(2)-Condorcet-UNI/NON are NP-hard.*

Theorem 11 *The D-HAM(2)-Maximin-UNI/NON problem and the D-HAM(2)-Maximin-UNI/NON problem are NP-hard.*

5 Conclusion

We have studied the complexity of the distance restricted bribery problem which differs from the traditional bribery problem [22] in that the bribed voters only recast new votes which are “close” to their original votes. In particular, we adopted the Hamming distance and the KT-distance to measure the closeness between two votes. We achieved both polynomial-time solvability results and NP-hardness results for Borda, Condorcet, Maximin and Copeland $^\alpha$. In particular, we achieved dichotomy results for Condorcet. A primary conclusion of our findings is that the constructive distance restricted bribery problem is generally NP-hard even when the distance is bounded by a very small integer (this holds for all cases studied in this paper except the Hamming distance restricted bribery problem for Borda, whose complexity remains open for every distance $\ell \geq 2$). On the other hand, there exist voting systems where the constructive distance restricted bribery problem is polynomial-time solvable, when the distance is bounded by 1 or 2. For the destructive distance restricted bribery problem, it turned out that for Maximin and Copeland $^\alpha$ it has become NP-hard even when the distance is bounded by 2. On the other hand, for many other cases, it is polynomial-time solvable for every possible distance bound. As the bribery problem proposed in [22] can be considered as a distance restricted bribery problem with the distance bound being considerably large ($m(m - 1)/2$ in KT-distance restriction and m in Hamming distance restriction, where m is the number of candidates), our work complements the complexity results of the bribery problem obtained in [11, 24, 26]. Of particular importance is that our work pinpoints the complexity border between polynomial-time solvability and NP-hardness of distance restricted bribery problem, with respect to the distance bound. See Table 1 for a summary of our results.

There remain several open problems as shown in Table 1. For example, we do not know the complexity of the D-KT(1)-Copeland $^\alpha$ -UNI/NON problem. Another avenue of research would be to explore these problems from the parameterized complexity viewpoint. Furthermore, exploring the same problems with respect to further distance measurements (see [14, 16, 33] for several distance measurements on linear orders) is also an interesting direction for future research.

REFERENCES

- [1] O. L. Astrachan, 'Bubble sort: An archaeological algorithmic analysis', in *SIGCSE*, pp. 1–5, (2003).
- [2] M. S. Bansal and D. Fernández-Baca, 'Computing distances between partial rankings', *Inf. Process. Lett.*, **109**(4), 238–241, (2009).
- [3] R. Berghammer and H. Schnoor, 'Control of Condorcet voting: Complexity and a relation-algebraic approach', *Eur. J. Oper. Res.*, **246**(2), 505–516, (2015).
- [4] N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier, 'Studies in computational aspects of voting – A parameterized complexity perspective', in *The Multivariate Algorithmic Revolution and Beyond*, pp. 318–363, (2012).
- [5] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond, 'Fixed-parameter algorithms for Kemeny rankings', *Theor. Comput. Sci.*, **410**(45), 4554–4570, (2009).
- [6] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond, 'How similarity helps to efficiently compute Kemeny rankings', in *AAMAS (1)*, pp. 657–664, (2009).
- [7] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas, 'Link analysis ranking: Algorithms, theory, and experiments', *ACM Trans. Internet Techn.*, **5**(1), 231–297, (2005).
- [8] S. J. Brams, D. M. Kilgour, and M. R. Sanver, 'A minimax procedure for electing committees', *Public. Choice.*, **132**(3-4), 401–420, (2007).
- [9] F. J. Brandenburg, A. Gleißner, and A. Hofmeier, 'Comparing and aggregating partial orders with Kendall-Tau distances', *Discrete Math., Alg. and Appl.*, **5**(2), (2013).
- [10] R. Bredereck, J. Chen, and G. J. Woeginger, 'Are there any nicely structured preference profiles nearby?', *Math. Soc. Sci.*, **79**, 61–73, (2016).
- [11] E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor, and I. Schnoor, 'Approximability of manipulating elections', in *AAAI*, pp. 44–49, (2008).
- [12] E. Chávez, K. Figueroa, and G. Navarro, 'Effective proximity retrieval by ordering permutations', *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(9), 1647–1658, (2008).
- [13] C. L. Dodgson, *A Method for Taking Votes on More than Two Issues*, Clarendon Press, 1876.
- [14] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, 'Rank aggregation methods for the web', in *WWW*, pp. 613–622, (2001).
- [15] P. J. Egan, 'Do something politics and double-peaked policy preferences', *J. Polit.*, **76**(2), 333–349, (2014).
- [16] E. Elkind, P. Faliszewski, and A. Slinko, 'Distance rationalization of voting rules', *Soc. Choice. Welfare.*, **45**(2), 345–377, (2015).
- [17] E. Elkind, P. Faliszewski, and A. M. Slinko, 'Swap bribery', in *SAGT*, pp. 299–310, (2009).
- [18] G. Erdélyi, M. Lackner, and A. Pfandler, 'Computational aspects of nearly single-peaked electorates', in *AAAI*, pp. 283–289, (2013).
- [19] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee, 'Comparing and aggregating rankings with ties', in *PODS*, pp. 47–58, (2004).
- [20] R. Fagin, R. Kumar, and D. Sivakumar, 'Comparing top k lists', *SIAM J. Discrete Math.*, **17**(1), 134–160, (2003).
- [21] P. Faliszewski, 'Nonuniform bribery', in *AAMAS (3)*, pp. 1569–1572, (2008).
- [22] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra, 'The complexity of bribery in elections', in *AAAI*, pp. 641–646, (2006).
- [23] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra, 'How hard is bribery in elections?', *J. Artif. Intell. Res. (JAIR)*, **35**, 485–532, (2009).
- [24] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra, 'Multimode control attacks on elections', *J. Artif. Intell. Res. (JAIR)*, **40**, 305–351, (2011).
- [25] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra, 'The complexity of manipulative attacks in nearly single-peaked electorates', *Artif. Intell.*, **207**, 69–99, (2014).
- [26] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe, 'Lull and Copeland voting computationally resist bribery and constructive control', *J. Artif. Intell. Res. (JAIR)*, **35**, 275–341, (2009).
- [27] T. F. Gonzalez, 'Clustering to minimize the maximum intercluster distance', *Theor. Comput. Sci.*, **38**, 293–306, (1985).
- [28] R. W. Hamming, 'Error detecting and error correcting codes', *AT & T Tech. J.*, **26**(2), 147–160, (1950).
- [29] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe, 'Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP', *J. ACM.*, **44**(6), 806–825, (1997).
- [30] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick, 'Voting schemes for which it can be difficult to tell who won the election', *Soc. Choice. Welfare.*, **6**(2), 157–165, (1989).
- [31] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick, 'How hard is it to control an election?', *Math. Comput. Model.*, **16**(8-9), 27–40, (1992).
- [32] M. Kalech, S. Kraus, G. A. Kaminka, and C. V. Goldman, 'Practical voting rules with partial information', *Auton. Agent. Multi-AG.*, **22**, 151–182, (2011).
- [33] M. G. Kendall, 'A new measure of rank correlation', *Biometrika.*, **30**(1/2), pp. 81–93, (1938).
- [34] S. Konieczny, J. Lang, and P. Marquis, 'DA² merging operators', *Artif. Intell.*, **157**(1-2), 49–79, (2004).
- [35] J. Lang, G. Pigozzi, M. Slavkovik, and L. van der Torre, 'Judgment aggregation rules based on minimization', in *TARK*, pp. 238–246, (2011).
- [36] A. P. Lin, 'The complexity of manipulating k -Approval elections', in *ICAART (2)*, pp. 212–218, (2011). <http://arxiv.org/abs/1005.4159>.
- [37] A. Lumini and L. Nanni, 'Detector of image orientation based on Borda count', *Pattern. Recongn. Lett.*, **27**(3), 180–186, (2006).
- [38] N. Mattei, M. S. Pini, F. Rossi, and K. B. Venable, 'Bribery in voting with CP-nets', *Ann. Math. Artif. Intell.*, **68**(1-3), 135–160, (2013).
- [39] R. Meir, A. D. Procaccia, J. S. Rosenschein, and A. Zohar, 'Complexity of strategic behavior in multi-winner elections', *J. Artif. Intell. Res. (JAIR)*, **33**, 149–178, (2008).
- [40] M. K. Miller and D. N. Osherson, 'Methods for distance-based judgment aggregation', *Soc. Choice. Welfare.*, 575–601, (2009).
- [41] S. Obraztsova and E. Elkind, 'Optimal manipulation of voting rules', in *AAMAS*, pp. 619–626, (2012).
- [42] M. S. Pini, F. Rossi, and K. B. Venable, 'Bribery in voting with soft constraints', in *AAAI*, pp. 803–809, (2013).
- [43] Y. Yang, 'Manipulation with bounded single-peaked width: A parameterized study', in *AAMAS*, pp. 77–85, (2015).
- [44] Y. Yang and J. Guo, 'The control complexity of r -Approval: from the single-peaked case to the general case', in *AAMAS*, pp. 621–628, (2014).
- [45] Y. Yang and J. Guo, 'Controlling elections with bounded single-peaked width', in *AAMAS*, pp. 629–636, (2014).
- [46] Y. Yang and J. Guo, 'How hard is control in multi-peaked elections: A parameterized study', in *AAMAS*, pp. 1729–1730, (2015).

Beyond IC Postulates: Classification Criteria for Merging Operators

Adrian Haret¹ and Andreas Pfandler^{1,2} and Stefan Woltran¹

Abstract. Merging is one of the central operations in the field of belief change, which is concerned with aggregating the opinions of individuals. Representation theorems provide a family of merging operators satisfying some natural desiderata for merging beliefs. However, little is known about how these operators can be further distinguished. In the field of social choice, on the other hand, numerous properties have been proposed in order to classify voting rules. In this work, we adapt these properties to the context of merging and investigate how they relate to the standard postulates. Our results thus lead to a more fine-grained classification of merging operators and shed light on the question of which particular merging operator is best suited in a concrete application domain.

1 Introduction

Belief merging studies methods for aggregating the opinions of individuals into a theory which captures the consensus of the agents involved. The standard approach in the literature focuses on the design of merging operators satisfying a set of normative properties. Consensus is then obtained as the theory which comes as close as possible to the agents' expressed beliefs, subject to the limitations expressed by the normative properties [13, 14]. In the field of belief merging the variety of measures of closeness used gives rise via representation theorems to a variety of merging operators with desirable properties. Belief merging differs from voting, as analyzed in (computational) social choice theory (for an overview see, e.g., [1, 16]), in that it does not require the agents to provide full rankings of the alternatives, but only to encode their first choices as logical theories. However, belief merging and voting still share a common goal and methodology, and it is natural to conclude that the two fields can be usefully brought to bear on each other.

One direction of research views voting as a merging task [5, 10], an approach which fits into the larger program of finding suitable logics in which to represent preferences and embed aggregation problems stemming from (computational) social choice [2, 6]. A different approach, which we follow here, looks at merging from a voting perspective and uses the rich set of criteria developed to analyze voting rules in order to classify existing merging operators. Surprisingly, this line of research has received little attention so far. Apart from some interest in strategy-proofness and connections with Arrow's theorem [3, 7, 12, 15], the only other social choice properties that have made their way in the literature on merging are the egalitarian properties discussed in [8]. Notwithstanding, the social choice literature on voting features many other properties whose ideas are relevant in the context of merging, but which have hitherto been left un-addressed. We aim

to fill this gap and investigate ways of looking at merging operators that improve upon the fundamental classification into majority and arbitration operators.

Our contribution lies, first of all, in proposing fourteen new properties for merging operators, obtained, mostly, by translating existing properties from the voting literature. In doing so, we contribute to a deeper understanding of merging as a social process, by exploiting a natural analogy between voting and belief merging. Thus, theories to be merged are voters, the merging operator is the voting rule, and interpretations of the propositional variables are the candidates;³ we also allow for the possibility of a constraint, which limits the range of possible results. In keeping with the merging literature, we take merging operators to be characterized by a core set of properties, known as the IC postulates [13, 14]. Our new properties are meant to extend this characterization by offering more fine-grained criteria for evaluating merging operators. We group the properties according to their character, and offer discussions on the behavior they are intended to model. Second, in the case of each new property, we study its relationship with the core set of IC postulates. When a property is not guaranteed by the IC postulates, we investigate which of the standard operators satisfy the property, give relevant counter-examples, and provide model-based representation results for the most prominent of these properties.

The motivation for proposing new properties is the same as the motivation behind the original IC postulates: we are interested in merging operators that are syntax independent, fair and that respond in expected ways to changes in the input, and we want general principles that capture these properties. Our claim, backed up by the voting literature, is that there are many ways of making these intuitions precise, some of which go beyond the core set of IC postulates.

2 Background

Propositional logic. We work with the language \mathcal{L} of propositional logic over a fixed *alphabet* $\mathcal{P} = \{p_1, \dots, p_n\}$ of propositional atoms. An interpretation is a set $w \subseteq \mathcal{P}$ of atoms, with the intended meaning that atom p is contained in w if the truth value of p is set to true. The set of all interpretations over \mathcal{P} is denoted by \mathcal{W} . We will often represent an interpretation by its corresponding bit-vector of length $|\mathcal{P}|$ (e.g., 101 is the interpretation $\{p_1, p_3\}$). If interpretation w satisfies formula φ , we call w a *model* of φ . We denote the set of models of φ by $[\varphi]$. A pre-order \leq on \mathcal{W} is a binary relation on \mathcal{W} which is reflexive and transitive. We denote by $w_1 < w_2$ the strict part of \leq , i.e., $w_1 \leq w_2$

³ We make an exception to this and treat propositional atoms as candidates when interpretations cannot be reliably seen to fulfil this role. Though this introduces an ambiguity in our notion of "candidate", we view it as a useful step to take in order to capture more voting properties than would otherwise be possible.

¹ Institute of Information Systems, TU Wien, Austria

² School of Economic Disciplines, University of Siegen, Germany

but $w_2 \not\leq w_1$. We write $w_1 \approx w_2$ to abbreviate $w_1 \leq w_2$ and $w_2 \leq w_1$. If \mathcal{M} is a set of interpretations, then the set of *minimal elements* of \mathcal{M} with respect to \leq is defined as $\min_{\leq} \mathcal{M} = \{w_1 \in \mathcal{M} \mid \nexists w_2 \in \mathcal{M} \text{ s.t. } w_2 \leq w_1, w_1 \not\leq w_2\}$. By a renaming ρ of the variables we understand a permutation of their names. A renaming ρ applied to any formula, knowledge base or profile changes the propositional variables in it according to ρ . For instance, if ρ swaps only variables p_1 and p_2 among them and $\varphi = p_1 \wedge \neg p_2 \wedge p_3$, then $\rho(\varphi) = p_2 \wedge \neg p_1 \wedge p_3$. We also extend here the notion of renaming to apply to interpretations. Thus, if ρ swaps p_1 and p_2 between them in a formula, then ρ applied to an interpretation swaps the first and second bits in the bit-vector representation. For instance, $\rho(101) = 011$. A transposition τ is a renaming that swaps exactly two elements among each other.

Belief Merging. A *knowledge base* is a finite set of propositional formulas over \mathcal{L} . A *profile* is a non-empty finite tuple $E = \langle K_1, \dots, K_n \rangle$ of consistent, but not necessarily mutually consistent knowledge bases. We denote by \mathcal{E} (resp. \mathcal{K}) the set of all profiles (resp. knowledge bases) over \mathcal{L} . If E_1 and E_2 are profiles, then $E_1 \sqcup E_2$ is the concatenation of E_1 and E_2 . Interpretation w is a model of $K \in \mathcal{K}$ (resp. $E \in \mathcal{E}$) if it is a model of every element in K (resp. E). We denote by $[K]$ and $[E]$ the set of models of K and E , respectively. We write $\bigwedge E$ for $\bigwedge_{K \in E} \bigwedge_{\varphi \in K} \varphi$, $\neg K$ for $\neg \bigwedge K$, and $\neg E$ for $\langle \neg K_1, \dots, \neg K_n \rangle$. Profiles E_1 and E_2 are *equivalent*, written $E_1 \equiv E_2$, if there exists a bijection $f: E_1 \rightarrow E_2$ such that for any $K \in E_1$ we have $[K] = [f(K)]$.

A merging operator is a function $\Delta: \mathcal{E} \times \mathcal{L} \rightarrow \mathcal{K}$, and we write $\Delta_\mu(E)$ instead of $\Delta(E, \mu)$. The formula μ is called *the constraint* and it encodes an external condition which needs to hold in the final result regardless of the input knowledge bases. It can be thought of as a set of legal requirements or limits of feasibility restricting the outcomes of the merging process. Next, logical postulates set out properties which any merging operator Δ should satisfy. An operator satisfying the following postulates is called an IC merging operator [13, 14]:

- (IC₀) $\Delta_\mu(E) \models \mu$
- (IC₁) If μ is consistent, then $\Delta_\mu(E)$ is consistent
- (IC₂) If $\bigwedge E$ is consistent with μ , then $\Delta_\mu(E) \equiv \bigwedge E \wedge \mu$
- (IC₃) If $E_1 \equiv E_2$ and $\mu_1 \equiv \mu_2$, then $\Delta_{\mu_1}(E_1) \equiv \Delta_{\mu_2}(E_2)$
- (IC₄) If $K_1 \models \mu$ and $K_2 \models \mu$, then $\Delta_\mu(\langle K_1, K_2 \rangle) \wedge K_1$ is consistent iff $\Delta_\mu(\langle K_1, K_2 \rangle) \wedge K_2$ is consistent
- (IC₅) $\Delta_\mu(E_1) \wedge \Delta_\mu(E_2) \models \Delta_\mu(E_1 \sqcup E_2)$
- (IC₆) If $\Delta_\mu(E_1) \wedge \Delta_\mu(E_2)$ is consistent, then $\Delta_\mu(E_1 \sqcup E_2) \models \Delta_\mu(E_1) \wedge \Delta_\mu(E_2)$
- (IC₇) $\Delta_{\mu_1}(E) \wedge \mu_2 \models \Delta_{\mu_1 \wedge \mu_2}(E)$
- (IC₈) If $\Delta_{\mu_1}(E) \wedge \mu_2$ is consistent, then $\Delta_{\mu_1 \wedge \mu_2}(E) \models \Delta_{\mu_1}(E) \wedge \mu_2$

Though these postulates lay out what properties $\Delta_\mu(E)$ should have, they do not spell out how to actually construct $\Delta_\mu(E)$, given E and μ . To this end it is useful to focus on so-called *assignments* that map any $E \in \mathcal{E}$ to a pre-order \leq_E on \mathcal{W} . We say that such an assignment *represents* a merging operator Δ if $[\Delta_\mu(E)] = \min_{\leq_E} [\mu]$, for any $E \in \mathcal{E}$ and $\mu \in \mathcal{L}$. Konieczny and Pino Pérez [13] have defined the central notion of *syncretic assignments*.

Definition 1. A *syncretic assignment* is a function mapping every $E \in \mathcal{E}$ to a total pre-order \leq_E on \mathcal{W} such that, for any $E, E_1, E_2 \in \mathcal{E}$, $K_1, K_2 \in \mathcal{K}$ and $w_1, w_2 \in \mathcal{W}$ the following conditions hold:

- (s₁) If $w_1 \in [E]$ and $w_2 \in [E]$, then $w_1 \approx_E w_2$.
- (s₂) If $w_1 \in [E]$ and $w_2 \notin [E]$, then $w_1 <_E w_2$.

- (s₃) If $E_1 \equiv E_2$, then $\leq_{E_1} = \leq_{E_2}$.
- (s₄) If $w_1 \in [K_1]$, then there is $w_2 \in [K_2]$ s.t. $w_2 \leq_{\{K_1, K_2\}} w_1$.
- (s₅) If $w_1 \leq_{E_1} w_2$ and $w_1 \leq_{E_2} w_2$, then $w_1 \leq_{E_1 \sqcup E_2} w_2$.
- (s₆) If $w_1 \leq_{E_1} w_2$ and $w_1 <_{E_2} w_2$, then $w_1 <_{E_1 \sqcup E_2} w_2$.

The classical result below characterizes all IC merging operators in terms of syncretic assignments.

Theorem 1. A merging operator Δ is an IC merging operator iff there is a syncretic assignment which represents it.

Specifying concrete merging operators is usually done via a notion of distance (that induces pre-orders \leq_{K_i}) and an aggregation function (which combines the individual rankings \leq_{K_i} into a final pre-order \leq_E). More precisely: a *pseudo-distance* is a function $d: \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}_+$ such that, for any $w_1, w_2 \in \mathcal{W}$, (i) $d(w_1, w_2) = d(w_2, w_1)$ and (ii) $d(w_1, w_2) = 0$ if and only if $w_1 = w_2$. An *aggregation function* is a function f such that, for any $x_1, \dots, x_n, x, y \in \mathbb{R}_+$ and any permutation π , (i) if $x \leq y$, then $f(x_1, \dots, x_n, x, y) \leq f(x_1, \dots, y, \dots, x_n)$, (ii) $f(x_1, \dots, x_n) = 0$ if and only if $x_1 = \dots = x_n = 0$, (iii) $f(x) = x$, and (iv) $f(x_1, \dots, x_n) = f(\pi(x_1), \dots, \pi(x_n))$.

The *Hamming distance* between interpretations w and w' is defined as $d_H(w, w') = |(w \setminus w') \cup (w' \setminus w)|$; the *drastic distance* between w and w' is given as $d_D(w, w') = 0$ if $w = w'$ and 1 otherwise. The minimal distance between interpretations and models of K_i yields \leq_{K_i} . The distance $d(w, K_i)$ between w and K_i is computed by taking the minimal distance between w and all $w' \in [K_i]$. Now, the pre-order \leq_{K_i} is defined by saying that $w \leq_{K_i} w'$ if $d(w, K_i) \leq d(w', K_i)$. For aggregating the rankings, common functions are summation (Σ), *GMAX* and *GMIN*. For d_H this gives us the operators $\Delta^{d_H, \Sigma}$, $\Delta^{d_H, GMAX}$ and $\Delta^{d_H, GMIN}$. These operators are all distinct, in the sense that they may give different results on the same input. On the other hand, operators defined using the drastic distance are all equivalent, in the sense that $\Delta_\mu^{d_D, \Sigma}(E) \equiv \Delta_\mu^{d_D, GMAX}(E) \equiv \Delta_\mu^{d_D, GMIN}(E)$, for any $E \in \mathcal{E}$ and $\mu \in \mathcal{L}$. We thus denote these operators by Δ^{d_D} . In general, we will write Δ^{d_H} and Δ^{d_D} when our results hold with all of the three aggregation functions presented. For details, see [13, 14].

Example 1. Consider three reviewers who are part of a conference committee. They have to arrive at a decision concerning three papers they have been assigned, in a process that requires combining their individual (perhaps mutually inconsistent) beliefs about which of the papers should be accepted or rejected. The acceptance of each paper is represented by a propositional atom: p_i means that paper i is accepted, for $i \in \{1, 2, 3\}$. The opinions of the three reviewers are encoded by three knowledge bases, as follows: $K_1 = \{p_1 \wedge p_2 \wedge \neg p_3\}$, $K_2 = \{\neg p_1 \wedge \neg p_2\}$, $K_3 = \{p_1 \wedge p_3\}$. In other words, Reviewer 1 thinks only Papers 1 and 2 should be accepted, Reviewer 2 thinks Papers 1 and 2 should be rejected but has no stated opinion on Paper 3, and Reviewer 3 thinks Papers 1 and 3 should be accepted. Additionally, the rule for their committee is that not all papers can be accepted. This rule can be encoded by the constraint $\mu = \neg(p_1 \wedge p_2 \wedge p_3)$.

Thus, if $E = \langle K_1, K_2, K_3 \rangle$ is the profile, the task is to compute $\Delta_\mu(E)$. We illustrate the operators $\Delta^{d_H, \Sigma}$, $\Delta^{d_H, GMAX}$, $\Delta^{d_H, GMIN}$ discussed above. First we compute a pre-order \leq_{K_i} on \mathcal{W} for each K_i based on the distance $d(w, K_i)$. In our example (using Hamming distance d_H), we obtain $d(010, K_3) = \min\{d_H(010, w') \mid w' \in [K_3]\} = \min\{d_H(010, 101), d_H(010, 111)\} = \min\{3, 2\} = 2$. The complete set of distances is featured in Table 1. The next step is to combine the pre-orders \leq_{K_i} into a new pre-order, reflecting

w	K_1	K_2	K_3	Σ	$GMAX$	$GMIN$
000	2	0	2	4	(2,2,0)	(0,2,2)
001	3	0	1	4	(3,1,0)	(0,1,3)
010	1	1	2	4	(2,1,1)	(1,1,2)
011	2	1	1	4	(2,1,1)	(1,1,2)
100	1	1	1	3	(1,1,1)	(1,1,1)
101	2	1	0	3	(2,1,0)	(0,1,2)
110	0	2	1	3	(2,1,0)	(0,1,2)
111	1	2	0	3	(2,1,0)	(0,1,2)

Table 1: Distances and aggregated values for Example 1

the consensus opinion. We use an aggregation function (in this case Σ , $GMAX$ and $GMIN$) to obtain the final ranking \leq_E . The aggregation function Σ adds the numbers interpretation-wise, and the final ranking \leq_E^Σ is determined by the order of the final levels for each interpretation. The aggregation functions $GMAX$ and $GMIN$ order the vector of levels for each interpretation in descending and ascending order, respectively. Then we determine \leq_E^{GMAX} and \leq_E^{GMIN} by ordering the vectors lexicographically. Finally, we pick from the models of μ (highlighted in grey in Table 1) the ones with minimal levels in the final ranking: $[\Delta_\mu^{d_H, \Sigma}(E)] = \{100, 101, 110\}$, $[\Delta_\mu^{d_H, GMAX}(E)] = \{100\}$, $[\Delta_\mu^{d_H, GMIN}(E)] = \{101, 110\}$.

We can now interpret this result back in propositional logic. For instance, $\Delta_\mu^{d_H, \Sigma}(E) \equiv \{p_1 \wedge \neg(p_2 \wedge p_3)\}$, thus saying that Paper 1 should be accepted, but Papers 2 and 3 cannot be accepted together. Notice that this result is not resolute, as it does not tell which of Papers 2 or 3 should be accepted, if any. On the other hand, $\Delta_\mu^{d_H, GMAX}(E) \equiv \{p_1 \wedge \neg p_2 \wedge \neg p_3\}$, thus saying that only Paper 1 should be accepted.

In the next section we will revisit this example several times to illustrate and clarify the definitions of the properties.

Voting Theory. Let C be a finite set of candidates with $|C| = m$ and $V = \{1, 2, \dots, n\}$ be a finite set of voters. The preference of a voter is modelled as a total order over C , the vote \succ . The top-ranked candidate of \succ is at position 1, the successor at position 2, \dots , and the last-ranked candidate is at position m . A collection of preference relations $\mathcal{P} = \langle \succ_1, \dots, \succ_n \rangle$ is called a *preference profile*. A voter i prefers candidate c over candidate c' if $c \succ_i c'$. An election is given by $E = (C, V, \mathcal{P})$. A *voting correspondence* \mathcal{F} is a mapping from an election E to a non-empty subset of the candidates $W \subseteq C$, i.e., the *winners* of the election. We denote a preference profile comprising pre-orders instead of total orders by $\langle \leq_1, \dots, \leq_n \rangle$. For more details, see [1] and [16, in particular Chapter 4].

3 Properties for Belief Merging

In this section we present a number of natural properties for belief merging, several of which stem from the (computational) social choice literature, where they are typically applied to voting procedures. We group these properties according to common themes of interest in the Knowledge Representation literature.

Syntax Independence

These properties require that the outcome does not depend on how knowledge is encoded. In other words, the concrete syntactic formulation of the profile should not affect the result of the merging process. Note that IC_3 already ensures syntax independence to some degree. However, not all properties defined here are implied by IC_3 and hence are more restrictive.

Anonymity. A voting system satisfies anonymity if the winner cannot be changed by permuting the votes in the profile. In a merging scenario, we denote by $\pi(E) = \langle K_{\pi(1)}, \dots, K_{\pi(n)} \rangle$ the profile obtained by changing the order of the knowledge bases in E in accordance with a permutation $\pi: \{1, \dots, n\} \rightarrow \{1, \dots, n\}$. Anonymity is then defined as follows:⁴

$$(\text{Anonymity}) \quad \Delta_\mu(E) \equiv \Delta_\mu(\pi(E)).$$

In Example 1, Anonymity requires that $\Delta^{d_H, \Sigma}$, $\Delta^{d_H, GMAX}$ and $\Delta^{d_H, GMIN}$ produce the same result, respectively, when the profile is $\langle K_1, K_2, K_3 \rangle$, or $\langle K_2, K_1, K_3 \rangle$, or is any other permutation of the knowledge bases. One can see that Anonymity is satisfied by the operators mentioned, as the final result does not depend on the order in which the pre-orders are aggregated.

Neutrality. In a voting scenario, neutrality requires that if two candidates are swapped in all votes, then they are also swapped in the result. The purpose is to ensure that all candidates are treated equally in the determination of the winners, i.e., their name does not matter. In a merging scenario, we have to enforce that renaming variables does not affect the merging outcome. We define neutrality as follows:

$$(\text{Neutrality}) \quad \rho(\Delta_\mu(E)) \equiv \Delta_{\rho(\mu)}(\rho(E)).$$

Under the renaming ρ that swaps p_1 and p_2 among them, the knowledge bases and constraint from Example 1 become: $\rho(K_1) = \{p_2 \wedge p_1 \wedge \neg p_3\}$, $\rho(K_2) = \{\neg p_2 \wedge \neg p_1\}$, $\rho(K_3) = \{p_2 \wedge p_3\}$ and $\rho(\mu) = \neg(p_2 \wedge p_1 \wedge p_3)$. Computing $\Delta_{\rho(\mu)}^{d_H, \Sigma}(\rho(E))$ we get that $\Delta_{\rho(\mu)}^{d_H, \Sigma}(\rho(E)) \equiv \{p_2 \wedge \neg(p_1 \wedge p_3)\} \equiv \rho(\Delta_\mu^{d_H, \Sigma}(E))$.

Entity resolution. Suppose that, at some point in the knowledge modelling process, different variables, e.g., p and q , are discovered to encode the same concept. The knowledge engineer would want to incorporate this equivalence in the merging outcome. One way to do this is going through the knowledge bases and the constraint and renaming p to q . This is a laborious and invasive operation, which might be infeasible if access to the knowledge bases is limited or if the knowledge bases are provided by the agents just in time before the merging process. Another way is to add the equivalence $p \leftrightarrow q$ directly to μ . The property we propose explores the relationship between these two operations and requires that all solutions of the latter operation are also solutions of the former. We denote by $\mu^{p/q}$ and $K^{p/q}$ the formula and knowledge base obtained from μ and K , respectively, by replacing every occurrence of p with q . We denote by $E^{p/q}$ the profile obtained from E by replacing every knowledge base K in E by $K^{p/q}$, if $K^{p/q}$ is consistent; if $K^{p/q}$ is inconsistent, we remove it.

$$(\text{Entity resolution}) \quad \Delta_{\mu \wedge (p \leftrightarrow q)}(E) \models \Delta_{\mu^{p/q}}(E^{p/q}).$$

Entity resolution has no direct equivalent in the voting scenario. Nonetheless, we believe it is worth investigating, as it bears some resemblance to Independence of clones (see below) and is motivated by a similar intuition: alternatives that are in some sense redundant should not skew the vote in their favour. In Example 1, we obtain that $\Delta_{\mu \wedge (p_1 \leftrightarrow p_2)}^{d_H}(E) \equiv \{p_1 \wedge p_2 \wedge \neg p_3\}$ (regardless of the aggregation function used). Replacing every occurrence of p_1 with p_2 in μ and E leaves us with $K_1^{p_1/p_2} = \{p_2 \wedge \neg p_3\}$, $K_2^{p_1/p_2} = \{\neg p_2\}$, $K_3^{p_1/p_2} = \{p_2 \wedge p_3\}$ and $\mu^{p_1/p_2} = \neg(p_2 \wedge p_3)$, and $\Delta_{\mu^{p_1/p_2}}^{d_H}(E^{p_1/p_2}) \equiv \{p_2 \wedge \neg p_3\}$. Clearly, in this case we have that $\Delta_{\mu \wedge (p_1 \leftrightarrow p_2)}^{d_H}(E) \models \Delta_{\mu^{p_1/p_2}}^{d_H}(E^{p_1/p_2})$. However, we show in Section 4 that this does not hold in general.

⁴ Here and in the following, variables such as E , K and μ are understood to be universally quantified, unless explicitly mentioned otherwise.

Fairness

The second group proposes a set of *fairness* properties, stemming from the intuition that all knowledge bases and all variables should be treated “equally” in the merging process. Fairness is featured in the IC postulates (through IC₄), but our proposals show that there is a wider range of constraints to consider.

Non-dictatorship. In a voting scenario, this property is satisfied if there is no single voter that alone determines the outcome of the election, and is usually featured as a key requirement that any reasonable voting method should satisfy. Here we distinguish between two notions: Non-Dictatorship₁ is in the spirit of the property usually found for voting, while Non-Dictatorship₂ has a more semantic flavour.

(Non-Dictatorship₁) There is no integer i such that for any $E \in \mathcal{E}$ and $\mu \in \mathcal{L}$, it holds that $\Delta_\mu(E) \equiv \Delta_\mu(\langle K_i \rangle)$.

(Non-Dictatorship₂) There is no $K \in \mathcal{K}$ such that for any $E \in \mathcal{E}$ and $\mu \in \mathcal{L}$, it holds that if a $K' \in \mathcal{K}$ occurs in E with $K' \equiv K$, then $\Delta_\mu(E) \equiv \Delta_\mu(\langle K' \rangle)$, for any $\mu \in \mathcal{L}$.

Property Non-Dictatorship₂ specifies that there is no knowledge base in the semantic sense (i.e., a specific set of beliefs modulo logical equivalence) which, if present in a profile, unilaterally determines the merging outcome. In a voting setting, this is equivalent to saying there is no *ranking of alternatives* (think of it as a magic key) which, if submitted by some voter, decides the winners. Non-dictatorship has been mentioned before in relation to Arrow’s theorem [12], though it has not been formalized explicitly.

Pareto consistency. A voting system is Pareto-consistent if whenever all voters prefer a candidate c_i over some candidate c_j , then c_i is preferred over c_j in the result. A stronger version stipulates that no candidates other than those preferred by all voters should appear in the result. In a merging scenario, we correspondingly distinguish between a weak and a strong version of Pareto consistency.

(Weak Pareto) $\Delta_\mu(\langle K_1 \rangle) \wedge \dots \wedge \Delta_\mu(\langle K_n \rangle) \models \Delta_\mu(\langle K_1, \dots, K_n \rangle)$.

(Strong Pareto) If $\Delta_\mu(\langle K_1 \rangle) \wedge \dots \wedge \Delta_\mu(\langle K_n \rangle)$ is consistent, then $\Delta_\mu(\langle K_1, \dots, K_n \rangle) \models \Delta_\mu(\langle K_1 \rangle) \wedge \dots \wedge \Delta_\mu(\langle K_n \rangle)$.

Replacing IC₅ and IC₆ in the IC postulates with Weak Pareto and Strong Pareto yields what is called a *pre-IC merging operator*. In [8] it has already been noted that any IC merging operator is also a pre-IC merging operator. Pareto conditions also occur in [4] in connection to a related set of operators called *fusion operators*.

Citizen’s sovereignty. In a voting scenario, citizen’s sovereignty requires that for any candidate c there is at least one election such that c is the winner. In other words, no candidate is disadvantaged by the voting system *per se*. In a merging scenario, we require that no formula is disadvantaged by the operator *per se*.

(Citizen’s sovereignty) For any formula φ there exist $E \in \mathcal{E}$ and $\mu \in \mathcal{L}$ such that $\Delta_\mu(E) \equiv \varphi$.

SC-Majority. This property requires that a candidate c is a winner whenever more than half of the voters have c as their most preferred candidate. Considering a formula φ as a set of candidates (i.e., φ ’s models) and the knowledge bases K_i as the voters, we have:

(SC-Majority) If $\varphi \in \mathcal{L}$ is consistent and $\varphi \models \Delta_\mu(\langle K_i \rangle)$ for a majority of $i \in \{1, \dots, n\}$, then $\varphi \models \Delta_\mu(E)$.

In Example 1 there is no consistent formula φ such that $\varphi \models \Delta_\mu(\langle K_i \rangle)$ for a majority of $i \in \{1, \dots, n\}$. Hence, when we view $\Delta_\mu^{dH}(E)$ as an election over the models of μ , there is no majority winner. However, merging the same profile under the constraint $\mu' = (p_1 \oplus p_2) \wedge \neg p_3$, we observe that μ' is a majority winner but $\Delta_{\mu'}^{dH}(E) \equiv \{p_1 \wedge \neg p_2 \wedge \neg p_3\}$. Clearly, though, $\mu' \not\models \Delta_{\mu'}^{dH}(E)$.

Condorcet criterion. In a voting scenario the Condorcet criterion is satisfied if the voting system selects the Condorcet winner, if it exists. The Condorcet winner is a candidate that beats every other candidate in pairwise majority comparisons. In a merging scenario, our proposal is to define majority comparisons in terms of complete formulas.⁵ We opted to present this version here as it directly captures the intuition of the Condorcet winner from the voting scenario. In a more extensive treatment of the topic we would present it alongside an equivalent simpler version.

Definition 2. Given a merging operator Δ , $E \in \mathcal{E}$, $\mu \in \mathcal{L}$ and two complete formulas $\varphi_1, \varphi_2 \in \mathcal{L}$ such that $\varphi_1 \models \mu$ and $\varphi_2 \models \mu$, a *head-to-head election between φ_1 and φ_2* occurs as follows: for every K_i in E , we say that φ_1 wins over φ_2 with respect to K_i if $\Delta_{\varphi_1 \vee \varphi_2}(\langle K_i \rangle) \wedge \varphi_1$ is consistent and $\Delta_{\varphi_1 \vee \varphi_2}(\langle K_i \rangle) \wedge \varphi_2$ is inconsistent. If both $\Delta_{\varphi_1 \vee \varphi_2}(\langle K_i \rangle) \wedge \varphi_1$ and $\Delta_{\varphi_1 \vee \varphi_2}(\langle K_i \rangle) \wedge \varphi_2$ are consistent, we say that φ_1 and φ_2 are tied with respect to K_i . We denote by $W_E(\varphi_1, \varphi_2)$ the number of wins of φ_1 over φ_2 in E . Finally, we say that φ_1 wins over φ_2 in a head-to-head election over E if $W_E(\varphi_1, \varphi_2) \geq W_E(\varphi_2, \varphi_1)$.⁶ A complete formula φ such that $\varphi \models \mu$ is a *weak Condorcet winner with respect to E* and μ if for any complete formula $\varphi' \models \mu$ such that $\varphi \not\models \varphi'$, it holds that φ wins over φ' in a head-to-head election over E .

(Condorcet’s criterion) If φ is a weak Condorcet winner with respect to E and μ , then $\varphi \models \Delta_\mu(E)$.

According to our definition, a weak Condorcet winner on formulas can be shown to coincide with the more familiar notion of a weak Condorcet winner from voting theory, by viewing the set of pre-orders $\{\leq_{\langle K_1 \rangle}, \dots, \leq_{\langle K_n \rangle}\}$ in a syncretic assignment as a voting profile where $[\mu]$ is the set of candidates (see Theorem 3 and the sc_{on} property). Applying this result here, we consider merging the profile E from Example 1 under a constraint μ' such that $[\mu'] = \{000, 001, 010, 100\}$ and using Hamming distance and Σ as aggregation function. We obtain the same table of distances from Example 1, except that we restrict our attention to the models of μ' . Table 2 records the number of wins of each interpretation in $[\mu']$ over the other in the resulting voting profile: an entry of k in row i and column j means that interpretation w_i has k wins over interpretation w_j . For instance, 000 has only one win over 001 (namely, $000 <_{K_1} 001$). Likewise, 001 has only one win over 000 (namely, $001 <_{K_3} 000$).⁷ Obviously, from a voting perspective it does not make sense to compare an interpretation to itself, thus the entries on the diagonal are marked with “-”. Inspection of Table 2 then shows that 001 and 100 are the only models that do not lose to any other interpretation, which means that they are the weak Condorcet winners in this profile. Hence the formulas $\varphi_1 = \neg p_1 \wedge \neg p_2 \wedge p_3$ and $\varphi_2 = p_1 \wedge \neg p_2 \wedge \neg p_3$ are the corresponding weak Condorcet winners.

⁵ Complete formulas have exactly one model.

⁶ We opted to go with the weak form of Condorcet winner because we did not wish to restrict the set of winners to have exactly one model. However, we could define a strong notion of Condorcet winner by requiring the inequality between $W_E(\varphi_1, \varphi_2)$ and $W_E(\varphi_2, \varphi_1)$ to be strict and our analysis would still go through. For reasons of space we omit this here.

⁷ We do not count the tie $000 \approx_{K_2} 001$.

	000	001	010	100
000	-	1	1	1
001	1	-	2	1
010	1	1	-	0
100	2	1	1	-

Table 2: Computing the Condorcet winners

On the other hand, $\Delta_{\mu'}^{d_H, \Sigma}(E) \equiv \{p_1 \wedge \neg p_2 \wedge \neg p_3\}$ and it is clear that $\varphi_1 \not\models \Delta_{\mu'}^{d_H, \Sigma}(E)$ and thus $\Delta_{\mu'}^{d_H, \Sigma}(E)$ does not select (all) the weak Condorcet winners of this profile. In Section 4 we will show that this observation generalizes to the other merging operators.

Intuitive Response to Profile Change

This group of properties ensures that changes in the knowledge base produce an intuitive change of the outcome. Having an *intuitive response* of the formalism is particularly important for knowledge engineers as it reduces unnatural behavior and makes the effects of changes in the knowledge bases easier to grasp.

Monotonicity. A voting system is monotone if the winner of an election cannot be turned into a non-winner by improving its rank in some of the votes. In the context of merging, we propose:

(Monotonicity) $\Delta_{\mu}(E_1 \sqcup E_2) \wedge \Delta_{\mu}(E_3) \models \Delta_{\mu}(E_1 \sqcup E_3)$.

The intuition behind this formalization stems from seeing the models of $\Delta_{\mu}(E)$ as the winners in the election where the models of μ are candidates and the knowledge bases in E are the voters. Thus, if any candidates elected by the profile $E_1 \sqcup E_2$ are also elected by the profile E_3 alone, then monotonicity would require that the same candidates should also be elected when we replace E_2 with E_3 in $E_1 \sqcup E_2$. The idea, to put it succinctly, is that a winner stays a winner, if its position is only increased in the votes.

To illustrate the property, consider the knowledge bases in Example 1 and a constraint μ' such that $[\mu'] = \{011, 100\}$. Then $100 \in [\Delta_{\mu'}^{d_H}(\langle K_1, K_2 \rangle)]$. In other words, the interpretation 100 is a winner in an election where K_1 and K_2 are the voters and the interpretations 011 and 100 (as models of μ') are the sole candidates. We also see, by consulting Table 1, that $100 \in [\Delta_{\mu'}^{d_H}(\langle K_3 \rangle)]$, i.e., the voter K_3 counts 100 among its most preferred states. Monotonicity would then require that replacing K_2 with K_3 in the profile $\langle K_1, K_2 \rangle$ would not harm the position of 100 in the result. And indeed, we have that $100 \in [\Delta_{\mu'}^{d_H}(\langle K_1, K_3 \rangle)]$, showing that Monotonicity is satisfied in this particular instance. However, in Section 4 we show that Monotonicity is not satisfied in general by Δ^{d_H} and Δ^{d_D} .

Participation. A voting system satisfies participation (also known as the *no-show paradox*) if it is not possible to change the winner from candidate c_i to candidate c_j by adding a vote in which candidate c_i is strictly preferred to candidate c_j . In a merging scenario, we consider adding a knowledge base K to a given profile E and require that $\Delta_{\mu}(E \sqcup \langle K \rangle)$ should not be ‘worse’ than $\Delta_{\mu}(E)$ with respect to K .

(Participation) If $\Delta_{\mu}(E) \wedge K$ is consistent, then $\Delta_{\mu}(E) \wedge K \models \Delta_{\mu}(E \sqcup \langle K \rangle)$.

In Example 1, take $\mu' = \neg p_2 \wedge p_3$, with $[\mu'] = \{001, 101\}$. We have that $[\Delta_{\mu'}^{d_H, \Sigma}(\langle K_1, K_2 \rangle)] = \{101\}$. In other words, if Reviewers 1 and 2 decided alone, then 101 would be their most preferred state, as chosen by $\Delta^{d_H, \Sigma}$. Notice that 101 is also a model of K_3 , i.e., Reviewer 3 also has 101 among its most preferred states. We

can imagine Reviewer 3 has a choice: she can either express her opinions, or stand by as a passive observer. Now, if there was a possibility that weighing in with her true opinions would *decrease* the chance that 101 appears in the result, then Reviewer 3 would have an incentive to keep her opinion to herself. This does not happen, as $101 \in [\Delta_{\mu'}^{d_H, \Sigma}(\langle K_1, K_2, K_3 \rangle)]$. Hence it is safe for Reviewer 3 to weigh in on the reviewing process with her true opinions. We would want all merging operators to emulate this property, as it incentivizes agents to participate with their honest opinions.

Reversal symmetry. This property holds in a voting system if the unique winner of an election can be turned into a non-winner by reversing all votes. In a merging scenario, we interpret the condition of having a unique winner as the outcome of merging being a complete formula, and we take reversing the vote to mean that every knowledge base is replaced with its negation, as defined in Section 2. Notice that we require the outcome to be a complete formula to reflect the requirement of a unique winner in the voting setting.

(Reversal symmetry) If $\Delta_{\mu}(E)$ is a complete formula and μ has more than one model, then $\Delta_{\mu}(E) \not\models \Delta_{\mu}(\neg E)$.

In Example 1, replacing every knowledge base with its negation gives $\neg K_1 = \{\neg(p_1 \wedge p_2 \wedge \neg p_3)\}$, $\neg K_2 = \{\neg(\neg p_1 \wedge \neg p_2)\}$ and $\neg K_3 = \{\neg(p_1 \wedge p_3)\}$. Merging these knowledge bases with $\Delta^{d_H, GMAX}$ under the constraint μ (from the example) produces the result $\Delta_{\mu}^{d_H, GMAX}(\neg E) = \{010, 011, 100, 101\}$. Thus, $\Delta_{\mu}^{d_H, GMAX}(E) \models \Delta_{\mu}^{d_H, GMAX}(\neg E)$, and hence $\Delta^{d_H, GMAX}$ does not satisfy Reversal symmetry. In Section 4 it is shown that this result extends to other merging operators as well.

Resolvability. In a voting scenario, resolvability (see, e.g., [17]) requires that any winner can be made the unique winner by adding a single vote. In a merging scenario, we require that we can refine the output of merging as much as we desire by adding just one knowledge base to E .

(Resolvability) For any $\varphi \in \mathcal{L}$ such that $\varphi \models \Delta_{\mu}(E)$, there is a $K \in \mathcal{K}$ such that $\Delta_{\mu}(E \sqcup \langle K \rangle) \equiv \varphi$.

It has been pointed out in Section 2 that the output of a merging operator is not always resolute, in the sense of selecting a completely specified state of affairs. In Example 1 we got that $\Delta_{\mu}^{d_H, \Sigma}(E) \equiv \{p_1 \wedge \neg(p_2 \wedge p_3)\}$, thus saying that Paper 1 should be accepted while Papers 2 and 3 cannot be accepted together, but not giving any additional information on which (if any) of Papers 2 and 3 should be accepted. This is because merging operators are designed to offer a solution based on the available information, and that might be insufficient to decide between a set of alternatives. However, in certain circumstances, such as the one offered in Example 1, we might want an answer that settles the question definitively. In such a case, it is reasonable to do so by eliciting more information from the agents involved. The Resolvability property analyzes the possibility that the result can be refined enough by adding a single vote, so as to settle on a decision regarding every option. In Example 1 we can settle on the decision where, for instance, only Paper 1 is accepted by adding the knowledge base $K_4 = \{p_1 \wedge \neg p_2 \wedge \neg p_3\}$ to the profile. Notice that our definition of resolvability does not require the operator to be resolute.

Independence of clones. In a voting scenario, we say that two candidates are clones if they are ranked next to each other in any vote of the election. A voting system is independent of clones if a non-winning candidate cannot be made a winner by adding clones to the election.

In a merging scenario, as it does not make too much sense to think of introducing new interpretations, we think of clones as new *variables* that are equivalent to existing ones. Thus, given a merging profile $E = \langle K_1, \dots, K_n \rangle$, a propositional variable p and a set of “new” propositional variables $\mathcal{Q} \subset \mathcal{P}$ not appearing in K_1, \dots, K_n (clones of p), we denote by $E^{p, \mathcal{Q}}$ the profile obtained by adding the formula $\bigwedge_{q \in \mathcal{Q}} (p \leftrightarrow q)$ to every knowledge base K_i contained in E . Independence of clones for merging operators is formulated as follows:

(Independence of clones) If every $K \in E^{p, \mathcal{Q}}$ is consistent, then $\Delta_\mu(E) \equiv \Delta_\mu(E^{p, \mathcal{Q}})$.

Consider merging the knowledge bases $\{p\}$ and $\{\neg p\}$ with the operator $\Delta^{d_H, \Sigma}$: we get that $\Delta^{d_H, \Sigma}(\{\{p\}, \{\neg p\}\}) \equiv \top$. Adding a clone q of p gives us that $\Delta^{d_H, \Sigma}(\{\{p, p \leftrightarrow q\}, \{\neg p, p \leftrightarrow q\}\}) \equiv \top$. Adding a clone to the profile does not change the final result and this seems fitting, as introducing the new information regarding q does not change the agents’ beliefs regarding the “main” issue, represented by p . Hence, one would like to see this behaviour reproduced more generally. However, Independence of clones as we have formulated it is a very strong property. Thus, adding a clone p_4 for p_1 in Example 1 produces the result that $\Delta_\mu^{d_H, \Sigma}(E) \equiv \{p_1 \wedge \neg(p_2 \wedge p_3) \wedge p_4\}$, $\Delta_\mu^{d_H, GMAX}(E) \equiv \{p_1 \wedge \neg p_2 \wedge \neg p_3 \wedge p_4\}$, $\Delta_\mu^{d_H, GMIN}(E) \equiv \{p_1 \wedge p_2 \wedge \neg p_3 \wedge p_4\}$. Obviously Independence of clones is not satisfied here, and Section 4 shows that this result generalizes.

Modularity

Modularity properties capture circumstances where a profile can be decomposed into sub-profiles while preserving the merging result.

Consistency. In a voting scenario, consistency requires that if an election E is arbitrarily divided into sub-elections E_1, \dots, E_n and if candidate c is a winner in all of the sub-elections E_1, \dots, E_n , then c is also a winner of E . For merging we formulate consistency as follows:

(Consistency) For any partition E_1, \dots, E_n of E it holds that $\Delta_\mu(E_1) \wedge \dots \wedge \Delta_\mu(E_n) \models \Delta_\mu(E)$.

Observe that Consistency and Weak Pareto do not coincide, as Consistency is stronger than Weak Pareto.

Stability

These properties are subtly different to those describing intuitive response to profile change: they model modifications of the knowledge bases which should *not* affect the result of the merging process.

Homogeneity. A voting procedure satisfies homogeneity if for any $k \geq 1$ and any election, the result cannot be changed by “repeating” each vote k times. In a merging scenario we require that the outcome of merging does not change if we expand the profile by adding multiple copies of itself. That is, the absolute “weights” of the knowledge bases are not relevant—rather it is the relative weights that matter.

(Homogeneity) $\Delta_\mu(E) \equiv \Delta_\mu(E \sqcup \dots \sqcup E)$.

Self-agreement. We require that the merging outcome is not disrupted if we add it back to E and merge the new profile.

(Self-agreement) $\Delta_\mu(E \sqcup \langle \Delta_\mu(E) \rangle) \equiv \Delta_\mu(E)$.

4 Relationship with IC postulates

In this section we analyze the properties introduced in Section 3. The results are summarized in Table 3. A significant number of the properties we introduced turn out to follow directly from the IC postulates, whereas there are some that hold for certain operators only.

Property	IC	Δ^{d_H}	Δ^{d_D}
Anonymity	✓	✓	✓
Neutrality		✓	✓
Entity resolution		×	✓
Non-Dictatorship ₁ , Non-Dictatorship ₂	✓	✓	✓
Weak Pareto*	✓	✓	✓
Strong Pareto*	✓	✓	✓
Citizen’s sovereignty	✓	✓	✓
SC-Majority	✗	×	×
Condorcet’s criterion		×	✓
Monotonicity		×	×
Participation	✓	✓	✓
Reversal symmetry		×	✓
Resolvability	✓	✓	✓
Independence of clones		×	×
Consistency	✓	✓	✓
Homogeneity	✓	✓	✓
Self-agreement	✓	✓	✓

Table 3: Summary of results. In the IC column, ✓ indicates that the property is implied by the IC postulates, and ✗ indicates that the property is inconsistent with the IC postulates. The last two columns indicate whether the property holds for operators based on Hamming distance (Δ^{d_H}) and drastic distance (Δ^{d_D}). Results for properties marked by * have already been studied [8].

Theorem 2. Anonymity, Non-Dictatorship₁, Non-Dictatorship₂, Weak Pareto, Strong Pareto, Citizen’s sovereignty, Participation, Resolvability, Consistency, Homogeneity and Self-agreement follow from the IC postulates.

Proof. For Anonymity take the bijection $f(K_i) = K_{\pi(i)}$ between E and $\pi(E)$ and apply IC₃. Non-Dictatorship₁ follows from Anonymity, as in the classical voting scenario. For Non-Dictatorship₂, suppose K_1 is a dictator for Δ . Choose a (consistent) K_2 such that $\bigwedge K_1 \wedge \bigwedge K_2$ is inconsistent, and $\mu = \bigwedge K_1 \vee \bigwedge K_2$. Clearly, $\Delta_\mu(\langle K_1, K_2 \rangle) \wedge \bigwedge K_1$ is consistent, and thus (by IC₄) it holds that $\Delta_\mu(\langle K_1, K_2 \rangle) \wedge \bigwedge K_2$ is consistent as well. But, since K_1 is a dictator, we have that $\Delta_\mu(\langle K_1, K_2 \rangle) \equiv \Delta_\mu(\langle K_1 \rangle)$. This leads to a contradiction. Weak Pareto and Strong Pareto are discussed in [8]. For Citizen’s sovereignty take $E = \langle \{\varphi\} \rangle$, $\mu = \varphi$ and apply IC₂. For Participation take $w \in [\Delta_\mu(E) \wedge K]$. By IC₁, this implies that $w \in [\mu]$. We also have that $w \in [K]$, and from IC₂ it follows that $\Delta_\mu(\langle K \rangle) \equiv \bigwedge \langle K \rangle \wedge \mu$, hence $w \in [\Delta_\mu(\langle K \rangle)]$. This implies that $w \in [\Delta_\mu(E) \wedge \Delta_\mu(\langle K \rangle)]$, and by IC₅ we get that $w \in [\Delta_\mu(E \sqcup \langle K \rangle)]$. For Resolvability, take $K = \{\varphi\}$. By IC₀ it follows that $\varphi \models \mu$. Hence, $K \wedge \mu$ is consistent, and by IC₂ it follows that $\Delta_\mu(\langle K \rangle) \equiv \bigwedge \langle K \rangle \wedge \mu \equiv \varphi$. It follows that $\Delta_\mu(E) \wedge \Delta_\mu(\langle K \rangle)$ is consistent. The conclusion follows by using IC₅–IC₆. Consistency follows from repeated application of IC₅, and Homogeneity from repeated application of IC₅–IC₆. For Self-agreement first show, using IC₀, IC₁ and IC₂, that $\Delta_\mu(\langle \Delta_\mu(E) \rangle) \equiv \Delta_\mu(E)$. From this, together with the fact that $\Delta_\mu(E) \wedge \Delta_\mu(\langle \Delta_\mu(E) \rangle)$ is consistent, plus IC₅–IC₆, we get $\Delta_\mu(E \sqcup \langle \Delta_\mu(E) \rangle) \equiv \Delta_\mu(E) \wedge \Delta_\mu(\langle \Delta_\mu(E) \rangle) \equiv \Delta_\mu(E)$. □

The remaining properties require a different kind of analysis. Below we present a series of conditions on assignments which turn out to characterize several important remaining properties.

Definition 3. For an assignment on profiles, we define the following properties, for any $w, w_1, w_2 \in \mathcal{W}$, $K_1, \dots, K_n \in \mathcal{K}$, $E, E_1, E_2, E_3 \in \mathcal{E}$, $\mathcal{M} \subseteq \mathcal{W}$ and transposition τ :⁸

- (S_{Neut}) If $w_1 \leq_E w_2$, then $\tau(w_1) \leq_{\tau(E)} \tau(w_2)$.
- (S_{Maj}) If $w_1 \leq_{\langle K_i \rangle} w_2$ for a majority of $i \in \{1, \dots, n\}$, then $w_1 \leq_{\langle K_1, \dots, K_n \rangle} w_2$.
- (S_{Con}) If $w \in \mathcal{M}$ is a weak Condorcet winner with respect to the preference profile $\langle \leq_{\langle K_1 \rangle}, \dots, \leq_{\langle K_n \rangle} \rangle$ and \mathcal{M} is the set of candidates, then $w \in \min_{\leq_{\langle K_1, \dots, K_n \rangle}} \mathcal{M}$.
- (S_{Mon}) If $w_1 \leq_{E_1 \sqcup E_2} w_2$ and $w_1 \leq_{E_3} w_2$, then $w_1 \leq_{E_1 \sqcup E_3} w_2$.
- (S_{Rev}) If $w_1 <_E w_2$, then $w_2 <_{\neg E} w_1$.

We say that *property s of assignments characterizes property P of merging operators* if it holds that a merging operator satisfies P iff it is represented by an assignment satisfying s.

Theorem 3. *Properties S_{Neut}, S_{Maj}, S_{Con}, S_{Mon} and S_{Rev} characterize Neutrality, SC-Majority, Condorcet's criterion, Monotonicity and Reversal symmetry, respectively.*

Proof. For S_{Maj}, S_{Con}, S_{Mon} and S_{Rev} it is straightforward to check that they characterize their respective properties for operators. For S_{Neut}, suppose first that we have a neutral assignment and a merging operator Δ represented by it. We know that any renaming ρ is the product of n transpositions. We show that Δ satisfies Neutrality by induction on n . In the base case of $n = 0$, ρ is the identity renaming and the claim holds trivially. For the inductive step, we assume the claim holds for permutations of length n , and show that it holds for permutations of length $n + 1$. Take, then, a renaming $\rho = \tau_1 \dots \tau_n \tau_{n+1}$. By the inductive hypothesis, we know that $\tau_1 \dots \tau_n(\Delta_\mu(E)) \equiv \Delta_{\tau_1 \dots \tau_n(\mu)}(\tau_1 \dots \tau_n(E))$. We apply τ_{n+1} to both sides. Using the results that for any $\varphi \in \mathcal{L}$ and transposition τ it holds that $[\tau(\varphi)] = \tau([\varphi])$ and $\tau(\tau(w)) = w$, and that for any $E \in \mathcal{E}$, $\mu \in \mathcal{L}$, it holds that $\tau(\Delta_\mu(E)) \equiv \Delta_{\tau(\mu)}(\tau(E))$, we derive the conclusion. Conversely, assume Δ is a merging operator that satisfies Neutrality but is represented by an assignment that does not satisfy S_{Neut}. Then there exists $E \in \mathcal{E}$, a transposition τ and $w_1, w_2 \in \mathcal{W}$ such that $w_1 \leq_E w_2$ and $\tau(w_2) <_{\tau(E)} \tau(w_1)$. Take $\mu \in \mathcal{L}$ such that $[\mu] = \{w_1, w_2\}$. We have that $w_1 \in [\Delta_\mu(E)]$ and hence $\tau(w_1) \in [\tau(\Delta_\mu(E))]$. On the other hand, $[\Delta_{\tau(\mu)}(\tau(E))] = \{w_2\}$. This shows that Δ is not neutral, which is a contradiction. \square

Theorem 4. *None of the operators Δ^{d_H} and Δ^{d_D} satisfies Monotonicity or Independence of clones. The operators Δ^{d_H} do not satisfy Entity resolution, Condorcet's criterion and Reversal symmetry, but Δ^{d_D} does. Furthermore, there is no IC merging operator that satisfies SC-Majority.*

Proof. We provide here the relevant counter-examples. For Monotonicity take $K_1 = \{p \wedge q\}$, $K_2 = \{\neg q\}$, $K_3 = \{p\}$, $\mu = p$ and $E_1 = \langle K_1 \rangle$, $E_2 = \langle K_2 \rangle$, $E_3 = \langle K_3 \rangle$. We get that $\Delta_\mu^{d_H}(E_1 \sqcup E_2) \equiv \Delta_\mu^{d_D}(E_1 \sqcup E_2) \equiv \{p\}$, $\Delta_\mu^{d_H}(E_3) \equiv \Delta_\mu^{d_D}(E_3) \equiv \{p\}$ and $\Delta_\mu^{d_H}(E_1 \sqcup E_3) \equiv \Delta_\mu^{d_D}(E_1 \sqcup E_3) \equiv \{p \wedge q\}$. For Independence of clones take $E = \langle K_1, K_2 \rangle$, $K_1 = \{p\}$ and $K_2 = \{q\}$, $\mu = p \vee q$ and add a clone r of p . Then $\Delta_\mu^{d_H}(E) \equiv \{p \wedge q\}$. In the new setup, we get that $\Delta_\mu^{d_H}(E^{p, \emptyset}) \equiv \{p \wedge q \wedge r\}$. For Entity resolution and Δ^{d_H} , take $E = \langle K_1, K_2, K_3 \rangle$ with $K_1 = \{p \wedge \neg q \wedge r\}$, $K_2 = \{p \wedge \neg r\}$, $K_3 = \{\neg p \wedge \neg q\}$ and $\mu = \top$. We get $\Delta_{\mu \wedge (p \leftrightarrow q)}^{d_H}(\langle K_1, K_2 \rangle) \equiv \{\neg p \wedge \neg q\}$, while $E^{p/q} =$

$\langle \{q \wedge \neg r\}, \{\neg q\} \rangle$ and $\Delta_{\mu \wedge (p \leftrightarrow q)}^{d_H}(E^{p/q}) \equiv \{\neg r\}$. For Entity resolution and Δ^{d_D} , take $w \in [\Delta_{\mu \wedge (p \leftrightarrow q)}^{d_D}(E)]$. We have that if $w \in [K_i]$ and $K_i^{p/q}$ is consistent, then $w \in [K_i^{p/q}]$. This implies that the number of 0's in w 's vector of scores in $\leq_{E^{p/q}}$ is equal to the number of 0's in w 's vector of scores in $\leq_{E^{p/q}}$. If $K_i^{p/q}$ is inconsistent, this is because K_i implies either $p \wedge \neg q$ or $\neg p \wedge q$, and thus w cannot be a model of K_i ; hence removing K_i can only decrease w 's final score. For Condorcet's criterion and $\Delta^{d_H, \Sigma}$ or $\Delta^{d_H, GMAX}$, take $E = \langle K_1, K_2, K_3 \rangle$, $K_1 = \{p \wedge q \wedge r\}$, $K_2 = K_3 = \{\neg p\}$, $\mu = (\neg p \wedge \neg q \wedge \neg r) \vee (p \wedge q \wedge r)$. Then $[K_1] = \{111\}$, $[K_2] = [K_3] = \{000, 001, 010, 011\}$ and $[\mu] = \{000, 111\}$. We have that $\varphi = \neg p \wedge \neg q \wedge \neg r$ is the only weak Condorcet winner with respect to E and μ , but $\Delta_\mu^{d_H, \Sigma}(E) \equiv \Delta_\mu^{d_H, GMAX}(E) \equiv \{p \wedge q \wedge r\}$. For $\Delta^{d_H, GMIN}$, take $E = \langle K_1, K_2, K_3 \rangle$, $K_1 = \{\neg p \wedge \neg q \wedge r\}$, $K_2 = \{\neg p \wedge q \wedge \neg r\}$, $K_3 = \{p \wedge q \wedge r\}$ and μ from before. Then the weak Condorcet winner with respect to E and μ is $\neg p \wedge \neg q \wedge \neg r$, but $\Delta_\mu^{d_H, GMIN}(E) \equiv \{p \wedge q \wedge r\}$. It is straightforward to check that d_D together with any aggregation function generates an assignment that satisfies S_{Con}. Together with Theorem 3 and our observation that a weak Condorcet winner φ with respect to E and μ corresponds to $[\varphi]$ being a weak Condorcet winner in the voting profile $\langle \leq_{\langle K_1 \rangle}, \dots, \leq_{\langle K_n \rangle} \rangle$ restricted to $[\mu]$, we get that Δ^{d_D} satisfies Condorcet's criterion. For Reversal symmetry and d_H , take $K_1 = \{p \rightarrow q\}$, $K_2 = \{p \wedge \neg q\}$, $\mu = \neg p$ and $E = \langle K_1, K_2 \rangle$. We get that $\Delta_\mu^{d_H}(E) \equiv \Delta_\mu^{d_H}(\neg E) \equiv \{\neg p \wedge \neg q\}$. It is straightforward to check that d_D with any aggregation function generates an assignment that satisfies S_{Rev}, thus Δ^{d_D} satisfies Reversal symmetry. To see why the IC postulates and SC-Majority are incompatible, suppose there is an IC merging operator which satisfies SC-Majority. Take $[\mu] = \{w_1, w_2\}$ and $[K_1] = \{w_1, w_2\}$, $[K_2] = \{w_1, w_2\}$, $[K_3] = \{w_1\}$. By SC-Majority we get that $\{w_1, w_2\} \subseteq [\Delta_\mu(\langle K_1, K_2, K_3 \rangle)]$. However, by IC₂ we get that $[\Delta_\mu(\langle K_1, K_2, K_3 \rangle)] = \{w_1\}$. \square

Finally, we show that Neutrality is not implied by the IC postulates, even though Δ^{d_H} and Δ^{d_D} satisfy it. To do so, we first provide a characterization of Neutrality in terms of a corresponding property for distance based operators. We call a pseudo-distance d *neutral* if for any transposition τ and $w_1, w_2 \in \mathcal{W}$, it holds that $d(w_1, w_2) = d(\tau(w_1), \tau(w_2))$. The characterization is then captured by the following result.

Theorem 5. *For any pseudo-distance d and aggregation function f , a merging operator $\Delta^{d, f}$ satisfies Neutrality if and only if d is neutral.*

Proof. For one direction of the proof, take an assignment generated using a neutral distance d and an aggregation function f . First we show that for any $w \in \mathcal{W}$, $K \in \mathcal{K}$ and transposition τ , it holds that $d(w, K) = d(\tau(w), \tau(K))$. Take $w' \in [K]$ such that $d(w, K) = d(w, w')$. Since τ is neutral, we get that $d(w, w') = d(\tau(w), \tau(w'))$. We have that $[\tau(K)] = \tau([K])$, and thus $\tau(w') \in [\tau(K)]$. We show now that $\tau(w')$ is at a minimal distance from $\tau(w)$ among the models of $\tau(K)$. Take, then, $\tau(w'') \in [\tau(K)]$, with $w'' \in [K]$. We have that $d(w, w') \leq d(w, w'')$, and since d is neutral it follows that $d(\tau(w), \tau(w')) \leq d(\tau(w), \tau(w''))$. Hence $d(\tau(w), \tau(K)) = d(\tau(w), \tau(w')) = d(w, w')$. From this we immediately derive that for any $E \in \mathcal{E}$, $w \in \mathcal{W}$ and neutral transposition τ , it holds that $d(w, E) = d(\tau(w), \tau(E))$. Thus, if $d(w_1, E) \leq d(w_2, E)$, then $d(\tau(w_1), \tau(E)) \leq d(\tau(w_2), \tau(E))$, for any $w_1, w_2 \in \mathcal{W}$. It follows that if $w_1 \leq_E w_2$ then $\tau(w_1) \leq_{\tau(E)} \tau(w_2)$, and therefore the assignment satisfies S_{Neut}. By Theorem 3 this implies $\Delta^{d, f}$ is neutral.

⁸ We remind the reader that transpositions applied to formulas swap exactly two atoms among each other and applied to interpretations they swap the corresponding bits in the bit-vector representation.

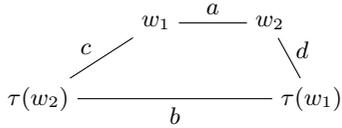


Figure 1: Distances between w_1 , w_2 , $\tau(w_1)$ and $\tau(w_2)$

Conversely, assume $\Delta^{d,f}$ satisfies Neutrality, but d is not neutral. Then there must be $w_1, w_2 \in \mathcal{W}$ and a transposition τ such that $d(w_1, w_2) \neq d(\tau(w_1), \tau(w_2))$. We show now that, by taking $\rho = \tau$, we can always find some profile E and constraint μ such that $\Delta^{d,f}$ does not satisfy Neutrality. Notice first that it is not possible to have $\tau(w_1) = w_1$ and $\tau(w_2) = w_2$, as this contradicts our assumption that $d(w_1, w_2) \neq d(\tau(w_1), \tau(w_2))$. In the following we analyze the remaining cases, keeping in mind that $\tau(\tau(w)) = w$ and that $[\tau(K)] = \tau([K])$. Let us denote $d(w_1, w_2) = a$, $d(\tau(w_1), \tau(w_2)) = b$, $d(w_1, \tau(w_2)) = c$ and $d(\tau(w_1), w_2) = d$ (see Figure 1). Without loss of generality, we assume that $a < b$. *Case 1.* $\tau(w_1) = w_1$, $\tau(w_2) \neq w_2$. Take K and μ such that $[K] = \{w_1\}$ and $[\mu] = \{w_2, \tau(w_2)\}$. We have that $[\tau(K)] = \tau([K]) = \{\tau(w_1)\} = \{w_1\}$, and $[\tau(\mu)] = [\mu]$. Then by $b = c$ we obtain $[\Delta_{\tau(\mu)}^{d,f}(\langle K \rangle)] = [\Delta_{\mu}^{d,f}(\tau(\langle K \rangle))] = \{w_2\}$. This shows that $\Delta^{d,f}$ is not neutral, since $[\tau(\Delta_{\mu}^{d,f}(\langle K \rangle))] = \tau([\Delta_{\mu}^{d,f}(\langle K \rangle)]) = \{\tau(w_2)\}$. *Case 2.* $\tau(w_1) \neq w_1$, $\tau(w_2) = w_2$. Analogous to Case 1. *Case 3.* $\tau(w_1) \neq w_1$, $\tau(w_2) \neq w_2$. For this case we have to analyze the relationship between a , b , c and d . *Case 3.1.* $\min\{a, c\} < \min\{b, d\}$ or $\min\{b, d\} < \min\{a, c\}$. Take $[K] = \{w_2, \tau(w_2)\}$ and $[\mu] = \{w_1, \tau(w_1)\}$. Clearly, $[\tau(K)] = [K]$ and $[\tau(\mu)] = [\mu]$. In this case we have that $d(w_1, K) = \min\{a, c\}$ and $d(\tau(w_1), K) = \min\{b, d\}$. Then $[\Delta_{\mu}^{d,f}(\langle K \rangle)]$ will consist of exactly one interpretation out of $\{w_1, \tau(w_1)\}$, call it w (see Table 4). But this shows that $\Delta^{d,f}$ cannot be neutral, because we will get the

w	$\{w_2, \tau(w_2)\}$
w_1	$\min\{a, c\}$
$\tau(w_1)$	$\min\{b, d\}$

Table 4: $\min\{a, c\} \neq \min\{b, d\}$

same result of $\{w\}$ for $[\Delta_{\tau(\mu)}^{d,f}(\tau(\langle K \rangle))]$, while $[\tau(\Delta_{\mu}^{d,f}(\langle K \rangle))] = \{\tau(w)\}$. *Case 3.2.* $\min\{a, c\} = \min\{b, d\}$. Here we analyze two sub-cases, but the reasoning follows the same lines as in the previous cases. *Case 3.2.1.* $a \leq c$, $d \leq b$, $a = d$. Take K and μ such that $[K] = \{w_1\}$ and $[\mu] = \{w_2, \tau(w_2)\}$. Then $[\tau(K)] = \{\tau(w_1)\}$ and $[\tau(\mu)] = [\mu]$ and we get that $[\Delta_{\mu}^{d,f}(\langle K \rangle)] = [\Delta_{\tau(\mu)}^{d,f}(\tau(\langle K \rangle))] = \{w_2\}$, whereas $[\tau(\Delta_{\mu}^{d,f}(\langle K \rangle))] = \{\tau(w_2)\}$. *Case 3.2.2.* $c \leq a$, $d \leq b$, $c = d$. Take K_1, K_2 and μ such that $[K_1] = \{w_2\}$, $[K_2] = \{\tau(w_2)\}$, $[\mu] = \{w_1, \tau(w_1)\}$. Then $[\tau(K_1)] = \{\tau(w_2)\}$, $[\tau(K_2)] = \{w_2\}$ and $[\tau(\mu)] = [\mu]$. Notice, now, that from $c = d$, $a < b$ and properties (i) and (iv) of f as an aggregation function (see Section 2), it follows that $f(a, c) < f(d, b)$. Consequently, $[\Delta_{\mu}^{d,f}(\langle K_1, K_2 \rangle)] = [\Delta_{\tau(\mu)}^{d,f}(\tau(\langle K_1, K_2 \rangle))] = \{w_1\}$. However, $[\tau(\Delta_{\mu}^{d,f}(\langle K_1, K_2 \rangle))] = \{\tau(w_1)\}$. Thus, $\Delta^{d,f}$ is not neutral. \square

Using Theorem 5, we can now state our last result.

Theorem 6. Neutrality does not follow from the IC postulates, but Δ^{d_H} and Δ^{d_D} satisfy it.

Proof. It is straightforward to check that d_H and d_D are neutral, hence by Theorem 5 it follows that Δ^{d_H} and Δ^{d_D} satisfy Neutrality. However Neutrality is not guaranteed by the IC postulates: there exist

distance-based operators satisfying the IC postulates where the distance d is nonetheless not neutral. One such example is a merging operator for the Horn fragment of propositional logic based on a custom-defined distance d_S [11]. In a three letter alphabet the definition of d_S specifies that $d_S(000, 001) = 1$ and $d_S(000, 010) = 2$. Thus, d_S is not neutral, which can be seen by considering the transposition that swaps the second and third bits among themselves. Nonetheless, $\Delta^{d_S, \Sigma}$ satisfies the IC postulates [11]. \square

We conclude by a few comments on our results. First, notice that Neutrality for distance-based operators depends only on the distance used and not on the aggregation function. Concerning the connection between our results and social choice, at first glance it might look disappointing that there is no IC operator satisfying SC-Majority, but it should be kept in mind that there are also important voting rules (e.g., Borda) which do not satisfy this property. Furthermore, it is a positive result that Participation holds for all IC operators, as this is not the case for important voting rules (e.g., Copeland, Dodgson and Young). Having Participation removes an agent's incentive for strategizing about whether to cast a vote. Also, it is not overly surprising that Independence of clones does not hold for all IC operators as it does not hold for many voting rules either (e.g., Plurality, Borda, Copeland and Dodgson). Finally, the result on Consistency is notable as this property does not hold for several important voting rules (e.g., Copeland, Dodgson and Young).

Note that we consider the strongest setting, where the constraint μ is unrestricted and properties have to hold for any μ and any profile. The cases when either domain restrictions are imposed on μ , or $\mu \equiv \top$, remain to be explored. Some of the proofs will carry over, whereas several results will have to be revisited.

5 Conclusion

In this work we have investigated eighteen desirable properties for belief merging operators, fourteen of which are newly formulated using insights from voting theory. We show that some follow from the IC postulates, some can never be satisfied by an IC operator, whereas others are only satisfied by certain IC operators. For properties of the last case, we additionally verified which of the standard operators satisfy them. If a property already follows from the postulates this is good news; if it does not, this shows that special care is needed when designing tailor-made operators. The properties proposed in this work are, however, to be seen as a first step on a long path and can certainly be refined and extended.

Likewise, there are quite a number of possible directions for future work. A natural step is to perform an extensive classification which is not limited to standard operators. One could also have a closer look at operators for fragments such as Horn, whenever a property is not satisfied in the general setting. Furthermore, for each considered property it is enticing to come up with a representation theorem for the setting where the IC postulates are extended by this property. Certainly also the relations between the properties studied in this work deserve a closer investigation. In particular, it would be interesting to come up with (IC) merging operators satisfying a maximal number of properties and to complement these results with impossibility theorems for the remaining cases. Also, as discussed above, the role of the constraint formula μ deserves a closer investigation, in which domain restrictions of μ are considered. Last but not least, we plan to explore the relation between judgment aggregation and belief merging—for the general case this relation was recently studied by Everaere, Konieczny, and Marquis [9]—with a special focus on devising new suitable properties.

Acknowledgments

This work was supported by the Austrian Science Fund (FWF) under grants P25518 and P25521, and the German Research Foundation (DFG): ER 738/2-1.

REFERENCES

- [1] *Handbook of Computational Social Choice*, eds., Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, Cambridge University Press, 2016.
- [2] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet, 'Preference handling in combinatorial domains: From AI to social choice', *AI Magazine*, **29**(4), 37–46, (2008).
- [3] Samir Chopra, Aditya K. Ghose, and Thomas A. Meyer, 'Social choice theory, belief merging, and strategy-proofness', *Information Fusion*, **7**(1), 61–79, (2006).
- [4] Amílcar Mata Díaz and Ramón Pino Pérez, 'Logic-based fusion of complex epistemic states', in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 11th European Conference, ECSQARU. Proceedings*, volume 6717 of *Lecture Notes in Computer Science*, pp. 398–409. Springer, (2011).
- [5] Daniel Eckert and Gabriella Pigozzi, 'Belief merging, judgment aggregation and some links with social choice theory', in *Belief Change in Rational Agents: Perspectives from Artificial Intelligence, Philosophy, and Economics*, volume 05321 of *Dagstuhl Seminar Proceedings*. IBFI, Schloss Dagstuhl, (2005).
- [6] Ulle Endriss, 'Applications of logic in social choice theory - (invited talk)', in *Computational Logic in Multi-Agent Systems - 12th International Workshop, CLIMA XII. Proceedings*, volume 6814 of *Lecture Notes in Computer Science*, pp. 88–91. Springer, (2011).
- [7] Patricia Everaere, Sébastien Konieczny, and Pierre Marquis, 'The strategy-proofness landscape of merging', *Journal of Artificial Intelligence Research (JAIR)*, **28**, 49–105, (2007).
- [8] Patricia Everaere, Sébastien Konieczny, and Pierre Marquis, 'On egalitarian belief merging', in *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014*. AAAI Press, (2014).
- [9] Patricia Everaere, Sébastien Konieczny, and Pierre Marquis, 'Belief merging versus judgment aggregation', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015*, pp. 999–1007. ACM, (2015).
- [10] Dov M. Gabbay, Odinaldo Rodrigues, and Gabriella Pigozzi, 'Connections between belief revision, belief merging and social choice', *Journal of Logic and Computation*, **19**(3), 445–446, (2009).
- [11] Adrian Haret, Stefan Rümmele, and Stefan Woltran, 'Merging in the Horn fragment', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, pp. 3041–3047. AAAI Press, (2015).
- [12] Sébastien Konieczny and Ramón Pino Pérez, 'Propositional belief base merging or how to merge beliefs/goals coming from several sources and some links with social choice theory', *European Journal of Operational Research*, **160**(3), 785–802, (2005).
- [13] Sébastien Konieczny and Ramón Pino Pérez, 'Merging information under constraints: A logical framework', *Journal of Logic and Computation*, **12**(5), 773–808, (2002).
- [14] Sébastien Konieczny and Ramón Pino Pérez, 'Logic Based Merging', *Journal of Philosophical Logic*, **40**(2), 239–270, (2011).
- [15] Pedrito Maynard-Zhang and Daniel J. Lehmann, 'Representing and aggregating conflicting beliefs', *Journal of Artificial Intelligence Research (JAIR)*, **19**, 155–203, (2003).
- [16] *Economics and Computation*, ed., Jörg Rothe, Springer, 2015.
- [17] Nicolaus Tideman, *Collective Decisions and Voting: The Potential for Public Choice*, Ashgate, 2006.

Schema-Based Debugging of Federated Data Sources

Andreas Nolle¹ and Christian Meilicke² and Melisachew Wudage Chekol² and German Nemirovski¹ and Heiner Stuckenschmidt²

Abstract. Information explosion leads to continuous growth of data distributed over different data sources. However, the increasing number of data sources increases the risk of inconsistency. In such a federative setting, description logics can be applied to define a central schema that serves as a conceptual view comprising and extending the semantics of each data source. Consequently, each data source is treated as a single knowledge base that is integrated in a federated knowledge base. Following this idea, we propose an approach for automated debugging of federated knowledge bases that targets the identification and repair of inconsistency. We report on experiments with a large distributed dataset from the domain of library science.

1 Introduction

The Linked Open Data (LOD) cloud grows continuously. However, the more data is available, the higher is the probability of inconsistency. Besides local clashes within data sources, two (or more) data sources can be contradictory. In the context of library science, for instance, one data source may catalogue a publication correctly as a paper, whereas in another source it is mistakenly defined as proceedings. One approach to tackle this problem is the use of federated data or information integration, where a central schema serves as a conceptual view that comprises and extends the semantics of each integrated data source. As a consequence, the central schema and its mappings to the different schemas which are used in the integrated data sources represent the interface to the distributed data. By using these mappings, original queries (mainly referring to the central schema) can be transformed into queries referring to the related schema of each data source. Thus, clients do not have to be aware of the local schema in each integrated data source [24].

Let us illustrate this by the following example. We will use this example throughout the remainder of the paper.

Example 1 Let \mathcal{T} be a central schema and \mathcal{A}_1 , \mathcal{A}_2 , and \mathcal{A}_3 denote three distributed data sources. \mathcal{T} contains the following axioms.

$$\begin{array}{ll} \text{Book} \sqsubseteq \text{Paper} \sqsubseteq \text{Publication} & \text{Paper} \sqsubseteq \neg \text{Book} \\ \text{Proceedings} \sqsubseteq \text{Book} & \text{Book} \sqsubseteq \text{Paper} \sqsubseteq \neg \text{SlideSet} \\ \exists \text{isPartOf} \sqsubseteq \text{Paper} & \exists \text{isPartOf}^- \sqsubseteq \text{Proceedings} \end{array}$$

The three data sources contain the following assertions:

\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_3
$\text{Paper}(\mathbf{I1})$	$\text{Paper}(\mathbf{I1})$	$\text{SlideSet}(\mathbf{I1})$
$\text{isPartOf}(\mathbf{I1}, \mathbf{AI15})$	$\text{Proceedings}(\mathbf{I1})$	$\text{SlideSet}(\mathbf{I2})$
$\text{Paper}(\mathbf{I2})$	$\text{isPartOf}(\mathbf{AI15}, \mathbf{I1})$	

The assertion that $\mathbf{I1}$ is a Paper (in \mathcal{A}_1) and the assertion that $\mathbf{I1}$ is a SlideSet (in \mathcal{A}_3) are obviously in contradiction due to the axiom $\text{Paper} \sqsubseteq \neg \text{SlideSet}$ in \mathcal{T} . In addition, as the assertion $\text{Paper}(\mathbf{I1})$ can also be found in \mathcal{A}_2 , it is also contradictory to \mathcal{A}_3 . Furthermore, we can entail this assertion in \mathcal{A}_1 from $\text{isPartOf}(\mathbf{I1}, \mathbf{AI15})$ and the axiom $\exists \text{isPartOf} \sqsubseteq \text{Paper}$ in \mathcal{T} .

Note that our example can easily be extended to the case where the integrated data sources use different terminologies that are linked by equivalence or subsumption axioms to an intermediary schema. Without loss of generality, we will in the remainder of this paper assume that there is only one central schema \mathcal{T} which might be the union of some data source specific schemas and an intermediary one containing mappings between the data source specific vocabularies. Furthermore, in our work, we will not address integration problems related to the incoherency of \mathcal{T} , i.e., we assume that \mathcal{T} is coherent. Note that there are other works that deal with debugging issues on the terminological level, e.g., [10].

The main contribution of our approach is to exploit explicit but also implicit redundancies caused by federating different sources to verify or disprove assertions that are involved in logical conflicts and to propose a resolution of these conflicts. In a setting with two or more data sources, where each data source contains several thousand assertions, it is challenging to propose a solution that takes the dependencies between the involved conflicts in an appropriate way into account. Based on techniques like query expansion (backward-chaining) and by identifying inconsistencies via clash queries we first collect all logical conflicts then we apply a two-phase debugging algorithm to resolve the previously collected conflicts.

In particular, we apply a majority voting scheme. Based on this approach we are able to resolve a subset of all conflicts in the first phase of our debugging algorithm. The second phase uses the outcome of the first phase to deduce a data source specific measure of trust for certain types of assertions. Repairs of additional conflicts can then be generated based on the statistical evidences gathered in the first phase. We argue why our algorithm generates reasonable repair plans and evaluate our approach against a large distributed LOD dataset from the domain of library science.

The rest of the paper is organized as follows. In Section 2 we introduce $DL\text{-Lite}_A$ as well as some fundamental terms and definitions related to inconsistency detection in federated $DL\text{-Lite}_A$ knowledge bases and conjunctive queries. In Section 3 we recall and extend our previous approach of inconsistency detection in federated $DL\text{-Lite}_A$ knowledge bases. Subsequently, we propose our algorithm for the generation of repairs in Section 4 comprising the two phases of resolvable and learned repairs. In Section 5 we discuss some evaluation results of our experiments with a large distributed LOD dataset. Before concluding in Section 7, we compare approaches related to our work in Section 6.

¹ Albstadt-Sigmaringen University, Germany, email: {nolle, nemirovskij}@hs-alsig.de

² Research Group Data and Web Science, University of Mannheim, Germany, email: {christian, mel, heiner}@informatik.uni-mannheim.de

2 Preliminaries

We briefly introduce our definition of federated $DL\text{-Lite}_A$ knowledge bases (KBs), discuss basic notions related to inconsistency in DL KBs, and describe conjunctive queries over $DL\text{-Lite}_A$ KBs.

2.1 Federated $DL\text{-Lite}_A$ Knowledge Bases

$DL\text{-Lite}$ is a family of lightweight description logics proposed by Calvanese et al. [4] with the aim to find a trade-off between expressiveness and reasoning complexity. This resulted in a family of languages comprising various $DL\text{-Lite}$ logics where reasoning, such as traditional DL reasoning services like checking KB satisfiability, can be done in PTIME in the of size of the TBox and query answering in AC^0 in the size of the ABox. Furthermore, it has been shown that members of the $DL\text{-Lite}$ family are one of the maximal logics that allow first-order logic (FOL)-rewritability of conjunctive query answering and therewith a processing of query answering through standard database technology. For this study, we consider the sub-family $DL\text{-Lite}_A$, which has been especially designed for dealing efficiently with huge amounts of extensional information.

In $DL\text{-Lite}_A$ concept, role, value-domain, and attribute expressions are formed according to the following syntax:

$$\begin{aligned} B &::= \perp_C \mid A \mid \exists Q \mid \delta(U) & E &::= \rho(U) \\ C &::= \top_C \mid B \mid \neg B \mid \exists Q.C & F &::= \top_D \mid T_1 \mid \dots \mid T_n \\ Q &::= P \mid P^- & V &::= U \mid \neg U \\ R &::= Q \mid \neg Q \end{aligned}$$

where \top_C denotes the *top* or *universal concept*, \perp_C the *bottom* or *empty concept*, A an *atomic concept*, B a *basic concept* and C a *general concept*. Similar to that, we have *atomic roles* denoted by P , *basic roles* by Q and *general roles* by R . *Atomic attributes* are represented by U and *general attributes* by V whereas E denotes a *basic value-domain* and F a *value-domain expression*. Furthermore, $\exists Q$ (*unqualified existential restrictions*) represent objects that are related by role Q to some objects, $\exists Q.C$ (*qualified existential restrictions*) denote objects that are related by Q to objects denoted by concept C , \neg denotes the negation of concepts, roles or attributes and P^- is used to represent the inverse of role P . Concerning an attribute U its *domain* is denoted by $\delta(U)$ and its *range* (set of values) by $\rho(U)$. *Value domains* are represented by $T_1 \mid \dots \mid T_n$, where each T_i denotes a pairwise disjoint datatype of values and \top_D the *universal value-domain* [4, 18]. In $DL\text{-Lite}_A$ a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ consists of a TBox \mathcal{T} also known as schema, and an ABox \mathcal{A} , the extensional knowledge part which represents a data source.

The TBox \mathcal{T} contains a set of axioms of the form

$$B \sqsubseteq C \quad Q \sqsubseteq R \quad E \sqsubseteq F \quad U \sqsubseteq V \quad (\text{funct } Q) \quad (\text{funct } U)$$

and the ABox \mathcal{A} is a finite set of assertions of the form

$$A(a) \quad P(a, b) \quad U(a, v).$$

TBox assertions of the form $B \sqsubseteq C$ denotes *concept inclusions*, $Q \sqsubseteq R$ *role inclusion*, $E \sqsubseteq F$ *value-domain inclusion* and $U \sqsubseteq V$ *attribute inclusion*. *Functionality assertions* on roles and attributes in \mathcal{T} are denoted by $(\text{funct } Q)$ and $(\text{funct } U)$, respectively. TBox assertions of the form $B_1 \sqsubseteq B_2$ and $Q_1 \sqsubseteq Q_2$ are called *positive inclusions (PI)* whereas $B_1 \sqsubseteq \neg B_2$ and $Q_1 \sqsubseteq \neg Q_2$ *negative inclusions (NI)*. For ABox assertions a and b represent object constants and v represents a value constant.

The semantics of $DL\text{-Lite}_A$ is given in terms of an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ (the domain) is a disjoint union of the two non-empty sets $\Delta_{\mathcal{O}}^{\mathcal{I}}$, the domain of objects, and $\Delta_{\mathcal{V}}^{\mathcal{I}}$, the domain of

values; and $\cdot^{\mathcal{I}}$ (the interpretation function) that maps each element in the signature Σ (also known as alphabet or vocabulary) to a subset of $\Delta_{\mathcal{O}}^{\mathcal{I}}$ and each value domain to a subset of $\Delta_{\mathcal{V}}^{\mathcal{I}}$. $DL\text{-Lite}_A$ adopts the unique name assumption (UNA), meaning that for every interpretation \mathcal{I} and constant pair $c_1 \neq c_2$, we have $c_1^{\mathcal{I}} \neq c_2^{\mathcal{I}}$. This means that different constant names (encoded as IRIs) are interpreted differently and refer to different individuals. In terms of further semantics we refer to the more precise definitions given in [4, 18].

In the context of federated settings, where each integrated data source uses different terminologies that are linked by an intermediary (central) schema, we can define a federated $DL\text{-Lite}_A$ KB as well as federated ABox assertions as follows:

Definition 1 *A federated $DL\text{-Lite}_A$ knowledge base is a $DL\text{-Lite}_A$ knowledge base \mathcal{K} with $\mathcal{K} = \langle \mathcal{T}_c \cup \bigcup_{i \in \mathbb{F}} \mathcal{T}_i, \bigcup_{i \in \mathbb{F}} \mathcal{A}_i \rangle$ where \mathcal{T}_c is a central TBox, each \mathcal{T}_i is a TBox and \mathcal{A}_i is an ABox in data source i and \mathbb{F} is a set of indices that refers to the federated data sources. A federated ABox assertion is a pair $\langle \alpha, i \rangle$ where α denotes an ABox assertion stated in \mathcal{A}_i . For compact presentation we will write only \mathcal{T} instead of $\mathcal{T}_c \cup \bigcup_{i \in \mathbb{F}} \mathcal{T}_i$ and \mathcal{A} instead of $\bigcup_{i \in \mathbb{F}} \mathcal{A}_i$ for the rest of this paper.*

2.2 Inconsistency in Description Logics

In description logics, an interpretation \mathcal{I} that satisfies all KB assertions in $\mathcal{T} \cup \mathcal{A}$ is called a *model*. The set of all models of \mathcal{K} is denoted by $Mod(\mathcal{K})$ and if $Mod(\mathcal{K}) \neq \emptyset$, we call \mathcal{K} *satisfiable* or *consistent* [2, 7]. Otherwise \mathcal{K} is called *inconsistent*. $\mathcal{K} \models \phi$ denotes that \mathcal{K} logically entails or satisfies a closed first-order logic sentence (formula) ϕ , if $\phi^{\mathcal{I}}$ is true for every $\mathcal{I} \in Mod(\mathcal{K})$. If a set of closed sentences denoted by F is entailed by \mathcal{K} , we can also write $\mathcal{K} \models F$ [21]. According to Kalyanpur et al. [11] an *explanation* (or justification) for $\mathcal{K} \models \phi$ is a subset \mathcal{K}' of \mathcal{K} such that $\mathcal{K}' \models \phi$ while $\mathcal{K}'' \not\models \phi$ for all $\mathcal{K}'' \subset \mathcal{K}'$. An explanation can be understood as a minimal reason that explains why ϕ follows from \mathcal{K} . Analogously, given an inconsistent knowledge base \mathcal{K} , we are interested in explanations for the inconsistency, i.e., minimal subsets \mathcal{K}' of \mathcal{K} such that $Mod(\mathcal{K}') = \emptyset$. More precisely, a minimal inconsistent subset (*MIS*) denoted by \mathcal{K}' is a subset of \mathcal{K} such that \mathcal{K}' is inconsistent while \mathcal{K}'' is consistent for all $\mathcal{K}'' \subset \mathcal{K}'$. From our running example, we can see that $\langle \mathcal{T}, \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \rangle$ is an inconsistent KB.

Example 2 *One of the explanations for the inconsistency mentioned in Example 1 is the set*

$$\{isPartOf(\mathbf{I1}, \mathbf{AI15}), SlideSet(\mathbf{I1}), \\ Paper \sqsubseteq \neg SlideSet, \exists isPartOf \sqsubseteq Paper\}.$$

A subset $\mathcal{R} \subseteq \mathcal{K}$ is called a *repair* (or repair plan) of \mathcal{K} , if \mathcal{K} is inconsistent and if $\mathcal{K} \setminus \mathcal{R}$ is consistent. As shown in [20], a hitting set over all MISs is a repair. Note that there is always a trivial repair $\mathcal{R} = \mathcal{K}$. However, we are especially interested in those repairs that remove a minimal number of assertions, i.e., \mathcal{R} is a minimal repair if \mathcal{R} is a repair and each proper subset of \mathcal{R} is not a repair.

2.3 Conjunctive Queries

A *conjunctive query (CQ)* q over a KB \mathcal{K} is a Datalog expression of the form $q(\mathbf{x}) \leftarrow conj(\mathbf{x}, \mathbf{y})$. $conj(\mathbf{x}, \mathbf{y})$ denotes the *body* of q and is a conjunction of *atoms* of the form $A(x)$, $P(x, y)$, $x = y$, or $x \neq y$ in which x and y are either constants in \mathcal{K} or variables in \mathbf{x} or \mathbf{y} , and A is a concept name or value-domain in \mathcal{K} and P is a role or attribute

name in \mathcal{K} . In addition, \mathbf{x} are *distinguished variables* that are part of the *head* $q(\mathbf{x})$ of a query q whereas \mathbf{y} are *non-distinguished variables* and do not occur in the head. If a variable does not correspond to the set of distinguished variables and does not occur in at least two query atoms, the variable is called *unbound* and is denoted by $_$. *Unions of conjunctive queries* (UCQ) are denoted by the expressions $q(\mathbf{x}) \leftarrow \text{conj}_1(\mathbf{x}, \mathbf{y}_1), \dots, q(\mathbf{x}) \leftarrow \text{conj}_n(\mathbf{x}, \mathbf{y}_n)$, where each $\text{conj}_i(\mathbf{x}, \mathbf{y}_i)$ is a conjunctive query.

Example 3 *The following query, over the KB in Example 1, selects all papers that have been published in proceedings:*

$$q(x) \leftarrow \text{Paper}(x), \text{isPartOf}(x, _).$$

3 Inconsistency Detection

This section recalls our previous approach of efficiently detecting inconsistency in federated KBs as first presented in [17]. We extend this approach and focus on the generation of federated explanations.

3.1 Inconsistency Detection in $DL\text{-Lite}_{\mathcal{A}}$

To determine if a KB is consistent or not, we have to search for ABox assertions, that are in conflict with the TBox or that are contradicting each other given the TBox. Lembo et al. [14] identified a complete set of six different patterns that cause clashes in $DL\text{-Lite}_{\mathcal{A}}$ KBs:

- an instantiation of an unsatisfiable (incoherent) concept, role or attribute such that $\mathcal{T} \models A \sqsubseteq \neg A$ and $A(a) \in \mathcal{A}$ (respectively $\mathcal{T} \models P \sqsubseteq \neg P$ and $P(a, b) \in \mathcal{A}$ for roles, and $\mathcal{T} \models U \sqsubseteq \neg U$ and $U(a, v) \in \mathcal{A}$ for attributes)
- ABox assertions contradicting axioms that restrict the interrelation of individuals such that $\mathcal{T} \models P \sqsubseteq \neg P^-$ or $\mathcal{T} \models \exists P \sqsubseteq \neg \exists P^-$ and $P(a, a) \in \mathcal{A}$
- incorrect datatypes such that $\mathcal{T} \models \rho(U) \sqsubseteq T$, $U(a, v) \in \mathcal{A}$ and $v^{\mathcal{I}} \notin T^{\mathcal{I}}$
- ABox assertions contradicting negative inclusions such that, e.g., $\mathcal{T} \models A \sqsubseteq \neg \exists P$ and $\{A(a), P(a, b)\} \subseteq \mathcal{A}$
- ABox assertions contradicting role functionality such that $(\text{funct } P) \in \mathcal{T}$ and $\{P(a, b_1), P(a, b_2)\} \subseteq \mathcal{A}$, where $b_1 \neq b_2$ (respectively $(\text{funct } P^-) \in \mathcal{T}$ and $\{P(a_1, b), P(a_2, b)\} \subseteq \mathcal{A}$, where $a_1 \neq a_2$, for the functionality of a inverse role)
- ABox assertions contradicting attribute functionality such that $(\text{funct } U) \in \mathcal{T}$ and $\{U(a, v_1), U(a, v_2)\} \subseteq \mathcal{A}$, where $v_1 \neq v_2$

The distribution of huge amounts of data over several sources makes state of the art methods for inconsistency detection, such as tableau-based reasoning algorithms, hardly applicable (see [17]), since they mostly require to have all the data in one place. We propose an alternative approach, comprising the formulation and evaluation of *federated clash queries*.

3.2 Clash Query Generation

Based on the clash definitions given above and referring to the work of Calvanese et al. [4] we define a mapping function φ to generate queries for inconsistency detection out of relevant axioms from \mathcal{T} . The function φ maps concepts, roles and attributes into query atoms as follows:

$$\begin{array}{lll} A \mapsto A(x) & \delta(U) \mapsto U(x, _) & P \mapsto P(x, y) \\ \exists P \mapsto P(x, _) & \rho(U) \mapsto U(_, y) & P^- \mapsto P(y, x) \\ \exists P^- \mapsto P(_, x) & T \mapsto T \end{array}$$

Based on φ , the clash types (a)–(f) can be mapped into queries (i)–(vi) as shown below:

- $\varphi(A \sqsubseteq \neg A) = q(x) \leftarrow \varphi(A)$,
 $\varphi(X \sqsubseteq \neg X) = q(x, y) \leftarrow \varphi(X)$, where $X \in \{P, U\}$,
- $\varphi(P \sqsubseteq \neg P^-) = q(x, y) \leftarrow \varphi(P), \varphi(P^-)$ and
 $\varphi(\exists P \sqsubseteq \neg \exists P^-) = q(x) \leftarrow \varphi(\exists P), \varphi(\exists P^-)$,
- $\varphi(\rho(U) \sqsubseteq T) = q(x, y) \leftarrow U(x, y), \text{datatype}(y) \neq T$,
- $\varphi(C \sqsubseteq \neg D) = q(x) \leftarrow \varphi(C), \varphi(D)$, where $C, D \in \{A, \exists P, \exists P^-, \delta(U)\}$,
 $\varphi(R \sqsubseteq \neg S) = q(x, y) \leftarrow \varphi(R), \varphi(S)$ and
 $\varphi(V_1 \sqsubseteq \neg V_2) = q(x, y) \leftarrow \varphi(V_1), \varphi(V_2)$,
- $\varphi(\text{funct } P) = q(x, y, z) \leftarrow P(x, y), P(x, z), y \neq z$ and
 $\varphi(\text{funct } P^-) = q(x, y, z) \leftarrow P(y, x), P(z, x), y \neq z$, and
- $\varphi(\text{funct } U) = q(x, y, z) \leftarrow U(x, y), U(x, z), y \neq z$,

where A, P and U denote an atomic concept, an atomic role, and an atomic attribute; $x, y, z, _$ are variables; $\exists P, \exists P^-$ and $\delta(U)$ are concepts; R and S are roles; V_1 and V_2 are attributes; $\text{datatype}(y)$ is an external function which computes the datatype of a given data value y ; and T denotes a datatype of values, where each different datatypes are pairwise disjoint. Except for the clash queries in (i), the queries in (ii)–(vi) contain two atoms and an inequality constraint in (v) and (vi) used as filters applied to the query answers. We refer to the queries in (ii)–(vi) as *two-atom queries*.

Example 4 *For the axiom $\text{Paper} \sqsubseteq \neg \text{Book}$ in Example 1, the mapping function φ generates the clash query: $\varphi(\text{Paper} \sqsubseteq \neg \text{Book}) = q(x) \leftarrow \text{Paper}(x), \text{Book}(x)$.*

According to the definition above, the mapping function φ generates UCQs that may contain inequalities (because of clash queries in (v) and (vi)), that in general makes query answering intractable. However, since those queries are of fixed length (two atoms and an inequality expression), the complexity of checking KB satisfiability by a reduction into query answering is in AC^0 in the size of the ABox and NLOGSPACE in the size of the KB as shown by Artale et al [1].

3.3 Clash Query Expansion

To ensure that all implicit knowledge is taken into consideration when computing the answers, the original query is expanded. The resulting set of expanded queries (UCQs) will contain atoms addressing all possible concepts, roles and attributes that implicitly provide individuals of the originally requested type. For $DL\text{-Lite}_{\mathcal{A}}$ KBs query expansion (backward-chaining) of a general UCQ can be efficiently done in PTIME in the size of the TBox [4].

Definition 2 *Given a TBox \mathcal{T} and a query $q(\mathbf{x})$ in the signature of \mathcal{T} . An expansion of $q(\mathbf{x})$ is a UCQ denoted by $q_{\text{exp}}(\mathbf{x}) = \bigcup_i q_i(\mathbf{x})$, that is a rewriting of $q(\mathbf{x})$ w.r.t. \mathcal{T} , such that $\langle \mathcal{T}, \mathcal{A} \rangle \models q(\mathbf{a})$ iff $\mathcal{A} \models q_{\text{exp}}(\mathbf{a})$, for any ABox \mathcal{A} and any tuple \mathbf{a} of individuals in \mathcal{A} .*

Example 5 *The expansion of the clash query in Example 4 is the following UCQ:*

$$\begin{array}{l} q_{\text{exp}}(x) \leftarrow \text{Paper}(x), \text{Book}(x), \\ q_{\text{exp}}(x) \leftarrow \text{Paper}(x), \text{Proceedings}(x), \\ q_{\text{exp}}(x) \leftarrow \text{Paper}(x), \text{isPartOf}(_, x), \\ q_{\text{exp}}(x) \leftarrow \text{isPartOf}(x, _), \text{Proceedings}(x), \\ q_{\text{exp}}(x) \leftarrow \text{isPartOf}(x, _), \text{Book}(x), \\ q_{\text{exp}}(x) \leftarrow \text{isPartOf}(x, _), \text{isPartOf}(_, x) \end{array}$$

Subsumption axioms in $DL\text{-Lite}_{\mathcal{A}}$ comprise only one element on the left and one element on the right hand side, or can be normalized to that form. Consequently, an expansion of a clash query is a UCQ

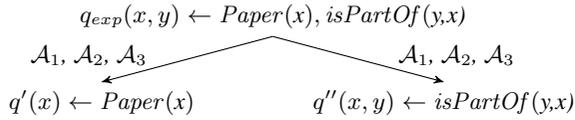
where each conjunct has one or at most two query atoms and an inequality constraint.

Since q_{exp} may comprise query atoms containing unbound variables, we replace those variables by new distinguished variables so as to make precise distinctions between different instantiations of them.

3.4 Generating Federated Explanations

Computing all inconsistency explanations, introduced earlier as MIS, requires to generate and expand all clash queries. The resulting UCQs comprise not only the semantics of the central TBox but also of each integrated data source. Thus, each atom of those queries may address several data sources. Local MISs are detected by evaluating the expanded queries directly against a data source. Federated MISs are detected by distributing the atoms of the expanded queries as atomic queries to the data sources. Consequently, we apply a simple federation algorithm where each query atom is evaluated at all data sources.

Example 6 According to that, the following federated queries will be generated for the expanded query $q_{exp}(x,y) \leftarrow Paper(x), isPartOf(y,x)$ (third conjunct of Example 5):



The results of these queries are tuples of instances. However, we are not interested in tuples of instances, but in the inconsistency explanations (MIS) that can be derived from these results. Because of that, the results of query pairs have to be joined and converted to federated ABox assertions (see Example 7). Moreover, since we assume that all of the terminological axioms are correct, we are only targeting the subset of a MIS that contain only ABox assertions. In the following we refer to such a subset of a MIS as a MISA (minimal inconsistency preserving sub-ABox).

Example 7 The set of MISAs resulting from the evaluation of the query in Example 6 is the set

$$\{ \{ \langle Paper(\mathbf{I1}), 1 \rangle, \langle isPartOf(\mathbf{AI15}, \mathbf{I1}), 2 \rangle \}, \\
 \{ \langle Paper(\mathbf{I1}), 2 \rangle, \langle isPartOf(\mathbf{AI15}, \mathbf{I1}), 2 \rangle \} \}.$$

None of these operations has an impact on the complexity which remains in AC^0 in the size of the ABox and in $NLOGSPACE$ in the size of the whole KB, given a fixed set of data sources.

4 Repair Plan Generation

The generation of a repair plan is divided into two phases. In the first phase we propose a partial repair plan following a simple majority voting scheme (Section 4.1), while in the second phase (Section 4.2) we try to repair the remaining conflicts following an approach guided by the statistics gathered in the first phase.

4.1 Phase 1: Majority Voting

To resolve the identified contradictions we follow the assumption that the more data sources are integrated, the higher is the probability that correct assertions occur redundantly. Conversely, the probability that an assertion is incorrect correlates with the number of contradictions in which the assertion is involved.

Based on this assumption, we propose a greedy approach, given in Algorithm 1, for generating repairs. The algorithm starts with the

Algorithm 1: GenerateResolvableRepairs(C)

Output: (partial) repair \mathcal{R} resolved by majority voting

```

begin
   $\mathcal{R} \leftarrow \emptyset$ 
   $\mathcal{C}_{unary} \leftarrow \text{GetSingletonMISA}(C)$ 
  foreach  $c \in \mathcal{C}_{unary}$  do
     $\mathcal{R} \leftarrow \mathcal{R} \cup \text{GetAssertion}(c)$ 
     $\mathcal{C}_{resolved} \leftarrow \text{GetResolvedMISAs}(\text{GetAssertion}(c), C)$ 
     $C \leftarrow C \setminus \mathcal{C}_{resolved}$ 
  while true do
     $\mathcal{C}_{card} \leftarrow \text{DetermineCardinalities}(C)$ 
     $\mathcal{C}_x \leftarrow \text{GetResolvableMISAsWithMinCard}(\mathcal{C}_{card})$ 
    if  $\mathcal{C}_x = \emptyset$  then
      break
    foreach  $c \in \mathcal{C}_x$  do
       $\alpha \leftarrow \text{GetAssertionWithMaxCard}(c, \mathcal{C}_{card})$ 
       $\mathcal{R} \leftarrow \mathcal{R} \cup \alpha$ 
       $\mathcal{C}_{resolved} \leftarrow \text{GetResolvedExplanations}(\alpha, C)$ 
       $C \leftarrow C \setminus \mathcal{C}_{resolved}$ 
    return  $\mathcal{R}$ 
end

```

trivial repair of a singleton MISA (resulting from clash type (b) or (c)) by removing the only element in each singleton MISA. This repair can also have an influence on the remaining steps, because the element of a singleton MISA might also appear in a MISA with two elements. The remaining part of the algorithm deals with a non trivial repair of MISA with two elements. In the main loop the algorithm first counts for each assertion in how many different MISAs it occurs. We refer to the resulting number as a *cardinality of an assertion*. We also compute the *cardinality of a MISA* which is defined as the sum of the cardinalities of its two elements. We call MISAs that have elements with different cardinalities *resolvable MISAs*. With the help of a majority voting heuristic, we can make a decision in favour of one of the two elements of a resolvable MISA. We select all resolvable MISAs with minimum cardinality and remove from these MISAs the element with higher cardinality, which is the element that is involved in more conflicts. Note that we resolve MISAs with minimum cardinality first, to reduce the impact (of wrong decisions) on subsequent decisions. After each removal operation we update the remaining set of MISAs and repeat this procedure as long as resolvable MISAs can be found. The algorithm terminates when no resolvable MISAs are left to be repaired.

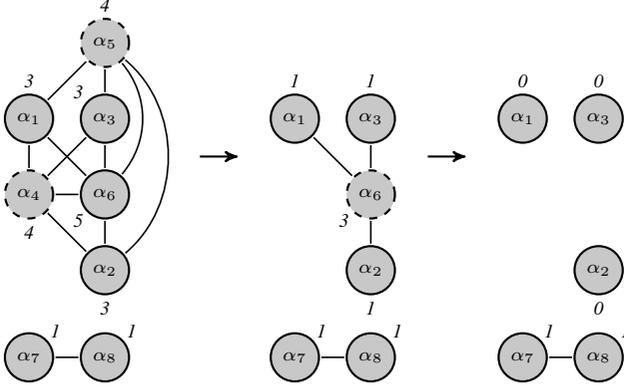
The algorithm is based on a heuristics that selects edges (MISAs) with minimal weight and removes from such edges the vertex (assertion) that is involved in more MISAs. Our algorithm runs in polynomial time with respect to the number of vertices. It does not guarantee, even for the resolvable cases, to construct a minimal vertex cover. The construction of a minimal vertex cover is known to be one of Karp's NP-complete problems [12].

Example 8 In our running example we have to deal with the federated assertions α_1 to α_8 listed as follows.

$$\begin{array}{ll}
 \alpha_1 = \langle Paper(\mathbf{I1}), 1 \rangle & \alpha_2 = \langle isPartOf(\mathbf{I1}, \mathbf{AI15}), 1 \rangle \\
 \alpha_3 = \langle Paper(\mathbf{I1}), 2 \rangle & \alpha_4 = \langle Proceedings(\mathbf{I1}), 2 \rangle \\
 \alpha_5 = \langle isPartOf(\mathbf{AI15}, \mathbf{I1}), 2 \rangle & \alpha_6 = \langle SlideSet(\mathbf{I1}), 3 \rangle \\
 \alpha_7 = \langle Paper(\mathbf{I2}), 1 \rangle & \alpha_8 = \langle SlideSet(\mathbf{I2}), 3 \rangle
 \end{array}$$

The set of MISAs \mathcal{C} for this example is $\{ \{ \alpha_1, \alpha_4 \}, \{ \alpha_1, \alpha_5 \}, \{ \alpha_1, \alpha_6 \}, \{ \alpha_2, \alpha_4 \}, \{ \alpha_2, \alpha_5 \}, \{ \alpha_2, \alpha_6 \}, \{ \alpha_3, \alpha_4 \}, \{ \alpha_3, \alpha_5 \}, \{ \alpha_3, \alpha_6 \}, \{ \alpha_4, \alpha_6 \}, \{ \alpha_5, \alpha_6 \}, \{ \alpha_7, \alpha_8 \} \}$. The assertion α_1 has a

cardinality of three because it appears in three MISAs; assertion α_4 has a cardinality of four. Thus, MISA $\{\alpha_1, \alpha_4\}$ has a cardinality of seven. In the following we represent each assertion as a vertex in a graph where each MISA is represented by an edge. This graph is shown in the following to illustrate the iterations of Algorithm 1. Note that we annotated the graph with assertion cardinalities, but omitted MISA cardinalities due to the lack of space.



As shown in the figure above, the algorithm needs three iterations to construct a repair, i.e., to construct a vertex cover for the corresponding conflict graph. Consequently, the resolvable repairs of our running example comprise α_4 , α_5 and α_6 . Assertion α_7 and α_8 yield an inconsistency, however, they are not conflicting with the other assertions. This inconsistency cannot be resolved by Algorithm 1, because α_7 and α_8 have the same cardinality which is one.

As illustrated in Example 8, the algorithm cannot resolve all logical conflicts. This will be the case, especially when the set of MISAs contains some MISAs that are unresolvable (by comprising elements having the same cardinalities) and are also not resolved during the process of executing Algorithm 1 due to an overlap with a resolvable MISA. Particularly, contradictory assertions of different values for a functional role or attribute result in MISAs that are predestinated to be unresolvable. How to deal with the remaining clashes is explained in the next section.

4.2 Phase 2: Learned Repairs

In the second phase we use the statistical evidence, that is implicitly available in the repair computed so far, to extend this repair. Suppose that a large fraction of the assertions of type $C(x)$ have been removed from a data source A_i , while most of the assertions $D(x)$ in A_j have not been removed. Now suppose that we have a MISA $\{\langle C(a), i \rangle, \langle D(b), j \rangle\}$. If we trust in the correctness of the repair that we conducted so far, we are justified in removing $\langle C(a), i \rangle$, because we have a higher trust in $\langle D(b), j \rangle$. Let us introduce the notion of trust formally.

Definition 3 Given a federated knowledge base $\mathcal{K} = \langle \mathcal{T}, \bigcup_{i \in \mathbb{F}} \mathcal{A}_i \rangle$, and a repair \mathcal{R} computed by Algorithm 1. Let σ be either a concept, a property or an attribute in the signature Σ of \mathcal{T} , and let $\Psi \subseteq \bigcup_{i \in \mathbb{F}} \mathcal{A}_i$ be a set of federated assertions, then $\text{sas}(\sigma, \Psi, i)$ is defined as the subset of assertions in Ψ that use σ and originate from A_i . The trust in σ with respect to i is defined as

$$\text{trust}(\sigma, i) = 1 - \frac{|\text{sas}(\sigma, \mathcal{R}, i)|}{|\text{sas}(\sigma, \bigcup_{i \in \mathbb{F}} \mathcal{A}_i, i)|}.$$

Based on this definition we define the trust of a federated assertion $\langle \alpha, i \rangle$ that uses σ as $\text{trust}(\langle \alpha, i \rangle) = \text{trust}(\sigma, i)$.

Example 9 From the repair of Example 8 it follows that we have $\text{trust}(\text{Paper}, 1) = 1 - \frac{0}{2} = 1$ and $\text{trust}(\text{SlideSet}, 3) = 1 - \frac{1}{2} = 0.5$. Thus, we remove α_8 as a learned repair due to the fact that α_7 has a higher trust.

This example shows how to apply the notion of trust to a single MISA, however, the set of all remaining MISAs might still contain overlapping MISAs. Therefore, we have to implement it as part of a more general algorithm. In [19] the authors proposed a linear algorithm for debugging terminological alignments. The proposed algorithm can be applied to any debugging scenario where a complete set of conflict sets, in our case the set of MISAs, is given. The algorithm, which we sketch in the following, requires a complete ordering of assertions that are involved in the remaining clashes. We derive this ordering from the trust values.

The input to Algorithm 2 are the unresolved MISAs \mathcal{C} , the previously computed repair \mathcal{R} , and a trust-ordered list $\mathcal{A}_{\text{trust}}$ of all assertions that occur in \mathcal{C} . The algorithm iterates over $\mathcal{A}_{\text{trust}}$ in descending order, thus, starting with an assertion α for which there is no assertion with higher trust. In each iteration the algorithm determines all those MISAs that contain α . For each such MISA $\{\alpha, \beta\}$ the assertion β is added to the repair \mathcal{R}' if the trust of β is lower than the trust of α . Thus, we finally remove β for the reason that we presented at the beginning of this section.

Algorithm 2: GenerateLearnedRepairs($\mathcal{C}, \mathcal{R}, \mathcal{A}_{\text{trust}}$)

Output: all learned repairs \mathcal{R}'

```

begin
   $\mathcal{R}' \leftarrow \emptyset$ 
  foreach  $\alpha \in \mathcal{A}_{\text{trust}}$  do
     $\mathcal{C}' \leftarrow \{\{x, y\} \in \mathcal{C} \mid \alpha \in \{x, y\}\}$ 
    foreach  $\{x, y\} \in \mathcal{C}'$  do
       $\beta \leftarrow x$  if  $x \neq \alpha$  otherwise  $y$ 
      if  $\text{trust}(\alpha) > \text{trust}(\beta)$  then
         $\mathcal{R}' \leftarrow \mathcal{R} \cup \beta$ 
         $\mathcal{C}_{\text{resolved}} \leftarrow \{\{x', y'\} \in \mathcal{C} \mid \beta \in \{x', y'\}\}$ 
         $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{C}_{\text{resolved}}$ 
  return  $\mathcal{R}'$ 
end
```

Algorithm 2 runs in the worst case in quadratic time with respect to the number of vertices \mathcal{C} (unresolved MISAs). Thus, the complexity of the whole federated debugging approach, starting from the generation of the clash queries up to the the generation of resolvable and learned repairs, runs in polynomial time.

Note that the algorithm will resolve all clashes if there is no MISA comprising two assertions with the same trust value. However, this will not always be the case. The repair of still remaining MISAs is not addressed in this paper. A possible extension of our approach could be the calculation of a general trust value for each data source over all of its assertions. Alternatively, a user could decide upon the problematic cases.

5 Experimental Evaluation

In order to evaluate the effectiveness of our approach we have set up a large distributed LOD dataset from the domain of library science. Specifically, we selected four LOD data sources, referred to as \mathcal{A}_1 to \mathcal{A}_4 in the following, and loaded their dumps into separate Virtuoso 7.2.2 instances (Open-Source Edition). In particular, these data

sources are FacetedDBLP³ (\mathcal{A}_1), BibSonomy⁴ (\mathcal{A}_2), RKB Explorer ePrints Open Archives⁵ (\mathcal{A}_3), and RKB Explorer DBLP⁶ (\mathcal{A}_4).

Note that our implementation relies on the usage of standard SPARQL interfaces and does not put any additional requirements on the data sources. Since the OWL 2 QL profile is based on *DL-Lite*, we have used it as specification language of our central TBox that includes the TBoxes of each data source. Note that we have applied some small modifications of the data source specific TBoxes to ensure that the federated TBox is coherent. Since the federated TBox lacks some negative inclusions and functionality assertions, we have added respective axioms to the central TBox.

In contrast to *DL-Lite_A* (see Section 2.1), the standard ontology language OWL, i.e., the OWL 2 QL profile, does not make the UNA, however, OWL provides the explicit object property `owl:sameAs` to express that two IRIs denote the same individual. Due to the fact that LOD sources following this strategy to link same individuals, we took the `owl:sameAs` assertions into account and modified our dataset such that all individuals representing the same entity also have the same IRI. Note that according to the work of Calvanese et al. [6] it is, under some restrictions, even possible to take into account `owl:sameAs` statements for query answering and retain the FOL-rewritability. But on grounds of simplicity of our experimental evaluation we embark on the strategy of resolving linked individuals by modifying the datasets. In addition to that, to gain a higher overlapping of the data sources we detected duplicates, especially by the unique attributes denoting the ISBN or the ISSN of a publication. The collection of the central TBox as well as the referenced TBoxes is available online⁷. For legal reasons we are currently not able to publish the final dataset of each integrated data source. Please contact us if you are interested in these datasets.

Based on the federated TBox our algorithm generates 422 clash queries, where 8 of which result from functionality assertions and 414 result from negative inclusions. Since some of those clash queries, i.e., the queries resulting from negative inclusions, can be implicitly derived by another clash query, the number of those clash queries is reduced to 281. The expansion of the remaining 289 clash queries results in 44,072 queries that have to be evaluated within the generation of explanations. Note that we do not consider clash queries of type (iii) in our evaluation, since they will produce only singleton MISAs (resulting from clash type (c): incorrect datatypes) whose resolution is trivial and not crucial in federated settings.

We have run our implementation of detecting and repairing inconsistency, called ClashSniffer, on a CentOS 6.7 virtual machine consisting of 4x Intel Xeon CPUs (à 4 cores @ 2.50 GHz) and 128 GB of RAM. The Virtuoso instances are hosted in an Ubuntu 14.04 LTS virtual machine with 6x Intel Xeon CPUs (à 4 cores @ 2.60 GHz) and 96 GB of RAM (16 GB of RAM are assigned to each Virtuoso instance). The runtime for inconsistency detection and the generation for appropriate explanations over all four data sources takes 80.1 min (minutes), where 56.5 min are required for evaluating the query parts. This runtime depends on the performance of the machines that host the data sources. The runtime for repair generation takes 6 min for the first phase and 12.2 min for the second phase.

Table 1 summarizes the characteristics of each data source and depicts the results of our experimental evaluation. Beside showing statistics for each data source on its own, the table shows two fed-

Table 1. Results of MISAs and Repairs

	#triples	#C -MISAs-	\mathcal{R} -resolvable repair-	\mathcal{R}' -learned repair-	rem. MISA rate
\mathcal{A}_1	72,372,256	3,266,765	46,128 (291,025)	1,187,461 (1,188,115)	54.72%
\mathcal{A}_2	17,765,873	1,096,337	4,654 (15,525)	246,289 (247,180)	76.0%
\mathcal{A}_3	166,320,474	12,016,391	1,024,414 (2,057,807)	420,081 (433,827)	79.26%
\mathcal{A}_4	27,897,291	26,504	521 (23,419)	4 (4)	11.62%
Σ	284,355,894	16,405,997	1,075,717 (2,387,776)	1,853,835 (1,869,126)	74.05%
\mathcal{F}	256,458,603	16,605,398	1,109,524 (4,770,357)	3,971,584 (10,368,294)	8.83%
\mathcal{F}'	284,355,894	18,146,950	1,993,136 (7,166,005)	3,267,659 (9,574,136)	7.75%

erated settings on which we have run our implementation of detecting and repairing inconsistency. We have defined the first federated setting \mathcal{F} that comprises data source \mathcal{A}_1 , \mathcal{A}_2 and \mathcal{A}_3 , whereas the second one, referred to as \mathcal{F}' , comprises all four data sources. The runtimes presented in the previous paragraph refer to the \mathcal{F}' setting, which comprises more than 284 million triples. We analyze these two settings in order to understand the impact of adding an additional data source, since we expect that the availability of an additional data source comprising complementary and potentially redundant information should have a positive impact not only on the quality of the repair but also on the quantity of MISAs solved by the repair. Note that we compare the two federated settings also against the local settings, where we apply the approach to each data source on its own. For that reason we have also added a row to the table headed with the Σ symbol, where we sum up the numbers for all single data sources.

The first data column of Table 1 illustrates the size of each data source and each federated KB, respectively. The second column depicts the number of detected clashes. The largest data source is ePrints Open Archives (\mathcal{A}_3) with more than 160 million triples and is also the data source with the highest number of local clashes. While local clashes are dominant in the dataset, we can also see that more conflicts can be detected by analyzing the data sources in a federated setting. The numbers increase from ≈ 16.4 (Σ) to ≈ 16.6 (\mathcal{F}) and to ≈ 18.1 million clashes (\mathcal{F}'). The clashes detected in \mathcal{F}' comprise 12,209,235 clashes (67.3%) that result from functionality assertions where 0.5% of these are federated and 5,937,715 clashes (32.7%) caused by negative inclusions with a rate of 28.3% federated clashes.

Figure 1 shows the number of all federated clashes (MISAs) and how they are distributed on the pairs of data sources in setting \mathcal{F}' . Note that each possible combination of data sources results in more than 40,000 clashes. Despite the fact that both data source \mathcal{A}_1 and \mathcal{A}_4 are based upon DBLP, it is interesting to see that these two data sources produce 77% of all federated clashes. A reason for this could be that the underlying DBLP dataset is parsed, converted, and interpreted differently and is mapped to distinct TBoxes.

The numbers in column three and four of Table 1 show the number of resolvable and learned repairs that are generated by our algorithm. The values in parenthesis represent the numbers of MISAs that are resolved by the generated repair. Note that this number is often significantly higher than the size of the repair, which indicates a high overlap of the MISAs. The last column comprises the rate of remaining clashes after our algorithm was applied. It is interesting to see that the rate of remaining MISAs in the federated settings \mathcal{F} and \mathcal{F}'

³ <http://dblp.l3s.de>

⁴ <http://www.bibsonomy.org>

⁵ <http://foreign.rkbexplorer.com>

⁶ <http://dblp.rkbexplorer.com>

⁷ <http://www.researchgate.net/publication/299852903>

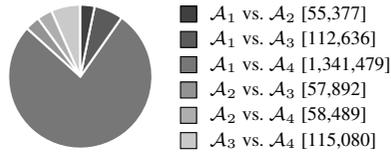


Figure 1. Distribution of Federated MISAs

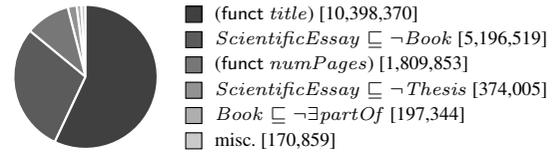


Figure 2. Axioms Causing Inconsistency

is significantly lower than applying our algorithm in a local setting of each single data source. Adding the additional data source \mathcal{A}_4 (\mathcal{F} vs. \mathcal{F}') results in a larger size of the resolvable repair but in a lower size of the learned repair. Moreover, this results in a higher number of new conflicts, but decreases also the relative number of remaining MISAs. Another positive effect is, that MISAs that are not resolvable in the first phase of \mathcal{F} are now solved in the first phase of \mathcal{F}' , why the number of the learned repair is decreased in \mathcal{F}' .

Table 2 highlights the data source specific impact of our approach. We compare the size of the local repair for each \mathcal{A}_i , taking only inconsistencies caused by assertions from \mathcal{A}_i into account, against the size of the subset of the federated repair restricted to assertions from \mathcal{A}_i . We conducted the comparison for both resolvable and learned repair of each federated setting \mathcal{F} and \mathcal{F}' . We can see, for example, that in the federated setting \mathcal{F} the size of the resolvable repair for \mathcal{A}_1 was increased by 13.59% and that the learned repair for \mathcal{A}_1 was increased by 66.95%. For this data source the effect of the additional data source in \mathcal{F}' is evidenced, since the size of the resolvable repair for \mathcal{A}_1 was increased by 1,896.71% compared to the local application of our approach at this data source. The reason for this significant increase is not only the fact that more MISAs of data source \mathcal{A}_1 can be solved (represented by the decreased rate of remaining MISAs), but also due to the decrease of the learned repair by -70.61% . Hence, more MISAs of this data source become resolvable, while in the federated setting \mathcal{F} these MISAs can be resolve only by the learned repair. This positive impact on the quality of the repair and also on the quantity of MISAs solved by the repair by an additional data source, comprising complementary and potentially redundant information, is also reflected by data source \mathcal{A}_2 . The last row shows the values based on summing up the results for all data sources. In average $+84.04\%$ are gained for the resolvable repair and this has again an impact measured in terms of $+68.55\%$ for the learned repair in \mathcal{F}' . Note that the federated setting has also an impact on reducing the number of local MISAs due to the fact that assertions from other data sources interfere with the assertions from local MISAs. Overall, the rate of remaining MISAs can be significantly reduced from originally 74.05% in local application (see Table 1) to 8.57% in federated setting \mathcal{F}' . These numbers illustrate the positive impact of the federated setting which allows to achieve a significantly higher recall rate for detecting problematic assertions.

Table 2. Impact of Federated Debugging

	\mathcal{F}			\mathcal{F}'		
	$\Delta\mathcal{R}$ resolvable repair	$\Delta\mathcal{R}'$ learned repair	rem. MISA rate	$\Delta\mathcal{R}$ resolvable repair	$\Delta\mathcal{R}'$ learned repair	rem. MISA rate
\mathcal{A}_1	+13.59%	+66.95%	24.37%	+1,896.71%	-70.61%	22.81%
\mathcal{A}_2	+236.53%	+3.1%	59.49%	+287.22%	+2.18%	59.49%
\mathcal{A}_3	+1.51%	+498.46%	0.15%	+1.52%	+500.83%	0.05%
\mathcal{A}_4	-	-	-	+37.24%	-100.0%	11.62%
Σ	+3.04%	+156.25%	8.95%	+84.04%	+68.55%	8.57%

To give an insight into the generated MISAs as well as the resolvable and learned repair for federated setting \mathcal{F}' we have done some further analysis. Starting with the generated MISAs, the axioms causing inconsistency are depicted in Figure 2. As already mentioned, most of the clashes result from functionality assertions, especially for the attribute *title*. Most of the clashes that are caused by a negative inclusion result from the axiom $ScientificEssay \sqsubseteq \neg Book$.

The computation of resolvable repair in the federated setting \mathcal{F}' comprised 413 iterations of the while loop in Algorithm 1. The highest cardinality found was 18,189. The reason for this high cardinality is that “*Bioinformatics*”⁸, which is a journal series comprising lots of articles assigned in data source \mathcal{A}_1 and \mathcal{A}_4 , is wrongly defined as an article in data source \mathcal{A}_3 .

After generating the resolvable repair the trust values for all concepts, roles and attributes occurring in unresolved MISAs are calculated with respect to each data source. The top 5 of lowest trust values derived are depicted in Table 3. Especially the two lowest trust values lead to the conclusion that assertions on *volume* attributes are probably misused in \mathcal{A}_1 and \mathcal{A}_3 . Having a more detailed look into the datasets of those sources confirms this conclusion, since *volume* attributes are in both data sources not used at the level of collections like proceedings, journals or books, but on the level of articles published in a collection. The low trust values for *volume* attributes reflect also the fact that the negative inclusion $ScientificEssay \sqsubseteq \neg Book$ is part of the top 5 axioms causing inconsistency (see Figure 2), since the expansion of *Book* comprise $\exists volume$.

Table 3. Top 5 of Lowest Trust Values

trust value	data source	$\sigma \in \Sigma$
0.1045	\mathcal{A}_3	http://purl.org/ontology/bibo/volume
0.2741	\mathcal{A}_1	http://swrc.ontoware.org/ontology#volume
0.8889	\mathcal{A}_1	http://swrc.ontoware.org/ontology#MasterThesis
0.9476	\mathcal{A}_2	http://swrc.ontoware.org/ontology#Booklet
0.9587	\mathcal{A}_2	http://swrc.ontoware.org/ontology#Unpublished

Finally, we have analyzed the remaining MISAs that are not solved by our approach. All of them are not federated and are exclusively caused by functionality assertions. Approximately, 53% of the remaining MISAs comprise the axiom (func *numPages*) and 47% the axiom (func *title*).

To evaluate the quality of the generated repairs 100 randomly selected MISAs of each phase in each federated setting are manually evaluated by three persons. Table 4 presents the precision of our debugging approach based on the sample we analyzed. If an URI is not accessible or at least two persons did not come to the same decision, the decision specific case is annotated as uncertain.

The evaluation results indicate a high precision of our approach and substantiate that the removal decisions are based on a reason-

⁸ <http://bioinformatics.oxfordjournals.org/>

Table 4. Quality of Repairs

		correct	incorrect	uncertain
resolvable repair	$\mathcal{R}_{\mathcal{F}}$	94%	0%	6%
	$\mathcal{R}_{\mathcal{F}'}$	97%	0%	3%
learned repair	$\mathcal{R}'_{\mathcal{F}}$	96%	0%	4%
	$\mathcal{R}'_{\mathcal{F}'}$	84%	2%	14%

able heuristics. The measured precision scores also confirm that the majority voting scheme underlying our approach is a valid premise which ensures also a high precision of the second phase, where we use the statistical evidence of the first phase to apply the notion of trust to the remaining MISAs. This is also suggested by the fact that the precision scores for resolvable and learned repairs are roughly in the same range. However, the subsets we analyzed are not comprehensive enough to enable a conclusion about the impact of adding an additional data source. The precision of the resolvable repair is slightly increased, at the same time we observe a marginal drop for the learned repair. While we cannot prove a positive impact on precision, the previously presented results in terms of detected MISAs and repair sizes have clearly shown the positive impact on recall.

6 Related Work

State-of-the-art DL reasoners that are used for inconsistency detection and its explanations basically process local KBs and are therefore inappropriate for distributed environments. Moreover, regardless of the supported language expressiveness or the underlying reasoning method (such as widely used tableau algorithms as in FaCT++ [23] or Pellet [22]; the hypertableau technique of Hermit [9]; consequence-driven approaches like the ones described by Kazakov [13]; and resolution based methods described by Motik & Sattler [16]), they do not deal with inconsistency detection in a federated setting.

To the best of our knowledge there is currently no ready to use approach that addresses inconsistency detection and generation of repairs in the context of federated KBs. However, there are some works in a similar direction.

Bonatti et al. [3] proposed an approach that can be applied to a scenario similar to the one we analyzed in our experiments. Their approach is based on annotated logic programs for tracking indicators of provenance and trust during the reasoning process. However, the reasoning itself is not conducted in a given federated setting, while our approach works directly on top of existing SPARQL interfaces. This shall not be confused with the fact that Bonatti et al. described a distributed implementation of the algorithm. Another important difference is the origin of the trust values. While we derive the trust values required for the second phase from explicit and implicit conflicts in the data sources using reasoning in the first phase, Bonatti et al. apply a well-known page rank algorithm that does not at all consider logical dependencies.

Calvanese et al. [4] present apart from the initial definition of the *DL-Lite* family among others a definition of a translation function δ that transforms negative inclusions and functionality assertions into queries (FOL formulas). This translation function is applied in the algorithm Consistent to each negative inclusion and functionality assertion that can be logically implied from the given TBox. Afterwards, a Boolean query comprising the union of all queries generated by δ is evaluated over the given ABox. In contrast to our approach the work of Calvanese et al. do not support *DL-Lite_A* KBs. Besides this,

Calvanese et al. [5] expand their approach to *DLR-Lite_{A, \sqcap}* , a new member of the *DL-Lite* family that permits among others the use of n -ary relations and conjunctions on the left-hand side of inclusion assertions. Despite the fact that the algorithms Consistent proposed in these works are similar to our approach, both only identify if there is an inconsistency but do not specify these inconsistencies in greater detail or give some explanations to them. Furthermore, our approach additionally comprises the federation of distributed *DL-Lite* KBs.

The approach proposed by Lembo et al. [14, 15] facilitate meaningful query results over inconsistent *DL-Lite* KBs under different inconsistency-tolerant semantics. Therefore, an additional rewriting under the defined semantics is applied to the rewritings produced by PerfectRef in order to implement inconsistency tolerance on query answering. Roughly speaking, queries generated by applying backward-chaining are extended such that triples producing inconsistency will not be considered on query answering. For this purpose, similar to our approach ontology axioms that can be contradicted by ABox assertions are used for query generation, i.e., its expansion, but with the difference that their aim is to exclude all assertions that cause inconsistency from query evaluation whereas our claim is to select these assertions, which is exactly the opposite. Although the method of Lembo et al. is suitable for accessing distributed data, it is not designed for inconsistency detection and explanation.

Several approaches have already been proposed to solve the problem of repair plan generation [8]. Depending on the specifics of the setting one might, e.g., be interested to remove a minimum number of assertions causing inconsistency by computing a smallest minimal hitting set over all explanations. However, to the best of our knowledge, none of these approaches consider federated settings and make precise distinctions between assertions occurring in different data sources.

7 Conclusions

In this paper we have described an approach for detecting and resolving inconsistency in federated large scale KBs. The approach is based on the generation of clash queries which are known to be complete for inconsistency detection in *DL-Lite_A* KBs. Once all logical conflicts have been collected, a majority voting scheme is applied in the first phase to determine a partial repair. This approach does not aim at generating a global optimal repair, but applies an efficient heuristic where each step in the algorithm corresponds to a reasonable decision. Based on determining the degree of trust for each assertion type with respect to each data source by analyzing the partial repair, we are able to extend the repair in the second phase.

We applied the approach in a federated setting using four LOD data sources from the domain of library science. Overall, the federated KB consists of more than 284 million triples and we are able to detect 18.1 million conflicts. The results of our experiments show that we are able to solve 92.25% of those conflicts by the proposed approach. Furthermore, according to our evaluation, we can conclude that the rate of remaining MISAs can be reduced by taking the federated setting into account. By manually annotating samples from the generated repairs, we measured a precision between 84% and 97%, which is a surprisingly good result for a fully automated approach.

So far we have focused on ABox assertions of a federated KB. In our future work, we will address a combined approach in which we try to strike a balance between repairing a potentially erroneous TBox and clashing ABox assertions. Furthermore, the assessment of trustworthiness in `owl:sameAs` statements is an open issue that will also be addressed in our future work.

REFERENCES

- [1] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyashev, 'The dl-lite family and relations', *Journal of artificial intelligence research*, **36**(1), 1–69, (2009).
- [2] Franz Baader, *The description logic handbook: theory, implementation, and applications*, Cambridge: Cambridge University Press, 2003.
- [3] Piero A Bonatti, Aidan Hogan, Axel Polleres, and Luigi Sauro, 'Robust and scalable linked data reasoning incorporating provenance and trust annotations', *Web Semantics: Science, Services and Agents on the World Wide Web*, **9**(2), 165–201, (2011).
- [4] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati, 'Tractable reasoning and efficient query answering in description logics: The DL-Lite family', *Journal of Automated Reasoning*, **39**(3), 385–429, (2007).
- [5] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati, 'Data complexity of query answering in description logics', *Artificial Intelligence*, **195**, 335–360, (2013).
- [6] Diego Calvanese, Martin Giese, Dag Hovland, and Martin Rezk, 'Ontology-based integration of cross-linked datasets', in *The Semantic Web - ISWC 2015*, 199–216, Springer, (2015).
- [7] Giorgos Flouris, Zhisheng Huang, Jeff Z Pan, Dimitris Plexousakis, and Holger Wache, 'Inconsistencies, negations and changes in ontologies', *Proceedings of the National Conference on Artificial Intelligence*, **21**(2), 1295, (2006).
- [8] Peter Haase and Guilin Qi, 'An analysis of approaches to resolving inconsistencies in DL-based ontologies', in *Proceedings of the International Workshop on Ontology Dynamics (IWOD-07)*, pp. 97–109, (2007).
- [9] Ian Horrocks, Boris Motik, and Zhe Wang, 'The Hermit OWL Reasoner', in *Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012)*, Manchester, UK, (2012).
- [10] Qiu Ji, Peter Haase, Guilin Qi, Pascal Hitzler, and Steffen Stadtmüller, 'Radonrepair and diagnosis in ontology networks', in *The semantic web: research and applications*, 863–867, Springer, (2009).
- [11] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin, 'Finding all justifications of OWL DL entailments', in *The Semantic Web*, pp. 267–280. Springer, (2007).
- [12] Richard M. Karp, 'Reducibility among combinatorial problems', in *Proceedings of a symposium on the Complexity of Computer Computations*, pp. 85–103, (1972).
- [13] Yevgeny Kazakov, 'Consequence-driven reasoning for Horn *SHIQ* ontologies', in *IJCAI*, volume 9, pp. 2040–2045, (2009).
- [14] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo, 'Query rewriting for inconsistent DL-Lite ontologies', in *Web Reasoning and Rule Systems*, 155–169, Springer, (2011).
- [15] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo, 'Inconsistency-Tolerant First-Order Rewritability of DL-Lite with Identification and Denial Assertions', in *Proceedings of the 25th International Workshop on Description Logics*, (2012).
- [16] Boris Motik and Ulrike Sattler, 'A comparison of reasoning techniques for querying large description logic aboxes', in *Logic for programming, artificial intelligence, and reasoning*, pp. 227–241. Springer, (2006).
- [17] Andreas Nolle, Christian Meilicke, Heiner Stuckenschmidt, and German Nemirovski, 'Efficient federated debugging of lightweight ontologies', in *Web Reasoning and Rule Systems*, 206–215, Springer, (2014).
- [18] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati, 'Linking data to ontologies', in *Journal on data semantics X*, 133–173, Springer, (2008).
- [19] Guilin Qi, Qiu Ji, and Peter Haase, 'A conflict-based operator for mapping revision', in *The Semantic Web-ISWC 2009*, 521–536, Springer, (2009).
- [20] Raymond Reiter, 'A theory of diagnosis from first principles', *Artificial Intelligence*, **32**, 57–95, (1987).
- [21] Sebastian Rudolph, 'Foundations of description logics', in *Reasoning Web. Semantic Technologies for the Web of Data*, 76–136, Springer, (2011).
- [22] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz, 'Pellet: A practical owl-dl reasoner', *Web Semantics: science, services and agents on the World Wide Web*, **5**(2), 51–53, (2007).
- [23] Dmitry Tsarkov and Ian Horrocks, 'FaCT++ description logic reasoner: System description', in *Automated reasoning*, 292–297, Springer, (2006).
- [24] Holger Wache, Thomas Voegelé, Ubbo Visser, Heiner Stuckenschmidt, Gerhard Schuster, Holger Neumann, and Sebastian Hübner, 'Ontology-based integration of information - a survey of existing approaches', in *IJCAI-01 workshop: ontologies and information sharing*, volume 2001, pp. 108–117. Citeseer, (2001).

Belief Contraction Within Fragments of Propositional Logic

Nadia Creignou and Raïda Ktari and Odile Papini¹

Abstract. Recently, belief change within the framework of fragments of propositional logic has gained attention. In the context of revision it has been proposed to refine existing operators so that they operate within propositional fragments, and that the result of revision remains in the fragment under consideration. In this paper we generalize this notion of refinement to belief change operators. Whereas the notion of refinement allowed one to define concrete rational operators adapted to propositional fragments in the context of revision and update, it has to be specified for contraction. We propose a specific notion of refinement for contraction operators, called *reasonable refinement*. This allows us to provide refined contraction operators that satisfy the basic postulates for contraction. We study the logical properties of reasonable refinements of two well-known model-based contraction operators. Our approach is not limited to the Horn fragment but applicable to many fragments of propositional logic, like Horn, Krom and affine fragments.

keywords: Belief change, belief contraction, fragments of propositional logic, knowledge representation and reasoning.

1 INTRODUCTION

Belief change in knowledge representation for artificial intelligence studies how a rational agent may modify his beliefs in presence of new information. Belief contraction is a belief change operation that occurs when some beliefs are retracted but no new information is added. Within the symbolic frameworks, where an agent's beliefs are represented by theories, the AGM paradigm [1, 12] became a standard that provides rational postulates any reasonable belief change operator, in particular any contraction operator, should satisfy. When a theory is represented by a propositional formula, Katsuno and Mendelzon [18] reformulated some of the AGM postulates. More recently Caridroit et al. [3] provided a complete reformulation of the AGM postulates for contraction and proposed a representation theorem that characterizes contraction operations in terms of total preorders over interpretations.

Belief contraction has been studied within the framework of propositional logic and several concrete belief contraction operators have been proposed [1, 12, 13, 11, 21, 14]. More recently, belief contraction has been investigated in the Horn fragment and several families of concrete contraction operators have been proposed [9, 24, 2, 26, 8, 27]. Our goal is to provide new contraction operators that operate in various fragments of propositional logic (including, but not restricted to, the Horn fragment).

The motivation of such a study is twofold. First, in many applications, the language is restricted *a priori*. For instance, a rule-based formalization of expert knowledge is much easier to handle for standard users. Second, some fragments of propositional logic allow

for efficient reasoning methods, and then an outcome of contraction within such a fragment can be evaluated efficiently.

We generalize the notion of refinement, initially defined for revision [6], to any belief change operator defined from $\mathcal{L} \times \mathcal{L}$ to \mathcal{L} where \mathcal{L} denotes propositional logic. A refinement adapts a belief change operator defined in a propositional setting such that it can be applicable in a propositional fragment. The basic properties of a refinement are first to guarantee the outcome of the belief change operation to remain within the fragment and second to approximate the behavior of the original belief change operator, in particular to keep the behavior of the original operator unchanged if the result already fits in the fragment. We characterize these refined operators in a constructive way.

We study the notion of refinement for contraction operators. Contrary to the case of revision and update [6, 5], the refined contraction operators do not necessarily satisfy the basic postulates for contraction. In order to overcome this problem, we introduce a specific notion of refinement for contraction operators, called *reasonable refinement*. This specification allows us to provide concrete rational contraction operators obtained from known model-based contraction operators defined from Dalal's and Satoh's revision operators within the Horn, Krom and affine fragments. We study the logical properties of these operators in terms of satisfaction of postulates for contraction.

An important contribution of our study is that it provides new rational belief contraction operators that work within propositional fragments. In the Horn case, they do not coincide with any contraction operator previously proposed in the literature.

2 PRELIMINARIES

2.1 Propositional logic

Let \mathcal{L} be the language of propositional logic built on an infinite countable set of variables (atoms) and equipped with standard connectives $\rightarrow, \oplus, \vee, \wedge, \neg$, and constants \top, \perp . A literal is an atom or its negation. A clause is a disjunction of literals. A clause is called *Horn* if at most one of its literals is positive; *Krom* if it consists of at most two literals. An \oplus -clause is defined like a clause but using exclusive- instead of standard-disjunction. We identify the following subsets of \mathcal{L} : \mathcal{L}_{Horn} as the set of all formulas in \mathcal{L} being conjunctions of Horn clauses; \mathcal{L}_{Krom} as the set of all formulas in \mathcal{L} being conjunctions of Krom clauses; and \mathcal{L}_{Affine} as the set of all formulas in \mathcal{L} being conjunctions of \oplus -clauses. In what follows we sometimes just talk about arbitrary fragments $\mathcal{L}' \subseteq \mathcal{L}$.

Let \mathcal{U} be a finite set of atoms. An interpretation is represented either by a set $m \subseteq \mathcal{U}$ of atoms (corresponding to the variables set to true) or by its corresponding characteristic bit-vector of length $|\mathcal{U}|$. For instance if we consider $\mathcal{U} = \{x_1, \dots, x_6\}$, the interpretation

¹ Aix Marseille Univ, CNRS, Marseille, France, email: {nadia.creignou,raïda.ktari,odile.papini}@univ-amu.fr

$x_1 = x_3 = x_6 = 1$ and $x_2 = x_4 = x_5 = 0$ will be represented either by $\{x_1, x_3, x_6\}$ or by $(1, 0, 1, 0, 0, 1)$.

For any formula ϕ , let $\text{Var}(\phi)$ denote the set of variables occurring in ϕ . As usual, if an interpretation m defined over \mathcal{U} satisfies a formula ϕ such that $\text{Var}(\phi) \subseteq \mathcal{U}$, we call m a model of ϕ . By $\text{Mod}(\phi)$ we denote the set of all models (over \mathcal{U}) of ϕ . Moreover, $\psi \models \phi$ if $\text{Mod}(\psi) \subseteq \text{Mod}(\phi)$ and $\psi \equiv \phi$ (ϕ and ψ are equivalent) if $\text{Mod}(\psi) = \text{Mod}(\phi)$. For fragments $\mathcal{L}' \subseteq \mathcal{L}$, we also use $T_{\mathcal{L}'}(\psi) = \{\phi \in \mathcal{L}' \mid \psi \models \phi\}$.

2.2 Characterizable fragments of propositional logic

Let us define the fragments of propositional logic we are interested in. This requires some formal definition.

Let \mathcal{B} be the set of Boolean functions $\beta: \{0, 1\}^k \rightarrow \{0, 1\}$ with $k \geq 1$, that have the following properties:

- *symmetry*, i.e., for all permutation σ , $\beta(x_1, \dots, x_k) = \beta(x_{\sigma(1)}, \dots, x_{\sigma(k)})$, and
- *0- and 1-reproduction*, i.e. for every $x \in \{0, 1\}$, $\beta(x, \dots, x) = x$.

Examples of such functions are: the binary AND function denoted by \wedge ; the binary OR function denoted by \vee ; the ternary MAJORITY function, $\text{maj}_3(x, y, z) = 1$ if at least two of the variables x, y , and z are set to 1, and 0 otherwise; and the ternary XOR function $\oplus_3(x, y, z) = x \oplus y \oplus z$.

Recall that we consider interpretations also as bit-vectors. We thus extend Boolean functions to interpretations by applying coordinate-wise the original function. So, if $m_1, \dots, m_k \in \{0, 1\}^n$, then $\beta(m_1, \dots, m_k)$ is defined by $(\beta(m_1[1], \dots, m_k[1]), \dots, \beta(m_1[n], \dots, m_k[n]))$, where $m[i]$ is the i -th coordinate of the interpretation m .

The next definition gives a general formal definition of closure.

Definition 1. Given a set $\mathcal{M} \subseteq 2^{\mathcal{U}}$ of interpretations and $\beta \in \mathcal{B}$, we define $Cl_\beta(\mathcal{M})$, the closure of \mathcal{M} under β , as the smallest set of interpretations that contains \mathcal{M} and that is closed under β , i.e., if $m_1, \dots, m_k \in Cl_\beta(\mathcal{M})$, then $\beta(m_1, \dots, m_k) \in Cl_\beta(\mathcal{M})$.

Definition 2. Let $\beta \in \mathcal{B}$. A set $\mathcal{L}' \subseteq \mathcal{L}$ of propositional formulas is a β -fragment if:

1. for all $\psi \in \mathcal{L}'$, $\text{Mod}(\psi) = Cl_\beta(\text{Mod}(\psi))$
2. for all $\mathcal{M} \subseteq 2^{\mathcal{U}}$ with $\mathcal{M} = Cl_\beta(\mathcal{M})$ there exists a $\psi \in \mathcal{L}'$ with $\text{Mod}(\psi) = \mathcal{M}$
3. if $\phi, \psi \in \mathcal{L}'$ then $\phi \wedge \psi \in \mathcal{L}'$.

We call fragments $\mathcal{L}' \subseteq \mathcal{L}$ which are β -fragments for a $\beta \in \mathcal{B}$ also characterizable fragments (of propositional logic).

Well-known fragments of propositional logic are \mathcal{L}_{Horn} which is an \wedge -fragment, \mathcal{L}_{Krom} which is a maj_3 -fragment and \mathcal{L}_{Affine} which is \oplus_3 -fragment [16, 23]. More generally such fragments were systematically investigated in [4].

2.3 Model-based contraction

Belief contraction consists in removing a belief from an agent's belief set (theory). More formally, in model-based approaches a belief set is represented by a formula, and a contraction operator, denoted by $-$, is a function from $\mathcal{L} \times \mathcal{L}$ to \mathcal{L} that maps two formulas ψ (the initial agent's beliefs) and μ (the belief to be removed) to a new formula

$\psi - \mu$ (the contracted agent's beliefs). We recall the KM postulates for belief contraction [18].

- (C1) $\psi \models \psi - \mu$
- (C2) If $\psi \not\models \mu$, then $\psi - \mu \models \psi$
- (C3) If $\psi - \mu \models \mu$, then $\models \mu$
- (C4*) $(\psi - \mu) \wedge \mu \models \psi$
- (C5) If $\psi_1 \equiv \psi_2$ and $\mu_1 \equiv \mu_2$, then $\psi_1 - \mu_1 \equiv \psi_2 - \mu_2$

(C1) ensures that after contraction, no new information was added to the initial agent's beliefs. (C2) expresses that if μ is not deducible from ψ , then no change is made by the contraction of the initial agent's beliefs. (C3) guarantees that the only possibility for the contraction of ψ by μ to fail is that μ is a tautology. (C4*) says that the initial belief set ψ is deducible from the conjunction of the result of the contraction of ψ by μ and from μ . (C5) reflects the principle of independence of syntax.

More recently Caridroit, Konieczny and Marquis [3] reformulated (C4*) and proposed two new postulates (C6) and (C7):

- (C4) If $\psi \models \mu$, then $(\psi - \mu) \wedge \mu \models \psi$
- (C6) $\psi - (\mu_1 \wedge \mu_2) \models (\psi - \mu_1) \vee (\psi - \mu_2)$
- (C7) If $\psi - (\mu_1 \wedge \mu_2) \not\models \mu_1$, then $\psi - \mu_1 \models \psi - (\mu_1 \wedge \mu_2)$

(C6) and (C7) express the minimality of change for the conjunction. (C6) says that the contraction by a conjunction always implies the disjunction of the two contractions by the conjuncts. (C7) says that if μ_1 has not been removed during the contraction by $\mu_1 \wedge \mu_2$, then the contraction by μ_1 must imply the contraction by the conjunction.

Caridroit, Konieczny and Marquis [3] proposed a representation theorem for model-based contraction operators in the same spirit as Katsuno and Mendelzon's representation theorem for revision. This theorem uses the notion of *faithful assignment* [17] which is a function that maps a formula ψ , to a pre-order \leq_ψ on the interpretations as follows:

- If $m_1 \in \text{Mod}(\psi)$ and $m_2 \in \text{Mod}(\psi)$, then $m_1 =_\psi m_2$,
- If $m_1 \in \text{Mod}(\psi)$ and $m_2 \notin \text{Mod}(\psi)$, then $m_1 <_\psi m_2$,
- If $\psi_1 \equiv \psi_2$, then $\leq_{\psi_1} = \leq_{\psi_2}$.

Proposition 1. [3] A contraction operator $-$ satisfies (C1)-(C7) if and only if there exists a faithful assignment that maps each formula ψ to a total preorder \leq_ψ such that $\text{Mod}(\psi - \mu) = \text{Mod}(\psi) \cup \text{min}(\text{Mod}(\neg\mu), \leq_\psi)$.

One can define model-based contraction operators from model-based revision operators using Harper's identity [15] $\psi - \mu \equiv \psi \vee (\psi \circ \neg\mu)$. We thus define two model-based contraction operators from well-known revision operators, namely, Dalal's [7] and Satoh's [22] revision operators.

In model-based revision operators the closeness between models relies on the symmetric difference between models, that is the set of propositional variables on which they differ.

Dalal measures the minimal change by the cardinality of model change. Let ψ and μ be two propositional formulas and m and m' be two interpretations, $m\Delta m'$ denotes the symmetric difference between m and m' and $|\Delta|^{min}(\psi, \mu)$ denotes the minimum number of propositional variables on which the models of ψ and μ differ and is defined as $\min\{|m\Delta m'| : m \in \text{Mod}(\psi), m' \in \text{Mod}(\mu)\}$. The Dalal revision operator [7], denoted by \circ_D , is then defined by: $\text{Mod}(\psi \circ_D \mu) = \{m \in \text{Mod}(\mu) : \exists m' \in \text{Mod}(\psi) \text{ s. t. } |m\Delta m'| = |\Delta|^{min}(\psi, \mu)\}$.

Satoh interprets the minimal change in terms of set inclusion instead of cardinality on model difference. More formally, let

$\Delta^{min}(\psi, \mu) = \min_{\subseteq} \{m \Delta m' : m \in \text{Mod}(\psi), m' \in \text{Mod}(\mu)\}$. The Satoh revision operator [22], denoted by \circ_S , is then defined by: $\text{Mod}(\psi \circ_S \mu) = \{m \in \text{Mod}(\mu) : \exists m' \in \text{Mod}(\psi) \text{ s.t. } m \Delta m' \in \Delta^{min}(\psi, \mu)\}$.

The contraction operator obtained from Dalal's revision operator in using Harper's identity is denoted by $-_D$, and is defined by $\text{Mod}(\psi -_D \mu) = \text{Mod}(\psi) \cup \text{Mod}(\psi \circ_D \neg\mu)$. The contraction operator obtained from Satoh's revision operator is denoted by $-_S$, and is defined by $\text{Mod}(\psi -_S \mu) = \text{Mod}(\psi) \cup \text{Mod}(\psi \circ_S \neg\mu)$.

Contraction operator $-_D$ satisfies (C1) – (C7) [3] while contraction operator $-_S$ satisfies (C1) – (C6), but violates (C7) [19].

3 REFINEMENT OF BELIEF CHANGE OPERATORS

The problem of standard belief change operators when applied in a fragment of propositional logic is illustrated in the following example, in the case of contraction.

Example 1. Let ψ and μ be two Horn formulas such that $\text{Mod}(\psi) = \{\emptyset, \{a\}, \{b\}\}$ and $\text{Mod}(\mu) = \{\emptyset, \{a\}, \{b\}, \{c\}\}$ (such formulas exist since these sets of models are closed under \wedge). Note that $\text{Mod}(\neg\mu) = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$. The result of the contraction of ψ by μ using Satoh's or Dalal's operator can be easily read in the following table, in which the distance between each model of ψ and each model of $\neg\mu$ is indicated.

Mod(ψ)	Mod($\neg\mu$)			
	{a,b}	{a,c}	{b,c}	{a,b,c}
{a}	1	1	3	2
{b}	1	3	1	2
\emptyset	2	2	2	3
	{a,b}	{a,c}	{b,c}	{a,b,c}

Therefore, $\text{Mod}(\psi -_S \mu) = \text{Mod}(\psi -_D \mu) = \{\emptyset, \{a\}, \{b\}, \{a, b\}, \{a, c\}, \{b, c\}\}$. This set is not \wedge -closed ($\{c\}$ is missing), therefore there is no formula in $\mathcal{L}_{\text{Horn}}$ that has this set of models.

In this example, in order to adapt $-_S$ (or likewise $-_D$) so that the outcome of the contraction is in $\mathcal{L}_{\text{Horn}}$ we have several options: one is to build the closure of the set of models, in our case we have to add $\{c\}$; or to remove either $\{a, c\}$ or $\{b, c\}$ or both.

The considerations of the above example, originally studied in the context of revision in [6], can be generalized to the following problem statement: given a belief change operator $\Delta: \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ and a fragment \mathcal{L}' of propositional logic, how can Δ be adapted (or refined) to a new operator \blacktriangle such that for all $\psi, \mu \in \mathcal{L}'$, also $\psi \blacktriangle \mu \in \mathcal{L}'$?

As proposed in [6] few natural desiderata for such refined operators can be stated.

Definition 3. Let \mathcal{L}' be a fragment of propositional logic and $\Delta: \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ a belief change operator. We call an operator $\blacktriangle: \mathcal{L}' \times \mathcal{L}' \rightarrow \mathcal{L}'$ a Δ -refinement for \mathcal{L}' if it satisfies the following properties, for each $\psi, \psi', \mu, \mu' \in \mathcal{L}'$:

- (i) *Consistency:* $\psi \blacktriangle \mu$ is satisfiable if and only if $\psi \Delta \mu$ is satisfiable.
- (ii) *Equivalence:* If $\psi \Delta \mu \equiv \psi' \Delta \mu'$, then $\psi \blacktriangle \mu \equiv \psi' \blacktriangle \mu'$.
- (iii) *Containment:* $T_{\mathcal{L}'}(\psi \Delta \mu) \subseteq T_{\mathcal{L}'}(\psi \blacktriangle \mu)$.
- (iv) *Invariance:* If $\psi \Delta \mu \in \mathcal{L}'$, then $T_{\mathcal{L}'}(\psi \blacktriangle \mu) = T_{\mathcal{L}'}(\psi \Delta \mu)$.

In [6] the authors defined such refined operators in the context of revision through the notion of β -mappings as defined below. This can be generalized to any belief change operator operating from $\mathcal{L} \times \mathcal{L}$ to \mathcal{L} .

Definition 4. Given $\beta \in \mathcal{B}$, we define a β -mapping, f_β , as an application from sets of models into sets of models, $f_\beta: 2^{2^{\mathcal{U}}} \rightarrow 2^{2^{\mathcal{U}}}$, such that for every $\mathcal{M} \subseteq 2^{\mathcal{U}}$:

1. $Cl_\beta(f_\beta(\mathcal{M})) = f_\beta(\mathcal{M})$, i.e., $f_\beta(\mathcal{M})$ is closed under β .
2. $f_\beta(\mathcal{M}) \subseteq Cl_\beta(\mathcal{M})$.
3. If $\mathcal{M} = Cl_\beta(\mathcal{M})$, then $f_\beta(\mathcal{M}) = \mathcal{M}$.
4. If $\mathcal{M} \neq \emptyset$, then $f_\beta(\mathcal{M}) \neq \emptyset$.

Starting from well-known belief change operators we can define new belief change operators adapted to any fragment of propositional logic \mathcal{L}' in using β -mappings.

Definition 5. Let $\Delta: \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ be a belief change operator and $\mathcal{L}' \subseteq \mathcal{L}$ a β -fragment of classical logic with $\beta \in \mathcal{B}$. Given a β -mapping f_β , we denote with $\Delta^{f_\beta}: \mathcal{L}' \times \mathcal{L}' \rightarrow \mathcal{L}'$ the operator for \mathcal{L}' defined as $\text{Mod}(\psi \Delta^{f_\beta} \mu) := f_\beta(\text{Mod}(\psi \Delta \mu))$. The class $[\Delta, \mathcal{L}']$ contains all operators Δ^{f_β} where f_β is a β -mapping.

Interestingly and as in [6], this class actually captures all refinements we had in mind.

Proposition 2. Let $\Delta: \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ be a belief change operator and $\mathcal{L}' \subseteq \mathcal{L}$ a characterizable fragment of classical logic. Then, $[\Delta, \mathcal{L}']$ is the set of all Δ -refinements for \mathcal{L}' .

Proof. A similar result was obtained in [6] for basic (revision) operators, i.e., operators satisfying $\top \Delta \mu \equiv \mu$. This assumption was only used to prove that any Δ -refinement can be defined through a β -mapping. We give here an alternative proof that does not rely on this assumption.

Let \blacktriangle be a Δ -refinement for \mathcal{L}' . We show that $\blacktriangle \in [\Delta, \mathcal{L}']$. Let f be defined as follows for any set \mathcal{M} of interpretations: $f(\emptyset) = \emptyset$ and for $\mathcal{M} \neq \emptyset$, if there exists a pair $(\psi_{\mathcal{M}}, \mu_{\mathcal{M}})$ of formulas from \mathcal{L}' such that $\text{Mod}(\psi_{\mathcal{M}} \Delta \mu_{\mathcal{M}}) = \mathcal{M}$, then we define $f(\mathcal{M}) = \text{Mod}(\psi_{\mathcal{M}} \blacktriangle \mu_{\mathcal{M}})$, otherwise $f(\mathcal{M}) = Cl_\beta(\mathcal{M})$. Thus the refined operator \blacktriangle behaves like the operator Δ^f .

We show that such a mapping f is a β -mapping. Note that since \blacktriangle is a β -refinement, it satisfies the property of equivalence, thus the actual choice of the pair $(\psi_{\mathcal{M}}, \mu_{\mathcal{M}})$ is not relevant, i.e., given \mathcal{M} , and pairs $(\psi_{\mathcal{M}}, \mu_{\mathcal{M}}), (\psi'_{\mathcal{M}}, \mu'_{\mathcal{M}})$ such that $\text{Mod}(\psi_{\mathcal{M}} \Delta \mu_{\mathcal{M}}) = \text{Mod}(\psi'_{\mathcal{M}} \Delta \mu'_{\mathcal{M}}) = \mathcal{M}$, we have that $\psi_{\mathcal{M}} \blacktriangle \mu_{\mathcal{M}}$ is equivalent to $\psi'_{\mathcal{M}} \blacktriangle \mu'_{\mathcal{M}}$. Thus f is well-defined.

We continue to show that the four properties in Definition 4 hold for f . Property 1 is ensured since for every \mathcal{M} , $f(\mathcal{M})$ is closed under β . Indeed, either $f(\mathcal{M}) = \text{Mod}(\psi_{\mathcal{M}} \blacktriangle \mu_{\mathcal{M}})$ and since $\psi_{\mathcal{M}} \blacktriangle \mu_{\mathcal{M}} \in \mathcal{L}'$ its set of models is closed under β , or $f(\mathcal{M}) = Cl_\beta(\mathcal{M})$. Let us show Property 2, i.e., $f(\mathcal{M}) \subseteq Cl_\beta(\mathcal{M})$ for any set of interpretations \mathcal{M} . It is obvious when $\mathcal{M} = \emptyset$ (then $f(\mathcal{M}) = \emptyset$), as well as when $f(\mathcal{M}) = Cl_\beta(\mathcal{M})$. Otherwise $f(\mathcal{M}) = \text{Mod}(\psi_{\mathcal{M}} \blacktriangle \mu_{\mathcal{M}})$ and since \blacktriangle satisfies containment $\text{Mod}(\psi_{\mathcal{M}} \blacktriangle \mu_{\mathcal{M}}) \subseteq Cl_\beta(\text{Mod}(\psi_{\mathcal{M}} \Delta \mu_{\mathcal{M}}))$. Therefore in any case we have $f(\mathcal{M}) \subseteq Cl_\beta(\mathcal{M})$. For showing Property 3 let us consider $\mathcal{M} \neq \emptyset$ such that $\mathcal{M} = Cl_\beta(\mathcal{M})$. If $f(\mathcal{M}) = Cl_\beta(\mathcal{M})$, then $f(\mathcal{M}) = \mathcal{M}$. Otherwise, $f(\mathcal{M}) = \text{Mod}(\psi_{\mathcal{M}} \blacktriangle \mu_{\mathcal{M}})$ where $\psi_{\mathcal{M}}, \mu_{\mathcal{M}} \in \mathcal{L}'$ such that $\text{Mod}(\psi_{\mathcal{M}} \Delta \mu_{\mathcal{M}}) = \mathcal{M}$. Since \blacktriangle satisfies invariance $\text{Mod}(\psi_{\mathcal{M}} \blacktriangle \mu_{\mathcal{M}}) = \mathcal{M}$. Thus, in any case, $f(\mathcal{M}) = \mathcal{M}$. Property 4 is ensured by consistency of \blacktriangle . \square

Hence, β -mappings will allow us to define a variety of refined operators. We will consider two β -mappings in particular, namely the closure Cl_β defined above and Min_β defined below.

Definition 6. Let $\beta \in \mathcal{B}$ and suppose that \leq is a fixed linear order on the set $2^{\mathcal{U}}$ of interpretations. We define the function Min_β as $Min_\beta(\mathcal{M}) = \mathcal{M}$ if $Cl_\beta(\mathcal{M}) = \mathcal{M}$, and $Min_\beta(\mathcal{M}) = Min_{\leq}(\mathcal{M})$ otherwise.

For \mathcal{L}' a β -fragment and Δ an operator, the corresponding operators Δ^{Cl_β} and Δ^{Min_β} are thus respectively given as $Mod(\psi \Delta^{Cl_\beta} \mu) = Cl_\beta(Mod(\psi \Delta \mu))$ and $Mod(\psi \Delta^{Min_\beta} \mu) = Min_\beta(Mod(\psi \Delta \mu))$.

Example 2. Recall Example 1 where we had $\psi, \mu \in \mathcal{L}_{Horn}$ with $Mod(\psi - \mu) = \{\emptyset, \{a\}, \{b\}, \{a, b\}, \{a, c\}, \{b, c\}\}$ ($- \in \{-D, -S\}$). Our refined operator $-^{Cl_\wedge}$ provides

$$Mod(\psi -^{Cl_\wedge} \mu) = \{\emptyset, \{a\}, \{b\}, \{a, b\}, \{a, c\}, \{b, c\}, \{c\}\}.$$

Assume that we have the following order, \leq on the set of interpretations $\emptyset < \{a\} < \{b\} < \{c\} < \{a, b\} < \{a, c\} < \{b, c\} < \{a, b, c\}$. Then the refined operator $-^{Min_\wedge}$ provides

$$Mod(\psi -^{Min_\wedge} \mu) = \{\emptyset\}.$$

A natural objective is now to study how refined belief change operators behave with respect to satisfaction of postulates that characterize rational operators. This has been already done for revision operators [6], as well as for update operators [5]. We aim at doing it for contraction.

Remind that Harper's identity allows one to define model-based contraction operators from model-based revision operators, $\psi - \mu \equiv \psi \vee (\psi \circ \neg\mu)$. Nevertheless this identity does not allow one to obtain a contraction operator that is adapted to a fragment from a revision operator that is so. Indeed, this identity makes first the revision operator act on the negation of a formula, and second consider the disjunction of two formulas. However characterizable fragments are neither closed under negation nor under disjunction (i.e., given two formulas μ_1 and μ_2 in a β -fragment \mathcal{L}' , neither $\neg\mu_1$, nor $\mu_1 \vee \mu_2$ is necessarily equivalent to a formula in \mathcal{L}').

So, in order to obtain contraction operators that are adapted to fragments it makes sense to study refinements of usual contraction operators. This is what we do in the next section.

4 REFINEMENT OF CONTRACTION OPERATORS

The characterization of refined operators gives a way to define concrete refined operators for which we can study the satisfaction of rationality postulates. The property of containment for a refinement (property (iii) in Definition 3) guarantees that the refined operator approximates the original operator, in the sense that the refinement preserves the logical consequences of the original operator within the considered fragment. In the context of revision this property ensures in particular that if μ is a logical consequence of the revision $\psi \circ \mu$, then μ is also a logical consequence of the refined revision $\psi \bullet \mu$. Hence, this property contributes to the preservation of basic postulates when refining revision operators. In contrast, it turns out to be insufficient in the case of contraction.

We say that a contraction operator satisfies a KM postulate (C_i) ($i = 1, \dots, 7$) in \mathcal{L}' if the respective postulate holds when restricted to formulas in \mathcal{L}' .

4.1 Reasonable refinements

In this section we first show a positive result concerning the preservation of two basic KM postulates by refinement of contraction operators.

Proposition 3. Let $-$ be a contraction operator satisfying KM postulate (C2) (resp., (C5)) and $\mathcal{L}' \subseteq \mathcal{L}$ be a characterizable fragment. Then each refinement of this operator $* \in [-, \mathcal{L}']$ satisfies (C2) (resp., (C5)) in \mathcal{L}' as well.

Proof. Since \mathcal{L}' a characterizable fragment, \mathcal{L}' is a β -fragment for some $\beta \in \mathcal{B}$. According to Proposition 2 we can assume that $* \in [-, \mathcal{L}']$ is an operator of the form $-^{f_\beta}$, where f_β is a suitable β -mapping. We show that $-^{f_\beta}$ satisfies (C2) and (C5) for all ψ and $\mu \in \mathcal{L}'$.

(C2) states that if $\psi \not\equiv \mu$, then $Mod(\psi - \mu) \subseteq Mod(\psi)$. Assume that $\psi \not\equiv \mu$. Since $-$ satisfies (C2), then $Mod(\psi - \mu) \subseteq Mod(\psi)$. Thus $Cl_\beta(Mod(\psi - \mu)) \subseteq Cl_\beta(Mod(\psi))$ by monotonicity of the closure. Hence, $Cl_\beta(Mod(\psi - \mu)) \subseteq Mod(\psi)$ since $\psi \in \mathcal{L}'$ and \mathcal{L}' is a β -fragment. According to property 2 in Definition 4 $f_\beta(Mod(\psi - \mu)) \subseteq Cl_\beta(Mod(\psi - \mu))$, hence $f_\beta(Mod(\psi - \mu)) \subseteq Mod(\psi)$. By definition of $*$, this means that $\psi * \mu \models \psi$.

(C5) : Let ψ_1, ψ_2, μ_1 and μ_2 in \mathcal{L}' such that $\psi_1 \equiv \psi_2$ and $\mu_1 \equiv \mu_2$. Since $-$ satisfies (C5), $\psi_1 - \mu_1 \equiv \psi_2 - \mu_2$. Since $*$ is a $--$ refinement, $\psi_1 * \mu_1 \equiv \psi_2 * \mu_2$ by the property of equivalence (Definition 3). \square

In contrast postulates (C1) and (C3) are not preserved by all refinements as illustrated by the following proposition.

Proposition 4. Let $- \in \{-D, -S\}$ and $\mathcal{L}' \in \{\mathcal{L}_{Horn}, \mathcal{L}_{Krom}\}$. Then the refined operator $-^{Min_\beta}$ violates postulates (C1) and (C3) in \mathcal{L}' .

Proof. Example 2 gives two formulas ψ and μ in \mathcal{L}_{Horn} such that on the one hand $Mod(\psi) \not\subseteq Mod(\psi -^{Min_\wedge} \mu)$, and on the other hand $Mod(\psi -^{Min_\wedge} \mu) \subseteq Mod(\mu)$ but μ is not a tautology. Therefore it proves the proposition in \mathcal{L}_{Horn} . Actually, it also proves it in the case of \mathcal{L}_{Krom} since it is easily seen that the given sets of models are also closed under maj_3 , and therefore there exist formulas in \mathcal{L}_{Krom} having these sets of models as well. \square

In conclusion, in the context of contraction, while the notion of refinement continues to express a kind of approximation of the original operator, it fails at preserving all basic postulates, in particular (C1) and (C3). Thus, refined contraction operators will not necessarily behave rationally. To overcome this difficulty we have to restrict refinements to *reasonable* ones, which are refinements having two additional properties.

Definition 7. Let \mathcal{L}' be a fragment of propositional logic and $- : \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ a contraction operator. We call an operator $* : \mathcal{L}' \times \mathcal{L}' \rightarrow \mathcal{L}'$ a $--$ reasonable refinement for \mathcal{L}' if it is a $--$ refinement that satisfies in addition the two following properties. For all ψ, ψ', μ and $\mu' \in \mathcal{L}'$,

- (v) : If $T_{\mathcal{L}}(\psi - \mu) \subseteq T_{\mathcal{L}}(\psi)$, then $T_{\mathcal{L}'}(\psi * \mu) \subseteq T_{\mathcal{L}'}(\psi)$.
- (vi) : If $T_{\mathcal{L}}(\mu) \not\subseteq T_{\mathcal{L}}(\psi - \mu)$, then $T_{\mathcal{L}'}(\mu) \not\subseteq T_{\mathcal{L}'}(\psi * \mu)$.

Property (v) states that if no new information is added to the initial agent's beliefs by the original operator, then none is either by the refined operator. Property (vi) means that if μ is not deducible from the result of the contraction $\psi - \mu$ by the original operator, then

it is not either from the result of the contraction $\psi * \mu$ by the refined operator.

The refinement by the closure is such a reasonable refinement.

Proposition 5. *For any contraction operator $- : \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ and any β -fragment $\mathcal{L}' \subseteq \mathcal{L}$ of classical logic, $-^{Cl_\beta}$ is a reasonable $--$ -refinement for \mathcal{L}' .*

Proof. The operator $-^{Cl_\beta}$ is a $--$ -refinement for \mathcal{L}' , it remains to show that it is a reasonable one, i.e., that it verifies properties (v) and (vi) in Definition 7.

(v) : Suppose that $T_{\mathcal{L}}(\psi - \mu) \subseteq T_{\mathcal{L}}(\psi)$, that is $\text{Mod}(\psi) \subseteq \text{Mod}(\psi - \mu)$. By monotonicity, $Cl_\beta(\text{Mod}(\psi)) \subseteq Cl_\beta(\text{Mod}(\psi - \mu))$. Since $\psi \in \mathcal{L}'$, we thus get $\text{Mod}(\psi) \subseteq \text{Mod}(\psi -^{Cl_\beta} \mu)$, hence $T_{\mathcal{L}'}(\psi -^{Cl_\beta} \mu) \subseteq T_{\mathcal{L}'}(\psi)$.

(vi) : Suppose that $T_{\mathcal{L}}(\mu) \not\subseteq T_{\mathcal{L}}(\psi - \mu)$. Then, $\text{Mod}(\psi - \mu) \not\subseteq \text{Mod}(\mu)$, and a fortiori $Cl_\beta(\text{Mod}(\psi - \mu)) \not\subseteq \text{Mod}(\mu)$, i.e. $\text{Mod}(\psi -^{Cl_\beta} \mu) \not\subseteq \text{Mod}(\mu)$. Since μ is in \mathcal{L}' , it follows that $T_{\mathcal{L}'}(\mu) \not\subseteq T_{\mathcal{L}'}(\psi -^{Cl_\beta} \mu)$. \square

We now show how to characterize all reasonable refinements.

4.2 Characterization of reasonable refinements

The characterization of all reasonable refinements of a contraction operator within a fragment uses the notion of β -contract-mapping defined as follows.

Definition 8. *Given $\beta \in \mathcal{B}$, we define a β -contract-mapping, f_β , as an application $f_\beta : 2^{2^{\mathcal{U}}} \times 2^{2^{\mathcal{U}}} \times 2^{2^{\mathcal{U}}} \rightarrow 2^{2^{\mathcal{U}}}$, such that for all sets of models $\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2$ in $2^{2^{\mathcal{U}}}$:*

1. $Cl_\beta(f_\beta(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2)) = f_\beta(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2)$,
2. $f_\beta(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) \subseteq Cl_\beta(\mathcal{M})$,
3. If $\mathcal{M} = Cl_\beta(\mathcal{M})$, then $f_\beta(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) = \mathcal{M}$,
4. If $\mathcal{M} \neq \emptyset$, then $f_\beta(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) \neq \emptyset$,
5. If $\mathcal{M}_1 \subseteq \mathcal{M}$, then $\mathcal{M}_1 \subseteq f_\beta(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2)$,
6. If $\mathcal{M} \not\subseteq \mathcal{M}_2$, then $f_\beta(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) \not\subseteq \mathcal{M}_2$.

Observe that by abuse of notation the application Cl_β can be defined by $Cl_\beta(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) = Cl_\beta(\mathcal{M})$. It is then easy to verify that this application satisfies all properties of Definition 8 and thus is a β -contract-mapping. The concept of contract-mapping allows us to define a family of reasonable refined operators for fragments of propositional logic as follows.

Definition 9. *Let $- : \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ be a contraction operator and $\mathcal{L}' \subseteq \mathcal{L}$ a β -fragment of classical logic with $\beta \in \mathcal{B}$. For a β -contract-mapping, f_β , we denote with $-^{f_\beta} : \mathcal{L}' \times \mathcal{L}' \rightarrow \mathcal{L}'$ the operator for \mathcal{L}' defined as*

$$\text{Mod}(\psi -^{f_\beta} \mu) := f_\beta(\text{Mod}(\psi - \mu), \text{Mod}(\psi), \text{Mod}(\mu)).$$

The class $\langle -, \mathcal{L}' \rangle$ contains all operators $-^{f_\beta}$ where f_β is a β -contract-mapping.

The next proposition reflects that the above class captures all reasonable refined contraction operators we had in mind.

Proposition 6. *Let $- : \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ be a contraction operator and $\mathcal{L}' \subseteq \mathcal{L}$ a characterizable fragment of propositional logic. Then, $\langle -, \mathcal{L}' \rangle$ is the set of all reasonable $--$ -refinements for \mathcal{L}' .*

Proof. Let us first show that any operator $-^{f_\beta} \in \langle -, \mathcal{L}' \rangle$ is a reasonable $--$ -refinement for \mathcal{L}' . Observe that while a β -contract-mapping is a ternary application, the first four properties defining it depend only on the first variable and coincide with the properties of a β -mapping. Therefore, according to Proposition 2 the operator $-^{f_\beta}$ is $--$ -refinement for \mathcal{L}' . We only have to prove that it satisfies the two additional properties in Definition 7.

(v) Suppose that $T_{\mathcal{L}}(\psi - \mu) \subseteq T_{\mathcal{L}}(\psi)$. Then, $\text{Mod}(\psi) \subseteq \text{Mod}(\psi - \mu)$ and according to property 5 in Definition 8, $\text{Mod}(\psi) \subseteq f_\beta(\text{Mod}(\psi - \mu), \text{Mod}(\psi), \text{Mod}(\mu))$, i.e., $\text{Mod}(\psi) \subseteq \text{Mod}(\psi -^{f_\beta} \mu)$. Hence, $T_{\mathcal{L}}(\mu) \subseteq T_{\mathcal{L}}(\psi -^{f_\beta} \mu)$, and a fortiori $T_{\mathcal{L}'}(\mu) \subseteq T_{\mathcal{L}'}(\psi -^{f_\beta} \mu)$.

(vi) Suppose that $T_{\mathcal{L}}(\mu) \not\subseteq T_{\mathcal{L}}(\psi - \mu)$. Then, $\text{Mod}(\psi - \mu) \not\subseteq \text{Mod}(\mu)$ and according to property 6 in Definition 8, $f_\beta(\text{Mod}(\psi - \mu), \text{Mod}(\psi), \text{Mod}(\mu)) \not\subseteq \text{Mod}(\mu)$, i.e., $\text{Mod}(\psi -^{f_\beta} \mu) \not\subseteq \text{Mod}(\mu)$. Hence, $\text{Mod}(\psi -^{f_\beta} \mu) \not\subseteq Cl_\beta(\text{Mod}(\mu))$ and since $\mu \in \mathcal{L}'$ $T_{\mathcal{L}'}(\mu) \not\subseteq T_{\mathcal{L}'}(\psi -^{f_\beta} \mu)$.

Conversely, given $*$ a reasonable $--$ -refinement for \mathcal{L}' . Let us prove that $*$ $\in \langle -, \mathcal{L}' \rangle$. Consider the application f defined for all triple of sets of interpretations $(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2)$ as follows. If $\mathcal{M} = \emptyset$, then $f(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) = \emptyset$. If $\mathcal{M} \neq \emptyset$ and if there exists a pair of formulas $(\psi_{\mathcal{M}}, \mu_{\mathcal{M}})$ in \mathcal{L}' , such that $\text{Mod}(\psi_{\mathcal{M}} - \mu_{\mathcal{M}}) = \mathcal{M}$ and $\text{Mod}(\psi_{\mathcal{M}}) = \mathcal{M}_1$ and $\text{Mod}(\mu_{\mathcal{M}}) = \mathcal{M}_2$, then we defined $f(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) = \text{Mod}(\psi_{\mathcal{M}} * \mu_{\mathcal{M}})$. Otherwise $f(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) = Cl_\beta(\mathcal{M})$.

First observe that this application is well defined. Indeed, since the operator $*$ is a reasonable $--$ -refinement for \mathcal{L}' , it does not depend on the choice of the pair $(\psi_{\mathcal{M}}, \mu_{\mathcal{M}})$. Moreover, this application satisfies the first four properties in Definition 8. We have to verify the last two ones.

(5) Suppose that $\mathcal{M}_1 \subseteq \mathcal{M}$ (the case where $\mathcal{M} = \emptyset$ is trivial). If $f(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) = Cl_\beta(\mathcal{M})$, then $\mathcal{M}_1 \subseteq f(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2)$. Now, let us turn to the case where $f(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) = \text{Mod}(\psi_{\mathcal{M}} * \mu_{\mathcal{M}})$, with $\mathcal{M} = \text{Mod}(\psi - \mu)$, $\mathcal{M}_1 = \text{Mod}(\psi)$ and $\mathcal{M}_2 = \text{Mod}(\mu)$. Since $\mathcal{M}_1 \subseteq \mathcal{M}$, $T_{\mathcal{L}}(\mathcal{M}) \subseteq T_{\mathcal{L}}(\mathcal{M}_1)$, that is $T_{\mathcal{L}}(\psi_{\mathcal{M}} - \mu_{\mathcal{M}}) \subseteq T_{\mathcal{L}}(\psi_{\mathcal{M}})$. Since $*$ satisfies property (v) in Definition 7, we get $T_{\mathcal{L}'}(\psi_{\mathcal{M}} * \mu_{\mathcal{M}}) \subseteq T_{\mathcal{L}'}(\psi_{\mathcal{M}})$. Therefore $\text{Mod}(\psi_{\mathcal{M}}) \subseteq \text{Mod}(\psi_{\mathcal{M}} * \mu_{\mathcal{M}})$ since $\psi_{\mathcal{M}} * \mu_{\mathcal{M}} \in \mathcal{L}'$. This proves that $\mathcal{M}_1 \subseteq f(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2)$.

(6) Suppose that $\mathcal{M} \not\subseteq \mathcal{M}_2$. If $f(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) = Cl_\beta(\mathcal{M})$, then $f(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) \not\subseteq \mathcal{M}_2$. If $f(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) = \text{Mod}(\psi_{\mathcal{M}} * \mu_{\mathcal{M}})$, with $\mathcal{M} = \text{Mod}(\psi_{\mathcal{M}} - \mu_{\mathcal{M}})$, $\mathcal{M}_1 = \text{Mod}(\psi_{\mathcal{M}})$ and $\mathcal{M}_2 = \text{Mod}(\mu_{\mathcal{M}})$, then $\text{Mod}(\psi_{\mathcal{M}} - \mu_{\mathcal{M}}) \not\subseteq \text{Mod}(\mu_{\mathcal{M}})$, i.e., $T_{\mathcal{L}}(\mu_{\mathcal{M}}) \not\subseteq T_{\mathcal{L}}(\psi_{\mathcal{M}} - \mu_{\mathcal{M}})$. Therefore, according to property (vi) in Definition 7, we get $T_{\mathcal{L}'}(\mu_{\mathcal{M}}) \not\subseteq T_{\mathcal{L}'}(\psi_{\mathcal{M}} * \mu_{\mathcal{M}})$. Therefore, $\text{Mod}(\psi_{\mathcal{M}} * \mu_{\mathcal{M}}) \not\subseteq \text{Mod}(\mu_{\mathcal{M}})$ since $\mu_{\mathcal{M}}$ is in \mathcal{L}' . This proves that $f(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) \not\subseteq \mathcal{M}_2$. \square

So far we have considered $-^{Cl_\beta}$ as one instantiation of a reasonable $--$ -refinement. In order to get further concrete reasonable refinements we need to define further β -contract-mappings. An additional example is as follows.

Definition 10. *Let $\beta \in \mathcal{B}$ and suppose that \leq is a total order on the set $2^{\mathcal{U}}$ of interpretations. We define the function p_β as*

$$p_\beta(\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2) = \begin{cases} \mathcal{M} & \text{if } \mathcal{M} = Cl_\beta(\mathcal{M}) \\ Cl_\beta(\mathcal{M}_1 \cup Min_{\leq}(\mathcal{M} \cap \overline{\mathcal{M}_2})) & \text{else and if } \mathcal{M}_1 \subseteq \mathcal{M} \\ & \text{and } \mathcal{M} \cap \overline{\mathcal{M}_2} \neq \emptyset \\ Cl_\beta(\mathcal{M}) & \text{otherwise} \end{cases}$$

It is easy to verify that the function p_β satisfies all six properties in Definition 8. As such p_β is a β -contract-mapping. Therefore, according to Proposition 6, for \mathcal{L}' a β -fragment and $-$ a contraction operator, it holds that the operator $-^{p_\beta}$ defined as

$$\text{Mod}(\psi -^{p_\beta} \mu) = p_\beta(\text{Mod}(\psi - \mu), \text{Mod}(\psi), \text{Mod}(\mu))$$

is a reasonable $-$ -refinement for \mathcal{L}' .

Example 3. Recall Example 1 where we had $\psi, \mu \in \mathcal{L}_{Horn}$ with $\text{Mod}(\psi) = \{\emptyset, \{a\}, \{b\}\}$, $\text{Mod}(\mu) = \{\emptyset, \{a\}, \{b\}, \{c\}\}$, and $\text{Mod}(\psi - \mu) = \{\emptyset, \{a\}, \{b\}, \{a, b\}, \{a, c\}, \{b, c\}\}$. Suppose that we have the following order, \leq , on the set of interpretations $\emptyset < \{a\} < \{b\} < \{c\} < \{a, b\} < \{a, c\} < \{b, c\} < \{a, b, c\}$.

Our refined operator $-^{p_\beta}$ provides $\text{Mod}(\psi -^{p_\beta} \mu) = Cl_\beta(\{\emptyset, \{a\}, \{b\}\} \cup Min_{\leq}(\{a, b\}, \{a, c\}, \{b, c\}))$, that is $\text{Mod}(\psi -^{p_\beta} \mu) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$.

4.3 Satisfaction of postulates

In this section we study the properties of our refined contraction operators in terms of satisfaction of KM postulates.

We first show a positive result concerning four basic postulates. We prove that (C1), (C2), (C3) and (C5) are preserved by any reasonable refinement. For the other postulates we obtain more negative results. As a negative result we know that for the Horn fragment, there is no reasonable refinement of any contraction operator that satisfies (C4). We prove that the refinements of Satoh's and Dalal's contraction operators by the two mappings we consider here, Cl_β and p_β , violate (C4) in the Krom fragment as well. We get a similar negative result for the postulate (C6) in both Horn and Krom fragments. For the postulate (C7) the results are more contrasted, the refinement by closure preserves this postulate, while the p_β -refinement does not.

Proposition 7. Let $-$ be a contraction operator and $\mathcal{L}' \subseteq \mathcal{L}$ a characterizable fragment. If $-$ satisfies postulate (C1), (resp. (C2), (C3) and (C5)), then so does any reasonable refinement of this operator $\ast \in \langle -, \mathcal{L}' \rangle$ in \mathcal{L}' .

Proof. Since a reasonable refinement is a refinement, according to Proposition 3 we only have to prove that (C1) and (C3) are preserved. We can assume that $\ast = -^{f_\beta}$ for some suitable β -contract-mapping f_β . Let ψ and μ two formulas in \mathcal{L}' .

(C1): Since $-$ satisfies (C1), $\text{Mod}(\psi) \subseteq \text{Mod}(\psi - \mu)$. According to property 5 in Definition 8, we have $\text{Mod}(\psi) \subseteq f_\beta(\text{Mod}(\psi - \mu), \text{Mod}(\psi), \text{Mod}(\mu))$, i.e., $\psi \models \psi -^{f_\beta} \mu$. So, $\psi \models \psi \ast \mu$.

(C3): Suppose that $\psi \ast \mu \models \mu$, i.e., $\text{Mod}(\psi -^{f_\beta} \mu) \subseteq \text{Mod}(\mu)$. According to property 6 in Definition 8, we get $\text{Mod}(\psi - \mu) \subseteq \text{Mod}(\mu)$. Since $-$ satisfies (C3), $\models \mu$ holds. \square

A natural question is whether one can find reasonable refined operators for characterizable fragments that satisfy all postulates. Actually, this question has already been answered in the Horn fragment.

Indeed, starting from another perspective Flouris et al. studied belief change in a more general setting than classical logic [10]. They gave a necessary and sufficient condition for the existence of a contraction operator satisfying the basic AGM postulates in terms of decomposability. But it was shown in [20] that the Horn fragment is not decomposable. Hence it is not possible to define a Horn contraction that satisfies postulate (C4). In particular the following holds.

Proposition 8. Let $-$ be a contraction operator. Then any reasonably refined operator $\ast \in \langle -, \mathcal{L}_{Horn} \rangle$ violates postulate (C4) in \mathcal{L}_{Horn} .

As far as we know, there is no such a general result for the Krom fragment. We get nevertheless a negative result for the refinement of Satoh's and Dalal's contraction operators by the two mappings we consider here, in the Krom fragment.

Proposition 9. Let $- \in \{-_D, -_S\}$. Then $-^{Cl_{maj_3}}$ and $-^{p_{maj_3}}$ violate postulate (C4) in \mathcal{L}_{Krom} .

Proof. (C4) states that if $\psi \models \mu$, then $(\psi - \mu) \wedge \mu \models \psi$. Let $- \in \{-_D, -_S\}$. By definition there is a $_{maj_3}$ -contract-mapping f_{maj_3} such that $\ast = -^{f_{maj_3}}$. Consider ψ and μ in \mathcal{L}_{Krom} such that $\text{Mod}(\psi) = \{\emptyset, \{a, b\}, \{c, d\}\}$ and $\text{Mod}(\mu) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, c\}, \{a, d\}, \{c, d\}, \{a, c, d\}\}$. Such formulas exist since the corresponding sets of models are $_{maj_3}$ -closed. Observe in addition that $\psi \models \mu$. We have $\text{Mod}(-\mu) = \{\{b, c\}, \{b, d\}, \{a, b, c\}, \{a, b, d\}, \{b, c, d\}, \{a, b, c, d\}\}$. One can easily check that $\text{Mod}(\psi - \mu) = \{\emptyset, \{a, b\}, \{c, d\}, \{a, b, c\}, \{a, b, d\}, \{b, c, d\}\}$. Observe that this set is not closed under $_{maj_3}$. In particular $\{c\} \in Cl_{maj_3}(\text{Mod}(\psi - \mu))$. Therefore, $\{c\}$, which is a model of μ but not a model of ψ , belongs to $\text{Mod}(\psi -^{Cl_{maj_3}} \mu)$, thus proving that $(\psi -^{Cl_{maj_3}} \mu) \wedge \mu \not\models \psi$.

Assume now that we have the following order on interpretations: $\{a, b, c\} < \{a, b, d\} < \{b, c, d\}$. Then $\text{Mod}(\psi -^{p_{maj_3}} \mu) = Cl_{maj_3}(\{\emptyset, \{a, b\}, \{c, d\}\} \cup \{\{a, b, c\}\})$. Therefore, $\{c\} \in \text{Mod}(\psi -^{p_{maj_3}} \mu)$ and we conclude as above. \square

We get also a negative result for postulate (C6).

Proposition 10. Let $- \in \{-_D, -_S\}$. Then $-^{Cl_\beta}$ and $-^{p_\beta}$ violate postulate (C6) in \mathcal{L}_{Horn} , and $-^{Cl_{maj_3}}$ violates postulate (C6) in \mathcal{L}_{Krom} .

Proof. Let $- \in \{-_D, -_S\}$. We first show that $-^{Cl_\beta}$ violates (C6) in \mathcal{L}_{Horn} . Let ψ , μ_1 and μ_2 be Horn formulas such that $\text{Mod}(\psi) = \{\{a, b, c, d\}\}$, $\text{Mod}(\mu_1) = \{\emptyset, \{c\}, \{d\}, \{a, b\}, \{c, d\}, \{a, b, c\}, \{a, b, d\}, \{a, b, c, d\}\}$ and $\text{Mod}(\mu_2) = \{\emptyset, \{b\}, \{d\}, \{a, c\}, \{b, d\}, \{a, b, c\}, \{a, c, d\}, \{a, b, c, d\}\}$. We have then $\text{Mod}(-(\mu_1 \wedge \mu_2)) = \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a, b, d\}, \{b, c, d\}, \{a, c, d\}\}$. On the one hand, $\text{Mod}(\psi - (\mu_1 \wedge \mu_2)) = \{\{a, b, c, d\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}\}$. This set is not closed under \wedge ($\{c, d\}$ is missing). Therefore, $\text{Mod}(\psi -^{Cl_\beta} (\mu_1 \wedge \mu_2)) = \{\{a, b, c, d\}, \{d\}, \{a, d\}, \{b, d\}, \{c, d\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}\}$. On the other hand $\text{Mod}(\psi - \mu_1) = \{\{a, b, c, d\}, \{b, c, d\}, \{a, c, d\}\}$. This set is not closed under \wedge . Therefore $\text{Mod}(\psi -^{Cl_\beta} \mu_1) = \{\{a, b, c, d\}, \{b, c, d\}, \{a, c, d\}, \{c, d\}\}$. Moreover $\text{Mod}(\psi - \mu_2) = \{\{a, b, c, d\}, \{a, b, d\}, \{b, c, d\}\}$, which is not closed under \wedge either ($\{b, d\}$ is missing). Therefore, $\text{Mod}(\psi -^{Cl_\beta} \mu_2) = \{\{a, b, c, d\}, \{a, b, d\}, \{b, c, d\}, \{b, d\}\}$.

Observe that $\text{Mod}(\psi -^{Cl_\beta} \mu_1) \cup \text{Mod}(\psi -^{Cl_\beta} \mu_2) = \{\{a, b, c, d\}, \{b, c, d\}, \{a, c, d\}, \{a, b, d\}, \{c, d\}, \{b, d\}\}$. We conclude that $\text{Mod}(\psi -^{Cl_\beta} (\mu_1 \wedge \mu_2)) \not\subseteq \text{Mod}(\psi -^{Cl_\beta} \mu_1) \cup \text{Mod}(\psi -^{Cl_\beta} \mu_2)$, which proves that $-^{Cl_\beta}$ violates (C6) in \mathcal{L}_{Horn} .

Let us now prove that $-^{p\beta}$ violates (C6) in \mathcal{L}_{Horn} . Consider ψ , μ_1 and μ_2 Horn formulas such that

$$\text{Mod}(\psi) = \{\emptyset, \{a, b, c\}\},$$

$$\text{Mod}(\mu_1) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{b, c\}, \{a, b, c\}\}$$

and

$$\text{Mod}(\mu_2) = \{\emptyset, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}.$$

Thus,

$$\text{Mod}(\neg(\mu_1 \wedge \mu_2)) = \{\{a\}, \{a, b\}, \{a, c\}\}.$$

Assume that we have the following order on the interpretations $\{a, b\} < \{a, c\}$.

On the one hand, $\text{Mod}(\psi - (\mu_1 \wedge \mu_2)) = \{\emptyset, \{a\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}$. This set is closed under \wedge and thus $\text{Mod}(\psi -^{p\wedge} (\mu_1 \wedge \mu_2)) = \{\emptyset, \{a\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}$. On the other hand $\mathcal{M} = \text{Mod}(\psi - \mu_1) = \{\emptyset, \{a, b\}, \{a, c\}, \{a, b, c\}\}$ is not closed under \wedge . Thus,

$$\begin{aligned} \text{Mod}(\psi -^{p\wedge} \mu_1) &= p\wedge(\text{Mod}(\psi - \mu_1)) \\ &= Cl_\beta(\text{Mod}(\psi) \cup \text{Min}_\leq(\mathcal{M} \cap \text{Mod}(\neg\mu_1))) \\ &= Cl_\beta(\{\emptyset, \{a, b, c\}\} \cup \text{Min}_\leq(\{\{a, b\}, \{a, c\}\})) \\ &= \{\emptyset, \{a, b\}, \{a, b, c\}\} \text{ for } \{a, b\} < \{a, c\}. \end{aligned}$$

Moreover $\text{Mod}(\psi - \mu_2) = \{\emptyset, \{a\}, \{a, b\}, \{a, b, c\}\}$, which is closed under \wedge . Thus, $\text{Mod}(\psi -^{p\wedge} \mu_2) = \{\emptyset, \{a\}, \{a, b\}, \{a, b, c\}\}$. Note that $\{a, c\} \in \text{Mod}(\psi -^{p\wedge} (\mu_1 \wedge \mu_2))$ and $\{a, c\} \notin \text{Mod}(\psi -^{p\wedge} \mu_1) \cup \text{Mod}(\psi -^{p\wedge} \mu_2)$, that is to say $\psi -^{p\wedge} (\mu_1 \wedge \mu_2) \not\models \psi -^{p\wedge} \mu_1 \vee \psi -^{p\wedge} \mu_2$. This proves that $-^{p\wedge}$ violates (C6) in \mathcal{L}_{Horn} .

Finally, for \mathcal{L}_{Krom} formulas ψ , μ_1 and μ_2 in \mathcal{L}_{Krom} having as sets of models $\text{Mod}(\psi) = \{\{a, b, c, d\}\}$,

$$\begin{aligned} \text{Mod}(\mu_1) &= \{\{a, c\}, \{b, d\}, \{a, b\}, \{c, d\}, \{a, b, c\}, \{a, c, d\}, \\ &\quad \{a, b, d\}, \{b, c, d\}, \{a, b, c, d\}\} \end{aligned}$$

and

$$\begin{aligned} \text{Mod}(\mu_2) &= \{\{a, b\}, \{c, d\}, \{a, d\}, \{b, c\}, \{a, b, c\}, \{a, c, d\}, \\ &\quad \{a, b, d\}, \{b, c, d\}, \{a, b, c, d\}\}. \end{aligned}$$

can be used to prove that $-^{Cl_{maj_3}}$ violates (C6) in \mathcal{L}_{Krom} . \square

For postulate (C7), we get a positive and a negative result.

Proposition 11. *Let $-$ be a contraction operator and \mathcal{L}' a β -fragment. If $-$ satisfies postulate (C7), then so does the refined operator $-^{Cl_\beta}$ in \mathcal{L}' .*

Proof. (C7) states that if $\psi -^{Cl_\beta} (\mu_1 \wedge \mu_2) \not\models \mu_1$ then $\psi -^{Cl_\beta} \mu_1 \models \psi -^{Cl_\beta} (\mu_1 \wedge \mu_2)$. Assume that $\psi -^{Cl_\beta} (\mu_1 \wedge \mu_2) \not\models \mu_1$, i.e. $Cl_\beta(\text{Mod}(\psi - (\mu_1 \wedge \mu_2))) \not\subseteq \text{Mod}(\mu_1)$. Since $\mu_1 \in \mathcal{L}'$, $Cl_\beta(\text{Mod}(\mu_1)) = \text{Mod}(\mu_1)$. We have $Cl_\beta(\text{Mod}(\psi - (\mu_1 \wedge \mu_2))) \not\subseteq Cl_\beta(\text{Mod}(\mu_1))$. By monotonicity of the closure operator it follows that $\text{Mod}(\psi - (\mu_1 \wedge \mu_2)) \not\subseteq \text{Mod}(\mu_1)$. Since $-$ satisfies (C7), we have $\text{Mod}(\psi - \mu_1) \subseteq \text{Mod}(\psi - (\mu_1 \wedge \mu_2))$. By monotonicity of the closure operator it follows that $Cl_\beta(\text{Mod}(\psi - \mu_1)) \subseteq Cl_\beta(\text{Mod}(\psi - (\mu_1 \wedge \mu_2)))$. Hence, $\text{Mod}(\psi -^{Cl_\beta} \mu_1) \subseteq \text{Mod}(\psi -^{Cl_\beta} (\mu_1 \wedge \mu_2))$, thus proving that $\psi -^{Cl_\beta} \mu_1 \models \psi -^{Cl_\beta} (\mu_1 \wedge \mu_2)$. \square

Proposition 12. *Let $- \in \{-_D, -_S\}$ and $\mathcal{L}' \in \{\mathcal{L}_{Horn}, \mathcal{L}_{Krom}\}$. Then, the refined operators $-^{p\beta}$ violates postulate (C7) in \mathcal{L}' .*

Proof. Let $- \in \{-_D, -_S\}$.

Let us first consider $\mathcal{L}' = \mathcal{L}_{Horn}$. Let ψ , μ_1 and μ_2 be Horn formulas having as sets of models

$$\text{Mod}(\psi) = \{\{a, b\}\},$$

$$\begin{aligned} \text{Mod}(\mu_1) &= \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \\ &\quad \{b, d\}, \{c, d\}, \{a, c, d\}, \{b, c, d\}, \{a, b, c, d\}\} \end{aligned}$$

and

$$\text{Mod}(\mu_2) = \{\emptyset, \{c\}, \{d\}, \{a, b\}, \{c, d\}, \{a, b, c, d\}\}.$$

We have

$$\begin{aligned} \text{Mod}(\neg(\mu_1 \wedge \mu_2)) &= \{\{a\}, \{b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \\ &\quad \{a, b, c\}, \{a, b, d\}, \{b, c, d\}, \{a, c, d\}\}. \end{aligned}$$

Assume that we have the following order on interpretations $\{a, b, c\} < \{a, b, d\} < \{a\} < \{b\}$.

On the one hand we get

$$\text{Mod}(\psi - (\mu_1 \wedge \mu_2)) = \{\{a, b\}, \{a\}, \{b\}, \{a, b, c\}, \{a, b, d\}\}.$$

This set is not closed under \wedge . According to the order on interpretations $\text{Mod}(\psi -^{p\wedge} (\mu_1 \wedge \mu_2)) = \{\{a, b\}, \{a, b, c\}\} \not\subseteq \text{Mod}(\mu_1)$. On the other hand $\text{Mod}(\psi - \mu_1) = \{\{a, b\}, \{a, b, c\}, \{a, b, d\}\}$, which is closed under \wedge . Therefore, $\text{Mod}(\psi -^{p\wedge} \mu_1) = \{\{a, b\}, \{a, b, c\}, \{a, b, d\}\}$. Note that $\text{Mod}(\psi -^{p\wedge} \mu_1) \not\subseteq \text{Mod}(\psi -^{p\wedge} (\mu_1 \wedge \mu_2))$, which means that $\psi -^{p\wedge} \mu_1 \not\models \psi -^{p\wedge} (\mu_1 \wedge \mu_2)$, thus proving that $-^{p\wedge}$ violates le postulat (C7) in \mathcal{L}_{Horn} .

Let us now turn to the Krom fragment. Consider two Krom formulas, ψ and μ_1 , having as sets of models

$$\text{Mod}(\psi) = \{\{a, b, c, d\}\}$$

and

$$\begin{aligned} \text{Mod}(\mu_1) &= \{\{a, c\}, \{b, d\}, \{a, b\}, \{c, d\}, \{a, b, c\}, \{a, c, d\}, \\ &\quad \{a, b, d\}, \{b, c, d\}, \{a, b, c, d\}\}. \end{aligned}$$

Let μ_2 be the formula obtained from μ_1 in exchanging the roles of c and d :

$$\begin{aligned} \text{Mod}(\mu_2) &= \{\{a, b\}, \{c, d\}, \{a, d\}, \{b, c\}, \{a, b, c\}, \{a, c, d\}, \\ &\quad \{a, b, d\}, \{b, c, d\}, \{a, b, c, d\}\}. \end{aligned}$$

Assume that we have the following order on interpretations $\{a, d\} < \{b, c\} < \{a, c\} < \{b, d\}$.

On the one hand,

$$\text{Mod}(\psi - (\mu_1 \wedge \mu_2)) = \{\{a, b, c, d\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}\},$$

which is not closed under maj_3 (e.g. $\{a, c, d\}$ is missing). According to the order on interpretations, $\text{Mod}(\psi -^{p\text{maj}_3} (\mu_1 \wedge \mu_2)) = \{\{a, b, c, d\}, \{a, d\}\} \not\subseteq \text{Mod}(\mu_1)$. On the other hand $\text{Mod}(\psi - \mu_1) = \{\{a, b, c, d\}, \{a, d\}, \{b, c\}\}$, which is closed under maj_3 . Therefore $\text{Mod}(\psi -^{p\text{maj}_3} \mu_1) = \{\{a, b, c, d\}, \{a, d\}, \{b, c\}\}$. Note that $\text{Mod}(\psi -^{p\text{maj}_3} \mu_1) \not\subseteq \text{Mod}(\psi -^{p\text{maj}_3} (\mu_1 \wedge \mu_2))$, which means $\psi -^{p\text{maj}_3} \mu_1 \not\models \psi -^{p\text{maj}_3} (\mu_1 \wedge \mu_2)$, thus proving that $-^{p\text{maj}_3}$ violates (C7) in \mathcal{L}_{Krom} . \square

5 CONCLUDING DISCUSSION

We have investigated to which extent established model-based belief change operators can be refined to work within propositional fragments. We have first defined desired properties any refined belief change operator should satisfy and provided a characterization of all such refined operators. We have then focused on contraction. Our study was carried out in the context of model-based contraction initiated by Katsuno and Mendelzon [18] and enriched by Caridroit et al. [3]. It contributes to popularize this approach, which allows one to study contraction operators in propositional fragments from the models point of view within a suitable formal framework.

Compared to revision and update, refining contraction operators is more involved. In order to obtain rational contraction operators the notion of refinement has to be specified. It requires to take into account not only the result of the initial contraction, but also two additional parameters, the initial belief set and the information to be removed. We have provided concrete refined contraction operators. We have shown that they satisfy the basic postulates, whereas the recovery postulate (C4) and the postulates dealing with the minimality of change (C6) and (C7) are more problematic.

In contrast to previous work on belief contraction that was mainly devoted to the Horn logic, our approach applies to any propositional fragment captured via closure properties on sets of models.

In the Horn case the proposed refined contraction operators provide new operators, that can be compared to two families of model-based contraction operators previously proposed within the Horn fragment, namely *Model-based Horn Contraction* (MHC) [25] and *Maxi Choice Horn Contraction based on Weak Remainder Sets* (MHCWR) [8].

The closure-based refinement coincides with MCH in the special case where the initial contraction operator is defined by $\psi - \mu = \text{Mod}(\psi) \cup \text{Min}(\text{Mod}(\neg\mu), \leq_\psi)$ where \leq_ψ is a faithful preorder over interpretations. This is the case, in particular, for Dalal's and Satoh's contraction operators. Note that, more generally, for any contraction operator satisfying (C1), (C2), (C3), (C5) and (C7), the closure-based refinement provides a contraction operator which operates within the Horn fragment and which satisfies these postulates as well.

The p_β -refinement can behave on some instances as an MHCWR operator (but is not such an operator). Indeed, when the result of the initial contraction is not closed, then $\text{Mod}(\psi -^{p_\beta} \mu) = \text{Cl}_\beta(\text{Mod}(\psi) \cup \{m\})$ where $m \in \text{Mod}(\neg\mu)$. However, while for an MHCWR operator the choice of $m \in \text{Mod}(\neg\mu)$ is arbitrary, in the case of p_β -refinement this model has to be chosen in $\text{Mod}(\psi - \mu) \cap \text{Mod}(\neg\mu)$. As such it corresponds to an instantiation of an MHCWR operator which obeys to the principle of minimal change. Let us examine once more Example 1. No matter what is the fixed order on the interpretations, the model $\{a, b, c\}$ (which is a counter-model of μ and as such a valid candidate for an MHCWR operator) will never be considered as a candidate to be in the result of the contraction by our refined operator. Indeed it is further away from ψ than any other counter-model of μ (e.g. for Dalal's contraction operator, for any model $m \in \text{Mod}(\psi -_D \mu) \cap \text{Mod}(\neg\mu)$, $\min\{|m' \Delta m| : m' \in \text{Mod}(\psi)\} = 1$, while $\min\{|m' \Delta \{a, b, c\}| : m' \in \text{Mod}(\mu)\} = 2$).

Natural extensions of this work are to study contraction when only the formula representing the belief set is in the fragment but not the formula representing the information to be removed, or when only the formula representing the information to be removed but not the formula representing the belief set. Our approach can handle these

extensions.

We plan to continue our study in exploring systematically other belief change operations, in particular belief erasure, which is to contraction as update is to revision.

Besides, more ambitious issues could be investigated, namely the computational complexity of refined contraction operators, and from another point of view, the existence of decomposable characterizable fragments, which would give more general results on the satisfaction or not of postulate (C4).

ACKNOWLEDGEMENTS

This work has been supported by the French National Research Agency, ASPIQ project ANR-12-BS02-0003 and AGGREG project ANR-14-CE25-0017.

REFERENCES

- [1] C.E. Alchourrón, P. Gärdenfors, and D. Makinson, 'On the logic of theory change: Partial meet contraction and revision functions', *Journal of Symbolic Logic*, **50**, 510–530, (1985).
- [2] R. Booth, T.A. Meyer, I.J. Varzinczak, and R. Wassermann, 'On the link between partial meet, kernel, and infra contraction and its application to Horn logic', *Journal of Artificial Intelligence Research (JAIR)*, **42**, 31–53, (2011).
- [3] T. Caridroit, S. Konieczny, and P. Marquis, 'Contraction in propositional logic', in *Proceedings of ECSQARU'15*, pp. 186–196, (2015).
- [4] N. Creignou, P. G. Kolaitis, and B. Zanuttini, 'Structure identification of boolean relations and plain bases for co-clones', *Journal of Computer and System Sciences*, **74**(7), 1103–1115, (2008).
- [5] N. Creignou, R. Ktari, and O. Papini, 'Belief update within propositional fragments', in *Proceedings of ECSQARU'15*, pp. 165–174, (2015).
- [6] N. Creignou, O. Papini, R. Pichler, and S. Woltran, 'Belief revision within fragments of propositional logic', *Journal of Computer and System Sciences*, **80**(2), 427–449, (2014).
- [7] M. Dalal, 'Investigations into theory of knowledge base revision.', in *Proceedings of AAAI'88*, pp. 449–479, (1988).
- [8] J. P. Delgrande and R. Wassermann, 'Horn clause contraction functions', *Journal of Artificial Intelligence Research (JAIR)*, **48**, 475–511, (2013).
- [9] J.P. Delgrande, 'Horn clause belief change: Contraction functions', in *Proceedings of KR'08*, pp. 156–165, (2008).
- [10] G. Flouris, D. Plexousakis, and G. Antoniou, 'Generalizing the AGM postulates: preliminary results and applications', in *Proceedings of NMR'04*, pp. 171–179, (2004).
- [11] A. Fuhrmann, 'Theory contraction through base contraction.', *Studia Logica*, **20**(2), 175–203, (1991).
- [12] P. Gärdenfors, 'Knowledge in flux', in *Cambridge University Press, Cambridge UK*, (1988).
- [13] P. Gärdenfors and D. Makinson, 'Revisions of knowledge systems using epistemic entrenchment', in *Proceedings of TARK'88*, pp. 83–95, (1988).
- [14] S.O. Hansson, 'Kernel contraction.', *Journal of Symbolic Logic*, **59**(3), 845–859, (1994).
- [15] W.L. Harper, 'Rational conceptual change', in *PSA Proceedings of the Biennial Meeting of the philosophy of Science Association*, volume 2: Symposia and invited Papers, pp. 462–494, (1977).
- [16] A. Horn, 'On sentences which are true of direct unions of algebras', *Journal of Symbolic Logic*, **16**, 14–21, (1951).
- [17] H. Katsuno and A.O. Mendelzon, 'Propositional knowledge base revision and minimal change', *Artificial Intelligence*, **52**(3), 263–294, (1991).
- [18] H. Katsuno and A.O. Mendelzon, 'On the difference between updating a knowledge base and revising it', in *Belief revision*, ed., P. Gärdenfors, pp. 183–203. Cambridge University Press, (1992).
- [19] R. Ktari, 'Changement de croyances dans des fragments de la logique propositionnelle', in *Phd thesis. Aix-Marseille University*, (2016).
- [20] M. Langlois, R.H. Sloan, B. Szörényi, and G. Turán, 'Horn complements: Towards Horn-to-Horn belief revision', in *Proceedings of AAAI'08*, pp. 466–471, (2008).

- [21] H. Rott, 'Belief contraction in the context of the general theory of rational choice', *Journal of logic, language and information*, **1**(1), 45–78, (1992).
- [22] K. Satoh, 'Nonmonotonic reasoning by minimal belief revision', in *Proceedings of FGCS'88*, pp. 455–462, Tokyo, (1988).
- [23] T. J. Schaefer, 'The complexity of satisfiability problems', in *Proceedings of STOC'78*, pp. 216–226, (1978).
- [24] M. Wu, D. Zhang, and M. Zhang, 'Language splitting and relevance-based belief change in Horn logic', in *Proceedings of AAAI'11*, pp. 268–273, (2011).
- [25] Z. Q. Zhuang and M. Pagnucco, 'Model based Horn contraction', in *Proceedings of KR'12*, pp. 169–178, (2012).
- [26] Z.Q. Zhuang and M. Pagnucco, 'Transitively relational partial meet Horn contraction', in *Proceedings of IJCAI'11*, pp. 1132–1138, (2011).
- [27] Z.Q. Zhuang and M. Pagnucco, 'Entrenchment-based Horn contraction', *Journal of Artificial Intelligence Research (JAIR)*, **51**, 227–254, (2014).

A Novel Cross-Modal Topic Correlation Model for Cross-Media Retrieval

Yong Cheng, Fei Huang, Cheng Jin, Yuejie Zhang¹ and Tao Zhang²

Abstract. A novel cross-modal topic correlation model CMTCM is developed in this paper to facilitate more effective cross-modal analysis and cross-media retrieval for large-scale multimodal document collections. It can be modeled as a cross-modal topic correlation model which explores the inter-related correlation distribution over the deep representations of multimodal documents. It integrates the deep multimodal document representation, relational topic correlation modeling, and cross-modal topic correlation learning, which aims to characterize the correlations between the heterogeneous topic distributions of inter-related visual images and semantic texts, and measure their association degree more precisely. Very positive results were obtained in our experiments using a large quantity of public data.

1 INTRODUCTION

With the explosive growth of multimodal documents on the Web, how to seamlessly handle the complex structures of multimodal documents to achieve more effective cross-media retrieval has become an important research focus [1]. Usually, a multimodal document is exhibited in a form with different modalities (i.e., both visual and semantic), such as a web image with user defined annotation tags/narrative text descriptions, or a news article with paired visual images and textual illustrations. However, due to the semantic gap, there may be significant differences and independence among visual images and semantic texts for multimodal documents, which leads to the huge difficulty and high uncertainty in making full use of the corresponding relationships between the visual features (in images) and semantic features (in descriptions) [2]. Thus integrating multimodal information sources involved in multimodal documents to enable multimodal topic correlation has been the critical component for supporting cross-media retrieval.

Although multimodal topic correlation has been extensively studied for cross-media retrieval since recent years [3] [4], it still remains the necessity of optimal solutions and three inter-related issues should be addressed simultaneously: 1) valid construction and discovery of valuable document element to characterize visual images and textual descriptions for multimodal document representation; 2) reasonable topic correlation modeling to identify better correlations between visual images and textual descriptions of multimodal documents; and 3) cross-modal topic correlation learning to optimize the objective measurement for inter-related image-description correlations. To address the first issue, it is very important to explore the optimal document element that can achieve

more precise and comprehensive visual and semantic feature expression for multimodal documents. To address the second issue, it is critical to establish a robust probabilistic topic model to maximize the likelihood of the observed multimodal documents in terms of the involved latent topics. To address the third issue, it is significant to map the attributes of different modalities into a common embedding space to efficiently maximize their statistical dependency and correlation.

Based on the above observations, a novel Cross-Modal Topic Correlation Model (CMTCM) is developed in this paper to facilitate more effective cross-media retrieval for large-scale multimodal document collections. Our scheme significantly differs from other earlier work as follows. a) The document element of “*deep word*” is created for encoding both the visual features in visual images (i.e., deep visual word) and the semantic features in textual descriptions (i.e., deep textual word) to obtain better deep multimodal document representation. Compared to the traditional visual word and textual word, the deep visual word that is closer to the visual image semantics can alleviate the problem of semantic gap to a great degree, and the deep textual word that integrates various relationship information among textual words can be more representative for expressing the specific semantics of textual descriptions. b) A relational topic correlation modeling scheme is designed to achieve more precise characterization of the inter-related multimodal correlations between visual images and textual descriptions, in which the topic generation and multimodal correlation learning are fused together to break the limitation of topic consistency in the traditional topic modeling. Different topic sets can be generated for different modality information, and at the same time the heterologous topic information from other modalities can be integrated in the topic generation process for one modality. c) An efficient learning mechanism for cross-modal topic correlation is established to achieve the objective decision-making for multimodal correlation, in which the deep topic features for different modalities are particularly mapped into a common space for mining their inter-related topic correlation. Compared to the traditional topic correlation learning strategies, the cross-modal topic correlation learning considers the heterologous property of topic for different modalities, and utilizes the specific mapping function to learn the topic correlation form different modalities. d) A new cross-modal topic correlation model is built by integrating the above deep multimodal document representation, relational topic correlation modeling, and cross-modal topic correlation learning, which can not

¹ School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China, email: {13110240027; 15210240036; jc; yjzhang}@fudan.edu.cn

² School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, China, email: taozhang@mail.shufe.edu.cn

only enable cross-media retrieval users to present on the multimodal query panel whatever they imagine in their mind, but also obtain the most relevant multimodal documents to the original query intention.

How to integrate multimodal information sources in topic correlation measurement for multimodal documents is an open issue, because it is hard to provide a common base for the correlations among multimodal documents because of the semantic gap. The main contribution of our work is that we effectively apply deep representation, relational modeling and cross-modal learning to enable cross-modal topic correlation model, which has provided a more reasonable base for us to integrate visual correlation with semantic correlation by determining an optimal correlated projection space. Such a cross-modal topic correlation model can be treated as an inter-related correlation distribution over deep representations of multimodal documents, in which the most important is to create more effective multimodal image-description topic correlation and measure what degree they are correlated. Our experiments on a large number of public data have obtained very positive results.

2 RELATED WORK

Topic correlation modeling is not a novel task, but has been the subject of extensive research in areas such as cross-media retrieval for large-scale multimodal documents. Earlier research placed the main emphasis on directly exploiting low-level visual features and simple semantic features to explore the image-description topic correlation [5] [6]. However, because of considering only limited shallow-level visual and textual implication in multimodal documents, such methods often demonstrate a poor performance. Recently, closer attention has been given to the methods that rely on cross-modal correlation mining with both deep-level visual and semantic features, that is, finding the high-level multimodal correlation to associate together visually and semantically correlated images and descriptions [7] [8]. Thus, there has been increasing research interests in leveraging the deep implication from multiple information sources and learning the cross-modal topic correlation for multimodal documents to satisfy more rigid requirements on the precision and efficiency for cross-media retrieval.

In recent years, there is some related research work for modeling the topic correlation between visual contents and semantic descriptions in multimodal documents. Blei et al. (2003) built a set of increasingly sophisticated models for a database of annotated images, culminating in correspondence latent Dirichlet allocation (Corr-LDA), a model that found conditional relationships between latent variable representations of sets of image regions and sets of words [9]. Wang et al. (2009) developed a probabilistic model that simultaneously learned the salient patterns among images that were predictive of their class labels and annotation terms, in which the supervised topic modeling (sLDA) was extended to classification problems and a probabilistic model of image annotation was embedded into the resulting supervised topic model [10]. Putthividhya et al. (2010) presented the topic-regression multimodal Latent Dirichlet Allocation (tr-mmLDA), a novel statistical topic model for the task of image and video annotation, which lay a latent variable regression approach to capture correlations between image or video features and annotation texts [11]. Rasiwasia et al. (2010) studied the problem of joint modeling for the text and image components of multimedia documents, in which the text component was represented as a sample from a hidden topic model learned with latent Dirichlet allocation, and images were represented as bags of

visual (SIFT) features [12]. Nguyen et al. (2013) proposed a novel method for image annotation based on combining feature-word distributions which mapped from the visual space to the word space, and word-topic distributions which formed a structure to capture label relationships for annotation [13]. Niu et al. (2014) addressed the problem of recognizing images with weakly annotated text tags, in which the text tags were first encoded as the relations among the images, and then a semi-supervised relational topic model (ss-RTM) was proposed to explicitly model the image contents and their relations [14]. Wang et al. (2014) proposed a supervised multimodal mutual topic reinforce modeling (M3R) approach, which sought to build a joint cross-modal probabilistic graphical model for discovering mutually consistent semantic topics via the appropriate interactions between model factors (e.g., categories, latent topics and observed multi-modal data) [3]. Zheng et al. (2014) considered the application of DocNADE to deal with multimodal data in computer vision, and proposed a supervised variant of DocNADE (SupDocNADE), which can be used to model the joint distribution over an image's visual words, annotation words and class label [15]. Tian et al. (2015) presented a novel model that utilized the rich surrounding texts of images to perform image annotation, in which the words that described the salient objects in images were extracted by integrating the text analysis, and a new probabilistic topic model was built to jointly model image features, extracted words and surrounding text [16]. Wu et al. (2015) proposed a cross-modal learning to the rank approach called CML²R to discover the latent joint representation of multimodal data, and they assumed that the correlations between the multimodal data were captured in terms of topics, and used a list-wise ranking manner to learn the discriminative ranking function [17]. Chen et al. (2015) addressed the image-text correspondence modeling gap by introducing Visual-Emotional LDA (VELDA), a novel topic model that captured image-text correlations through multiple evidence sources (namely, visual and emotional, yielding the method's namesake) for cross-modality image retrieval [4].

Unfortunately, all these existing approaches have not yet provided good solutions for the following crucial issues, which are tightly coupled with each other. **(1) Discovering Deep Information for Multimodal Document Representation** -- Most existing methods focus on exploiting the regular feature description of visual and semantic exhibition in multimodal documents, and do not consider the deep feature information in different modalities of the same multimodal document. This will result in a serious information loss problem for global visual semantics or inherent semantic associations, and forms the insufficient feature descriptions for multimodal documents. With the deep exploration of visual and semantic appearances for multimodal documents, it appears such a discovery mechanism can mitigate the lack of deep visual and semantic feature information. According to our best knowledge, no existing research has made full use of such both deep visual and semantic information to achieve more accurate multimodal document representation for topic correlation modeling. **(2) Relational Topic Correlation Modeling in Deep Level** -- Most existing work concerns finding the best topic correlation for multimodal documents underlying such an assumption that the latent topic sets for visual image and textual description of each multimodal document should be consistent, that is, the heterologous topic information is not considered for different modality information in the same multimodal document. However, for a multimodal document, the inter-related information in different modalities may not be completely incoordinate. With such an over-

strong assumption, the obvious noises will be introduced into the topic correlation measure between different modality information, and meanwhile the deep inclusion in multimodal content cannot be fully utilized. It's a very significant way to fuse multimodal topic feature information in the deep level, set a novel topic generation pattern, and form an optimal relational topic correlation modeling scheme under more reasonable assumptions. **(3) Cross-Modal Correlation Learning with Deep Topic Features** -- Most existing approaches concentrate on directly matching the topic distributions in different modalities to capture the inter-related correlation between visual image and textual description of the same multimodal document. It's due to a simple intuition that the more similar the topic distributions of different modalities are, the higher correlation they have. However, such a straightforward correlation learning strategy may lead to the imprecise correlation evaluation without the in-depth consideration of the deep topic features and the topic heterogeneity in different modalities. Cross-modal correlation learning could provide helpful hints on mining multimodal information for topic correlation modeling. Establishing multimodal associations between deep visual and semantic topic features may shed light on the in-depth understanding for multimodal documents. Thus, the explicit learning of cross-modal correlations between deep visual and semantic topic features becomes very important. From the viewpoint of multimodal document exploitation, it's a significant way to combine both deep visual and semantic topic abstractions for images and descriptions in a joint space and establish an effective cross-modal joint learning mechanism.

To tackle the above obstacles, we have developed a novel framework by integrating the deep multimodal document representation (i.e., mining the valuable multimodal feature information in the deep level), the relational topic correlation modeling (i.e., bridging the semantic gap between inter-related visual contents and semantic descriptions), and the cross-modal correlation learning (i.e., fusing the optimization mapping strategy to obtain more accurate multimodal topic correlation). In our study, we realize that a multimodal document usually appears with multiple correlated visual and semantic words and spans multimodal associations in both deep visual and semantic word levels. Our cross-modal topic correlation model aims at exploring the deep multimodal correlations involved in images and their descriptions to improve the reasoning ability for topic correlation. It's a new attempt on exploiting such deep feature representation, modeling and learning optimization strategies on cross-modal topic correlation model to facilitate cross-media retrieval.

3 DEEP MULTIMODAL DOCUMENT REPRESENTATION

The multimodal information is the significant expression and exhibition for multimodal document content, that is, the visual image and textual description in each multimodal document. To acquire the cross-modal topic correlation between the visual image and textual description in each multimodal document, the optimal basic element for multimodal document representation should be detected and represented more precisely. Thus the deep multimodal document representation is implemented to exploit multiple document elements in the deep level (i.e., deep visual word and deep textual word) and explore the multimodal associations between deep visual property elements and deep semantic expression elements, as shown in Figure 1.

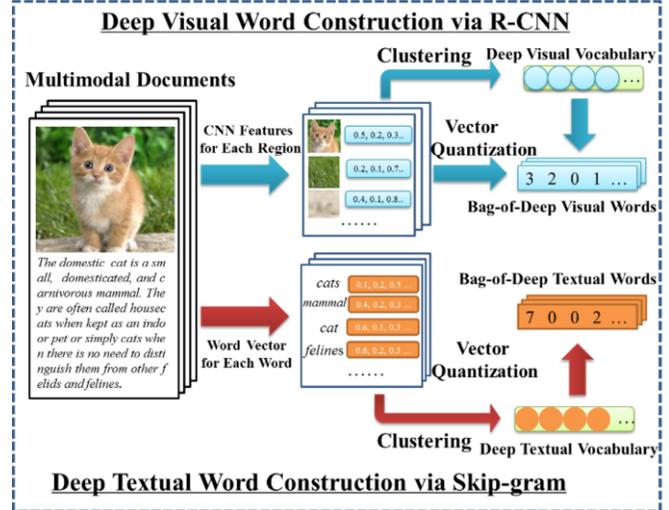


Figure 1. An instantiation for Deep Multimodal Document Representation.

3.1 Deep Visual Word Construction via R-CNN

The Region-based Convolution Neural Network (R-CNN) is a method combining region proposals with CNNs [18], which means extracting all the CNN features for all the regions in the image and is widely used in the field of computer vision [19] [20] [21]. R-CNN first uses selective search methods to generate possible object locations for each image in terms of image regions, and then extracts the feature vector from each region proposal based on CNN. For this purpose, each image region is converted to a fixed pixel size of 227×227 , and all the features are computed through a network with five convolutional layers and two fully connected layers. The advantage of R-CNN is that the visual features extracted via CNN are closer to the image semantics, which can alleviate the problem of semantic gap to a certain degree. Furthermore, the regions contain the important spatial information in the image, and the visual words constructed by R-CNN can better represent the deep image contents. Thus we leverage R-CNN to construct the deep visual words for representing the deep visual semantic properties in images. Firstly, each image is represented in the form of bag-of-regions based on R-CNN, and each region can be viewed as a visual word. Since each region is represented as a feature vector, we use the Vector Quantization (VQ) method [22] to project the higher dimensional features into a sparse presentation. We construct the deep visual word vocabulary by clustering all the region features into a fixed-number of classes, and then project all the regions in the same image into the deep visual word vocabulary. Finally, each image can be represented in the form of bag-of-deep-visual-words. Compared to the traditional visual word descriptors like SIFT, the main advantage of deep visual word is that it can compute visual features with a hierarchical and multi-stage process, which is more informative and effective for visual recognition than using just low-level and superficial visual features.

3.2 Deep Textual Word Construction via Skip-gram

The Skip-gram model is an efficient method to learn the distributed representations of textual words in a vector space from large amounts of unstructured text data [23], which has achieved better performance in a wide range of natural language processing tasks [20] [24]. Its training objective aims at learning deep word vector

representations that are good at predicting the nearby textual words, which can capture more precise syntactic and semantic relationships among textual words and group similar textual words together. Compared to other learning methods for textual word vector, the advantage of Skip-gram is that the training process is extremely efficient for massive text data since it does not involve dense matrix multiplications. Thus we leverage the Skip-gram model to construct the deep textual word for better representing the deep textual semantic properties in textual descriptions.

Let D^T be the textual description part of the whole multimodal document corpus, \mathcal{W} denotes all the raw textual words in D^T , and V is the textual word vocabulary. For each textual word w in \mathcal{W} , I_w and O_w are the input and output vector representations for w , $Context(w)$ represents the nearby textual words of w , here the context window size is set as 5. We define the set of all the input and output vectors for each textual word as a long vector $\omega \in R^{2*|V|*dim}$ and dim is the dimension number of the input or output vector, thus the objective function of Skip-gram can be described as:

$$\begin{aligned} BSG(\omega) &= \operatorname{argmax}_{\omega} \frac{1}{|M|} \sum_{i=1}^{|M|} \sum_{j=1}^{|Context(w_i)|} \log P(w_j | w_i) \\ &= \operatorname{argmax}_{\omega} \frac{1}{|M|} \sum_{i=1}^{|M|} \sum_{j=1}^{|Context(w_i)|} \frac{\exp(O_{w_j} \cdot I_{w_i})}{\sum_{k=1}^{|V|} \exp(O_{w_k} \cdot I_{w_i})} \end{aligned} \quad (1)$$

Since the computing cost is extremely high for the standard softmax formulation of Skip-gram, the Negative Sampling is utilized to compute $\log P(w_j | w_i)$ approximatively.

$$\log P(w_j | w_i) = \log \sigma(O_{w_j} \cdot I_{w_i}) + \sum_{k=1}^m E_{w_k \sim P(w)} \log \sigma(O_{w_j} \cdot I_{w_k}) \quad (2)$$

where $\sigma(\cdot)$ is the sigmoid function; and m is the number of negative samples, each sample is drawn from the noise distribution $P(w)$ based on the textual word frequency. With the learned textual word vector representations, we quantize all these textual word vectors by using the K -means clustering to obtain a discrete set of text terms, which form the new deep textual word vocabulary. Since the textual word vector considers the relationships between textual words, the clustering algorithm allocates the textual words with the high semantic similarity to one new textual word, and all these new textual words constitute the deep textual word vocabulary. Thus each description can be represented in the form of bag-of-deep textual words. Compared to the raw textual word, the main advantage of deep textual word is the consideration of the semantic relationships among raw textual words, which makes the deep textual words more representative to describe textual contents.

4 RELATIONAL TOPIC CORRELATION MODELING VIA CROSS-MODAL LEARNING

The general consideration for multimodal image-description topic correlation is that the topic distribution for the visual appearances of visual image is heterologous but related with the distribution for the semantic exhibitions of textual description. To achieve more precise topic correlation of multimodal documents, it's very useful to establish an effective topic correlation modeling pattern for evaluating the intrinsic image-description association degree among all the multimodal documents in the whole database. Thus a relational topic correlation model is built to fine measure the image-description association, in which the cross-modal learning mechanism is especially conducted over multimodal topic distributions to facilitate more refined evaluation for multimodal topic correlation, as shown in Figure 2.

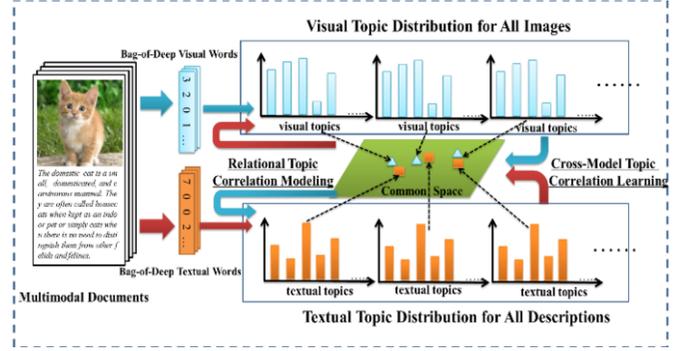


Figure 2. An instantiation for Topic Correlation Modeling.

4.1 Relational Topic Correlation Modeling

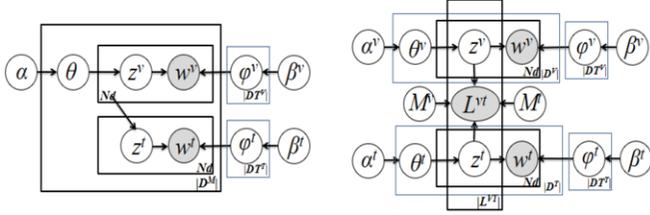
Our main purpose aims at building a joint probabilistic model to maximize the likelihood of the observed multimodal documents. We assume that each deep word is generated from one latent topic, and such a topic is derived from a multinomial distribution over all the topics. The major difference of our modeling is that the latent topic sets for different modalities are different, that is, the number and meaning of the topics from different modalities are different. Such an intuitive perception is because that in many cases the visual image and the textual description in the same multimodal document are semantically correlated but not consistent on latent topics.

We consider splitting the multimodal document collection D^M into three parts, that is, the visual image set D^V , the textual description set D^T , and the linkage set L^{VT} which indicates the multimodal image-description associations. D^V is composed of the deep visual word set DW^V and DV^V is the deep visual vocabulary, while D^T is composed of the deep textual word set DW^T and DV^T is the deep textual vocabulary. For $l^v \in L^{VT}$, $l^v=1$ means that the visual image $d^v \in D^V$ and the textual description $d^t \in D^T$ is relevant, otherwise irrelevant when $l^v=0$. Given DT^V is the visual topic set, DT^T is the textual topic set, α and β are two hyper-parameters for the topic proportion and the topic-deep-word distribution, θ , θ^v and θ^t are the topic distributions for each multimodal document d , its visual image d^v and its textual description d^t , ϕ is the topic-deep-word distribution for each topic, z is the actual deep-word-related topics generated from θ , $Dir(\cdot)$ and $Mult(\cdot)$ denotes the Dirichlet and multinomial distribution, n denotes the n^{th} deep word, and N_d is the total number of deep words in the multimodal document d , the basic framework of our modeling is shown as follows.

1. For each visual topic $t^v \in DT^V$, sample the topic-deep-visual-word distribution over the deep visual vocabulary, i.e., $\phi^{v,t^v} \sim Dir(\phi^v | \beta^v)$.
2. For each textual topic $t^t \in DT^T$, sample the topic-deep-textual-word distribution over the deep textual vocabulary, i.e., $\phi^{t,t^t} \sim Dir(\phi^t | \beta^t)$.
3. For each visual image $d^v \in D^V$:
 - (a) Sample the visual topic distribution $\theta^{d^v} \sim Dir(\theta^v | \alpha^v)$.
 - (b) For each deep visual word w^{v,d^v}, n^v :
 - i. Sample the visual topic assignment $z^{v,d^v}, n^v \sim Mult(\theta^{d^v})$.
 - ii. Sample the deep visual word $w^{v,d^v}, n^v \sim Mult(\phi^{v,z^v,d^v}, n^v)$.
4. For each textual description $d^t \in D^T$:
 - (a) Sample the textual topic distribution $\theta^{d^t} \sim Dir(\theta^t | \alpha^t)$.
 - (b) For each deep textual word w^{t,d^t}, n^t :
 - i. Sample the textual topic assignment $z^{t,d^t}, n^t \sim Mult(\theta^{d^t})$.
 - ii. Sample the deep textual word $w^{t,d^t}, n^t \sim Mult(\phi^{t,z^t,d^t}, n^t)$.
5. For each linkage $l^v \in L^{VT}$ for the relationship between the visual image d^v and the textual description d^t :

(a) Sample the linkage indicator $l^{vt} \sim TCor(l^{vt} | \bar{z}_{d^v}, \bar{z}_{d^t}, M^v, M^t)$, where \bar{z}_{d^v} and \bar{z}_{d^t} are the empirical topic frequencies for d^v and d^t , $\bar{z}_{d^v} = \frac{1}{Nd^v} \sum_{n^v=1}^{Nd^v} z^{v, n^v}$, $\bar{z}_{d^t} = \frac{1}{Nd^t} \sum_{n^t=1}^{Nd^t} z^{t, n^t}$, $M^v \in R^{|\mathcal{T}^v| \times dim}$ and $M^t \in R^{|\mathcal{T}^t| \times dim}$ are two mapping matrices to map the visual and textual topic distributions into one common space with the dimension of dim , and $TCor(l^{vt})$ denotes the topic correlation between d^v and d^t , $TCor(l^{vt}=1)$ is the topic correlation, while $TCor(l^{vt}=0)$ is the pairwise topic uncorrelation.

Compared to the classical Corr-LDA, our modeling does not treat the multimodal document as a single one, but deals with the visual image and the textual description separately, which makes the limitation for the topic sets of different modalities looser and allows different constituents and properties for such topic sets, then the linkage L^{VT} is utilized to link two empirical topic distributions of the visual image and the textual description, as shown in Figure 3:



(a) Corr-LDA-based Modeling (b) Our Modeling Mechanism
Figure 3. Comparison between the Corr-LDA modeling and ours.

Based on the above modeling assumption, a joint probabilistic topic model can be built to maximize the likelihood of the observed multimodal documents, which is defined as:

$$\begin{aligned} & P(D^v, D^t, L^{VT}) \\ &= P(\varphi^v, \varphi^t, \theta^v, \theta^t, Z^v, Z^t, DW^v, DW^t, L^{VT} | \alpha^v, \alpha^t, \beta^v, \beta^t, M^v, M^t) \\ &= \prod_{t^v \in \mathcal{T}^v} Dir(\varphi_{t^v}^v | \beta^v) \prod_{t^t \in \mathcal{T}^t} Dir(\varphi_{t^t}^t | \beta^t) * \\ & \quad \prod_{d^v \in D^v} Dir(\theta^{d^v} | \alpha^v) \prod_{w^v \in \mathcal{W}^v} Mult(z^{v, d^v, n^v} | \theta^{d^v}) Mult(w^{v, d^v, n^v} | z^{v, d^v, n^v}) * \\ & \quad \prod_{d^t \in D^t} Dir(\theta^{d^t} | \alpha^t) \prod_{w^t \in \mathcal{W}^t} Mult(z^{t, d^t, n^t} | \theta^{d^t}) Mult(w^{t, d^t, n^t} | z^{t, d^t, n^t}) * \\ & \quad \prod_{l^{vt} \in L^{VT}} TCor(l^{vt} | \bar{z}_{d^v}, \bar{z}_{d^t}, M^v, M^t) \end{aligned} \quad (3)$$

where the first part means the generation of topic-deep-word distributions, the middle two parts indicate the generation of deep visual and textual words, and the last part represents the generation of image-description linkages.

Our relational topic correlation model considers the heterogeneity of the multimodal topics, and exploits the linkage probability function $TCor(\cdot)$ to associate the topic distributions of different modalities, which can break the constraint of the topic consistency in traditional multimodal topic models.

4.2 Cross-Modal Correlation Learning

Due to the topic heterogeneity for different modalities, directly learning the topic correlation over multimodal topic distributions becomes computationally intractable. Thus we develop a specific cross-modal learning mechanism by projecting multimodal topic distributions into a common space and making sure that the cross-modal correlation can be maximized.

As the important part in computing the multimodal topic correlation probability function $TCor(l^{vt})$, the mapping matrices M^v and M^t aim at mapping the heterogeneous topic distributions into one common space. For the visual topic distribution \bar{z}_{d^v} and the textual topic distribution \bar{z}_{d^t} , f^v and f^t are two new feature vectors in the common space for \bar{z}_{d^v} and \bar{z}_{d^t} . We can compute $TCor(l^{vt})$ with the correlation measurement between f^v and f^t based on two commonly-used vector correlation evaluation patterns, shown as follows:

$$TCor(l^{vt} | \bar{z}_{d^v}, \bar{z}_{d^t}, M^v, M^t) = \begin{cases} \begin{cases} \text{sigmoid}(f^v \cdot f^t), & l^{vt} = 1 \\ 1 - \text{sigmoid}(f^v \cdot f^t), & l^{vt} = 0 \end{cases} \\ \begin{cases} 0.5 + 0.5 * \text{cosine}(f^v, f^t) & l^{vt} = 1 \\ 0.5 - 0.5 * \text{cosine}(f^v, f^t) & l^{vt} = 0 \end{cases} \end{cases} \quad (4)$$

$$f^v = \bar{z}_{d^v} * M^v, f^t = \bar{z}_{d^t} * M^t$$

where *Pattern 1* utilizes the sigmoid function to map the dot product value into $[0, 1]$, and *Pattern 2* computes the topic correlation by normalizing the cosine similarity of two vectors.

Based on the generated multimodal topic distributions, the cross iterative learning is explored to further learn the cross-modal topic correlation more precisely. We consider using Maximum Likelihood Estimate (MLE) to optimize M^v and M^t by maximizing the log probability of Formula (4), and the objective function for cross-modal learning is defined as:

$$\begin{aligned} & F(M^v, M^t) \\ &= \begin{cases} \text{argmax}_{(M^v, M^t)} \sum_{l^{vt}=1} \log \frac{1}{1+e^{-(f^v \cdot f^t)}} + \sum_{l^{vt}=0} \log \frac{e^{-(f^v \cdot f^t)}}{1+e^{-(f^v \cdot f^t)}} \\ \text{argmax}_{M^v, M^t} \sum_{l^{vt}=1} \log \left(0.5 + \frac{f^v \cdot f^t}{2+|f^v|+|f^t|} \right) + \sum_{l^{vt}=0} \log \left(0.5 - \frac{f^v \cdot f^t}{2+|f^v|+|f^t|} \right) \end{cases} \end{aligned} \quad (5)$$

With the above objective function, M^v and M^t can be computed by using the gradient descent strategy. It's worth noting that in the actual training process the training numbers of positive image-description linkages ($l^{vt}=1$) and negative linkages ($l^{vt}=0$) are imbalance, the number of negative ones is far more than that of positive ones. To solve this problem, we randomly sample negative linkages under the constraint that the pairwise image and description are from different classes, and set the proportion for positive and negative linkages as 1:1. Such cross-modal learning can bridge the gap between the heterogeneous topic distributions via mapping the multimodal topic distributions into the learned common space, in which the topic correlation can be integrated to the whole modeling.

4.3. Related Model Inference

Since the exact inference of topic model is generally intractable, some approximate strategies are usually conducted in the model inference. Thus we adopt the collapsed Gibbs sampling to infer the model parameters due to its simplicity and effectiveness [25].

The Gibbs sampling aims at inferring the latent topic for each deep word in each multimodal document. We first compute the marginal probability distribution of the observed deep words, topic assignments and linkages by integrating the other latent variables, shown as follows:

$$\begin{aligned} & P(Z^v, Z^t, DW^v, DW^t, L^{VT}) \\ &= \int \cdot \int (P(\varphi^v, \varphi^t, \theta^v, \theta^t, Z^v, Z^t, DW^v, DW^t, L^{VT})) d\varphi^v d\varphi^t d\theta^v d\theta^t \\ &= \prod_{d^v \in D^v} \frac{\Gamma(|D^v| |\alpha^v|)}{\Gamma(\alpha^v) |D^v|^{|\alpha^v|}} \frac{\prod_{t^v \in \mathcal{T}^v} \Gamma(m_{d^v, t^v}^v + \alpha^v)}{\Gamma(\sum_{t^v \in \mathcal{T}^v} m_{d^v, t^v}^v + |D^v| |\alpha^v|)} \prod_{d^t \in D^t} \frac{\Gamma(|D^t| |\alpha^t|)}{\Gamma(\alpha^t) |D^t|^{|\alpha^t|}} \frac{\prod_{t^t \in \mathcal{T}^t} \Gamma(m_{d^t, t^t}^t + \alpha^t)}{\Gamma(\sum_{t^t \in \mathcal{T}^t} m_{d^t, t^t}^t + |D^t| |\alpha^t|)} * \\ & \quad \prod_{t^v \in \mathcal{T}^v} \frac{\Gamma(|D^v| |\beta^v|)}{\Gamma(\beta^v) |D^v|^{|\beta^v|}} \frac{\prod_{w^v \in \mathcal{W}^v} \Gamma(n_{t^v, w^v}^v + \beta^v)}{\Gamma(\sum_{w^v \in \mathcal{W}^v} n_{t^v, w^v}^v + |V^v| |\beta^v|)} \prod_{t^t \in \mathcal{T}^t} \frac{\Gamma(|D^t| |\beta^t|)}{\Gamma(\beta^t) |D^t|^{|\beta^t|}} \frac{\prod_{w^t \in \mathcal{W}^t} \Gamma(n_{t^t, w^t}^t + \beta^t)}{\Gamma(\sum_{w^t \in \mathcal{W}^t} n_{t^t, w^t}^t + |V^t| |\beta^t|)} * \\ & \quad \prod_{l^{vt} \in L^{VT}} TCor(l^{vt} | \bar{z}_{d^v}, \bar{z}_{d^t}, M^v, M^t) \end{aligned} \quad (6)$$

where $m_{d, t}$ is the number of the topic t that occurs in the related document d , and $n_{t, w}$ is the number of the deep word w assigned to t . Based on this probability distribution, we can further deduce the single-variable probability distribution of the topic assignment z for the Gibbs sampling. The sampling rules for z^v and z^t are defined as:

$$\begin{aligned} & P(z^{v, d^v, n^v} = t^v | Z^{-d^v, n^v}, DW^v, DW^t, L^{VT}) \propto P(Z^t, Z^v, DW^v, DW^t, L^{VT}) \\ & \propto \frac{\hat{m}_{d^v, t^v}^v + \alpha^v}{\sum_{t^v \in \mathcal{T}^v} \hat{m}_{d^v, t^v}^v + \alpha^v} \frac{\hat{n}_{t^v, w^v}^v + \beta^v}{\sum_{w^v \in \mathcal{W}^v} \hat{n}_{t^v, w^v}^v + \beta^v} \prod_{l^{vt} \in L^{VT}} TCor(l^{vt} | \bar{z}_{d^v}, \bar{z}_{d^t}, M^v, M^t) \\ & \quad \frac{d^v \text{ in } l^{vt}}{d^v \text{ in } l^{vt}} \\ & P(z^{t, d^t, n^t} = t^t | Z^{-d^t, n^t}, DW^v, DW^t, L^{VT}) \propto P(Z^t, Z^v, DW^v, DW^t, L^{VT}) \quad (7) \\ & \propto \frac{\hat{m}_{d^t, t^t}^t + \alpha^t}{\sum_{t^t \in \mathcal{T}^t} \hat{m}_{d^t, t^t}^t + \alpha^t} \frac{\hat{n}_{t^t, w^t}^t + \beta^t}{\sum_{w^t \in \mathcal{W}^t} \hat{n}_{t^t, w^t}^t + \beta^t} \prod_{l^{vt} \in L^{VT}} TCor(l^{vt} | \bar{z}_{d^v}, \bar{z}_{d^t}, M^v, M^t) \\ & \quad \frac{d^t \text{ in } l^{vt}}{d^t \text{ in } l^{vt}} \end{aligned}$$

where $\hat{m}_{d,t}$ denotes the occurrence number of the topic t in the document d excluding the current deep word; similarly $\hat{n}_{t,w}$ denotes the occurrence number of the deep word w assigned to the topic t excluding the current deep word. As described in Formula (5), the mapping matrices M and M' can be updated in each sampling iteration by using the gradient descent method to get the optimized values. With the statistics of the topic assignment z acquired in the sampling process, the other latent variables like ϕ^V , ϕ^T , θ^V , θ^T can be computed as:

$$\begin{aligned} \theta^V_{d^v,t^v} &= \frac{m^v_{d^v,t^v} + \alpha^v}{\sum_{t^v \in DT^V} m^v_{d^v,t^v} + |DT^V| \alpha^v}, \theta^T_{d^t,t^t} = \frac{m^t_{d^t,t^t} + \alpha^t}{\sum_{t^t \in DT^T} m^t_{d^t,t^t} + |DT^T| \alpha^t} \\ \phi^V_{t^v,w^v} &= \frac{n^v_{t^v,w^v} + \beta^v}{\sum_{w^v \in DV^V} n^v_{t^v,w^v} + |DV^V| \beta^v}, \phi^T_{t^t,w^t} = \frac{n^t_{t^t,w^t} + \beta^t}{\sum_{w^t \in DV^T} n^t_{t^t,w^t} + |DV^T| \beta^t} \end{aligned} \quad (8)$$

5 EXPERIMENT AND ANALYSIS

5.1 Dataset and Evaluation Metrics

Our dataset is established based on two benchmark datasets of *Nus-Wide (Nus)* [26] and *Wiki_10cats (Wiki)* [12]. The *Nus-Wide* dataset is collected from the *Flickr* website, which contains 269,648 images with 1,000-dimensional tags and 81-dimensional concepts. Each image in *Nus-Wide* is annotated with several user-defined tags. As the work in [3], we only select those image-text pairs that belong to the 10 largest categories. As a result, we get 20,000 image-annotation pairs for training, 1,000 pairs for verification and 4,000 pairs as testing queries. As for *Wiki_10cats*, all the image-text pairs are collected from the *Wikipedia's "featured articles"*, which is a continually updated collection of articles selected by *Wikipedia's* editors. These articles are accompanied by one or more pictures from Wikimedia Commons. In our work, 1,866 *Wikipedia* multimodal documents from the 10 most populated categories are selected for our experiment, with 2,173 pairs for training, 200 pairs for verification and 693 pairs as testing queries. In addition, each image/annotation or description in both datasets is represented as a 500-dimensional bag of deep visual/textual words by a specific grid search method.

To evaluate the effectiveness of our algorithm, the ground truth image-description correlation is considered to measure the official criteria of Precision-Recall (P - R) curves and Mean Average Precision (MAP) for cross-media retrieval. Cross-media retrieval allows different retrieval manners with the original queries in different modalities, that is, image query-to-text retrieval (return all the relevant texts for the given image query) or text query-to-image retrieval (return all the relevant images for the given text query). To measure the average performance of different retrieval manner, $AMAP$ (the average MAP for both retrieval manners) is also used as an evaluation criterion in our experiment. The ranking score for such different retrieval manners can be defined as:

$$\begin{aligned} \text{RankingScore}(\text{image query} - \text{to} - \text{text}) \\ &= \text{RankingScore}(d^v|d^v) = \frac{TCor(I^{v^t}=1|\theta^v_{d^v,t^t}, \theta^t_{d^t,t^t}, M^v, M^t)}{\sum_{d^t \in DT^T} TCor(I^{v^t}=1|\theta^v_{d^v,t^t}, \theta^t_{d^t,t^t}, M^v, M^t)} \\ \text{RankingScore}(\text{text query} - \text{to} - \text{image}) \\ &= \text{RankingScore}(d^v|d^t) = \frac{TCor(I^{v^t}=1|\theta^v_{d^v,t^t}, \theta^t_{d^t,t^t}, M^v, M^t)}{\sum_{d^t \in DV^V} TCor(I^{v^t}=1|\theta^v_{d^v,t^t}, \theta^t_{d^t,t^t}, M^v, M^t)} \end{aligned} \quad (9)$$

where M^v and M^t are obtained through Formula (5) in the training process; and θ_d is the topic distribution for the test document d , which can be calculated by aggregating all the word-topic distribution for each word in d based on the topic-word distribution ϕ obtained through Formula (8) in the training process.

5.2 Experiment on Cross-Modal Topic Correlation Model

Our cross-modal topic correlation model is created by integrating Deep Multimodal Document Representation (DMDR), Relational Topic Correlation Modeling (RTCM), and Cross-Modal Correlation Learning (CMCL). To show the effect of each part, we focus on investigating the whole performance of our cross-modal topic correlation model for cross-media retrieval. We compare the performance rising speeds for different scheme settings of DMDR, RTCM and CMCL, which implies the effectiveness difference between the general topic correlation modeling without DMDR, RTCM or CMCL and our proposed modeling with the integration of these three components. The related experimental results for cross-modal topic correlation model with different scheme settings are shown in Figure 4–6.

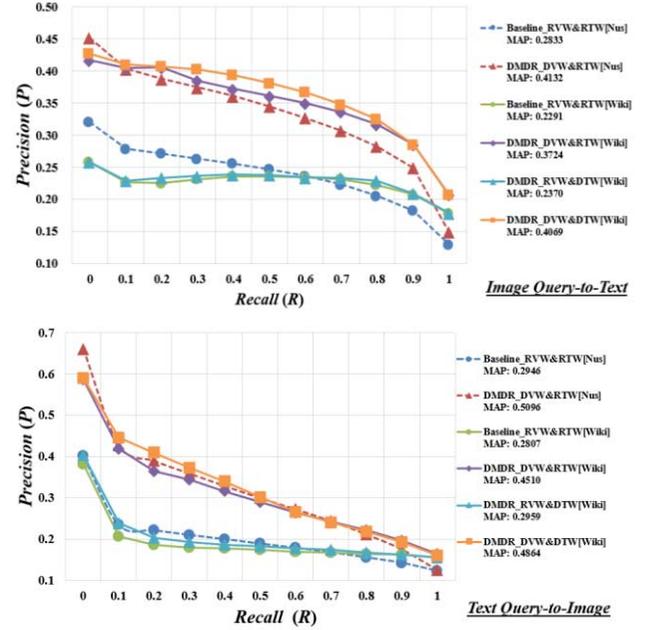


Figure 4. The experimental results on our model with different scheme settings of DMDR, in which we take the representation with raw visual and textual words as the baseline (*Baseline_RVW&RTW*) and make a comparison to DMDR with deep visual and raw textual words (*DMDR_DVW&RTW*), DMDR with raw visual and deep textual words (*DMDR_RVW&DTW*) and DMDR with deep visual and textual words (*DMDR_DVW&DTW*).

It can be seen from Figure 4 that for the cross-modal topic correlation model with DMDR on *Nus-Wide* and *Wiki_10cats*, we can obtain the best cross-media retrieval performance ($MAP=0.5096$) in the evaluation pattern of fusing *DMDR_DVW&DTW* with RTCM and CMCL. In comparison with the baseline pattern using the raw visual and textual word information for multimodal document representation, the performance could be greatly promoted by successively adding the deep visual and textual word representation, which confirms the obvious advantage of our deep multimodal document representation for cross-modal topic correlation model. Through comparing the baseline with two patterns using single deep visual or textual word information, our model can still gain the significant advantage for the performance on both *Nus-Wide* and *Wiki_10cats*. Meanwhile, we can find the performance with single deep visual word information appears better, which shows the beneficial effect of deep visual features on cross-modal topic correlation model. Comparing the results on *Nus-Wide* and

Wiki_10cats, the results on *Nus-Wide* appear less performant on the whole *P-R* curves, while better on the *MAP* values that are measured with the performance statistics for the top-50 ranking results. Overall, the performance difference for *Nus-Wide* and *Wiki_10cats* is not obvious, which reflects the performance advantage to some degree. Due to the differences between these two datasets, we do not compare two patterns of *DMDR_RVW&DTW* and *DMDR_DVW&DTW* on *Nus-Wide*. *Nus-Wide* is a typical social annotated image dataset with discrete tags in each textual annotation, and these discrete annotation tags are independent and meaningful for describing images, so we just use the raw text words in our experiment. While *Wiki_10cats* is a *Wikipedia* featured article dataset with successive narrations in each textual description, there are a lot of redundant information in the raw documents, so the deep textual words are applied in *Wiki_10cats*. As shown in Figure 4, the same conclusions as above can be drawn from the *P-R* curves and *MAP* values for both *Image Query-to-Text* and *Text Query-to-Image* retrieval on *Nus-Wide* and *Wiki_10cats*, which show the consistence of our model on the performance indicators for different cross-media retrieval manners. These results are consistent with what we expect given deep affluent feature descriptions for multimodal documents.

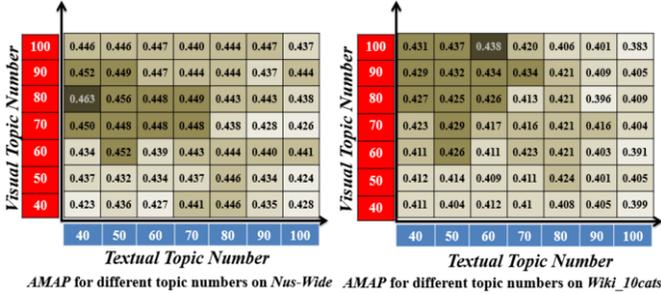


Figure 5. The experimental results on our model with different scheme settings of RTCM, in which we introduce the density graph to show the *AMAP* values for the average performance of cross-media retrieval with different numbers of visual and textual topics.

It can be viewed from Figure 5 that the best performance can be achieved on both *Nus-Wide* and *Wiki_10cats* when the numbers for visual and textual topics are under different settings. This confirms the advantage of our modeling mechanism with the basis assumption that the topics in different modalities are heterologous. Meanwhile, we can find that the best performance can be obtained when the number of visual topics is more than the number of textual topics, which indicates the deep feature information involved in the visual image is richer than that in the textual description on both *Nus-Wide* and *Wiki_10cats*. In addition, we also observe that the best performance on *Nus-Wide* can be acquired when the visual and textual topic numbers are set as 40 and 80 respectively, while on *Wiki_10cats* the best performance can be implemented when the visual and textual topic numbers are set as 60 and 100 respectively. It's obvious that the topic numbers utilized on *Nus-Wide* is smaller than those on *Wiki_10cats*, which is also due to the structural differences between these two datasets as mentioned above, and the contents in *Wiki_10cats* are more complicated and diverse.

It can be found from Figure 6 that for the cross-modal topic correlation model with CMCL on *Nus-Wide* and *Wiki_10cats*, we can obtain the best performance (*MAP*=0.5096) in the evaluation pattern of fusing *CMCL_Sigmoid* with *DMDR* and *RTCM*. In comparison with the baseline *Corr-LDA* model [9], which has the tight restriction that the topic distributions for different modalities

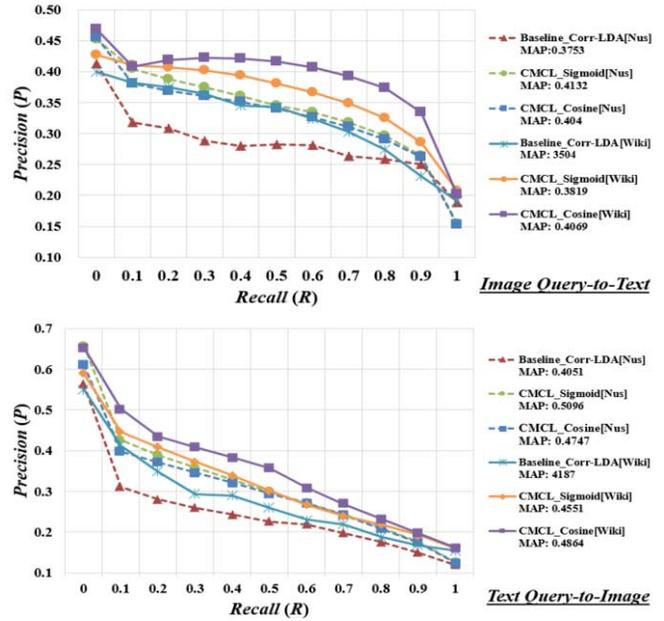


Figure 6. The experimental results on our model with different scheme settings of CMCL, in which we introduce the *Corr-LDA* (without learning but restricting the same topic distributions in different modalities) as the baseline (*Baseline_Corr-LDA*) and make a comparison with the models using two kinds of learning pattern, that is, CMCL with the sigmoid function to compute the topic correlation probability $TCor(CMCL_Sigmoid)$, and CMCL with the cosine function to compute $TCor(CMCL_Cosine)$.

must be same, the whole performance could be greatly promoted by integrating the learning with *CMCL_Sigmoid* or *CMCL_Cosine* in the topic correlation modeling, which confirms the obvious advantage of our cross-modal correlation learning scheme. Comparing two learning patterns of *CMCL_Sigmoid* and *CMCL_Cosine*, the *CMCL_Sigmoid*-based learning achieves the better performance than the *CMCL_Cosine*-based one on *Nus-Wide*, while on *Wiki_10cats* the *CMCL_Cosine*-based learning performs better due to the different dataset structure from *Nus-Wide*. The same conclusions as above can also be drawn from the *P-R* curves and *MAP* values for both *Image Query-to-Text* and *Text Query-to-Image*.

From all the above observations, it's worth noting that our cross-modal topic correlation model is available and presents more impactful ability for discovering the meaningful deep multimodal features and correlations. Our framework can not only significantly improve the cross-modal topic correlation measurement, but also greatly enhance the cross-media retrieval in different manners. The same conclusions from two different datasets of *Nus-Wide* and *Wiki_10cats* show the consistence of our model on different data sources. An instantiation of some cross-media retrieval results with our cross-modal topic correlation model is shown in Figure 7.

5.3 Comparison with Existing Approaches

Compared to the common topic correlation methods in recent years, our approach is a new exploration for taking full advantage of deep information in cross-modal topic correlation measurement. To give full exhibition to the superiority of our topic correlation model, we have also performed a comparison between our method and the other existing classical approaches in recent years. Three approaches developed by Blei et al. (2003) [9], Pereira et al. (2014) [27] and Wang et al. (2014) [3] respectively are analogous with ours to some



Figure 7. An instantiation of some retrieval results with our model.

extent, and then we accomplished them on the same dataset. The experimental results are presented in Table 1, which reflect the difference of power among these four approaches.

Table 1. The comparison results between our and the other approaches.

Dataset	Approach	Evaluation Pattern	MAP		AMAP
			Text Query-to-Image	Image Query-to-Text	
Nus-Wide	Corr-LDA (Blei et al., 2003) (Blei)	Original	0.2513	0.2444	0.2479
		Original+DMDR	0.4051	0.3753	0.3902
	LDA-KCCA (Pereira et al., 2014) (Pereira)	Original	0.3021	0.2726	0.2874
		Original+DMDR	0.4214	0.3829	0.4022
	MFR (Wang et al., 2014) (Wang)	Original	0.2631	0.2714	0.2673
		Original+DMDR	0.4611	0.4092	0.4352
Our Approach	No_DMDR +RTCM+CMCL	0.2946	0.2833	0.2890	
	DMDR+RTCM+CMCL	0.5096	0.4132	0.4614	
Wiki_10cats	Corr-LDA (Blei et al., 2003) (Blei)	Original	0.2261	0.2157	0.2209
		Original+DMDR	0.4187	0.3504	0.3846
	LDA-KCCA (Pereira et al., 2014) (Pereira)	Original	0.2563	0.2268	0.2415
		Original+DMDR	0.4154	0.3587	0.3871
	MFR (Wang et al., 2014) (Wang)	Original	0.2387	0.2135	0.2261
		Original+DMDR	0.4394	0.375	0.4072
Our Approach	No_DMDR +RTCM+CMCL	0.2807	0.2291	0.2549	
	DMDR+RTCM+CMCL	0.4864	0.4069	0.4467	

It can be found from Table 1 that for the topic correlation models on *Nus-Wide* and *Wiki_10cats* by Blei/Pereira/Wang *et al.*'s approaches, we can obtain the best *AMAP* values of 0.3902, 0.4022 and 0.4352 in the evaluation pattern of *Original+DMDR* respectively. The main reason is that when considering the deep multimodal feature attributes all these three approaches can explore more precise multimodal topic distribution information for topic correlation measurement and then the relatively better *AMAP* values can be obtained on both *Nus-Wide* and *Wiki_10cats* in comparison with their original performance exhibitions (i.e., *Blei/Pereira/Wang (Original)*). This obviously confirms the prominent role of our DMDR in the topic correlation modeling. Comparing the results of *Blei/Pereira/Wang (Original)* and our baseline model with *No_DMDR+RTCM+CMCL*, we can find the best *AMAP* value of 0.2890 appears in the results of our model, which is obviously higher than those *AMAP* values of 0.2479, 0.2874 and 0.2673 for *Blei/Pereira/Wang (Original)* respectively. This implies that our cross-modal topic correlation modeling mechanism with RTCM and

CMCL is feasible for facilitating more effective topic correlation evaluation and optimization. Compared to the improved Blei/Pereira/Wang *et al.*'s approaches that integrate with DMDR, our model with DMDR, RTCM and CMCL can still present the better performance on both *Nus-Wide* and *Wiki_10cats*, and the best *AMAP* value of 0.4614 differs greatly from those of Blei/Pereira/Wang *et al.*'s. This indicates that our approach is really superior to Blei/Pereira/Wang *et al.*'s, and also further confirms that our cross-modal topic correlation model with DMDR, RTCM and CMCL is exactly a better way for determining cross-modal image-description topic correlation and can support cross-media retrieval with queries in different modalities more effectively. From the view of computational load performance, the retrieval efficiency of our model is high during the process of cross-media retrieval, and can meet the demand of real-time response.

5.4 Analysis and Discussion

Through the analysis for the topic correlated image-description linkages with failure, it can be found that the modeling quality is highly related to the following aspects. (1) The modeling effect is closely associated with the appropriate representation for multimodal document. It's easier to introduce superficial and noisy information for images and descriptions, which will seriously affect the whole retrieval performance. (2) For multimodal topic model modeling, it's helpful to integrate the topic generation and cross-modal topic correlation analysis into one model, which can adaptively generate the latent topics related to both visual contents and textual information. (3) In some multimodal documents, the textual description has very weak correlation to the visual image content, which leads to the huge semantic topic gap between images and descriptions. It's hard for such documents to successfully implement precise cross-modal topic correlation measurement. This may be the stubbornest problem. (4) The modeling effect is greatly influenced by the topic distribution and number, but such important information may be changed dynamically. It's better to establish the adaptive strategy for finding the optimal settings.

6 CONCLUSIONS AND FUTURE WORK

A new cross-modal topic correlation model is implemented to exploit multimodal correlations between visual images and textual descriptions to enable more effective cross-media retrieval for large-scale multimodal documents. The deep words are conducted to discover deep features for multimodal document representation. A relational topic correlation modeling scheme is designed to achieve more precise characterization of multimodal correlations. An efficient learning mechanism is established to achieve more objective decision-making for cross-modal image-description topic correlation. Our future work will focus on adding supervised information to our model and making our system available online, so that more Internet users can benefit from our research.

7 ACKNOWLEDGMENTS

This work is supported by National Natural Science Fund of China (61572140), Shanghai Municipal R&D Foundation (16511105402&16511104704), Shanghai Philosophy Social Sciences Planning Project (2014BYY009), and Zhuxue Program of Fudan University. Yuejie Zhang is the corresponding author.

REFERENCES

- [1] R.Datta, D.Joshi, J.Li, and J.Z.Wang. 'Image Retrieval: Ideas, Influences, and Trends of the New Age', *ACM Computing Surveys (CSUR)* 40(2), Article 5, (2008).
- [2] J.P.Fan, X.F.He, N.Zhou, J.Y.Peng, and R.Jain. 'Quantitative Characterization of Semantic Gaps for Learning Complexity Estimation and Inference Model Selection', *IEEE Transactions on Multimedia* 14(5):1414-1428, (2012).
- [3] Y.F.Wang, F.Wu, J.Song, X.Li, and Y.T.Zhuang. 'Multi-modal Mutual Topic Reinforce Modeling for Cross-media Retrieval', In *Proceedings of MM 2014*, 307-316, (2014).
- [4] T.Chen, H.M.SalahEldeen, X.N.He, M.Y.Kan, and D.Y.Lu. 'VELDA: Relating an Image Tweet's Text and Images', In *Proceedings of AAAI 2015*, (2015).
- [5] K.Barnard, P.Duygulu, D.Forsyth, N.Freitas, D.M.Blei, and M.I.Jordan. 'Matching Words and Pictures', *Journal of Machine Learning Research*. 3:1107-1135, (2003).
- [6] X.Wang, Y.Liu, D.Wang, and F.Wu. 'Cross-media Topic Mining on Wikipedia', In *Proceedings of MM 2013*, 689-692, (2013).
- [7] A.Frome, G.S.Corrado, J.Shlens, S.Bengio, J.Dean, M.A.Ranzato, and T.Mikolov. 'DeViSE: A Deep Visual-Semantic Embedding Model', In *Proceedings of NIPS 2013*, (2013).
- [8] F.X.Feng, X.J.Wang, and R.F.Li. 'Cross-modal Retrieval with Correspondence Autoencoder', In *Proceedings of MM 2014*, 7-16, (2014).
- [9] D.M.Blei, and M.I.Jordan. 'Modeling Annotated Data', In *Proceedings of SIGIR 2003*, 127-134, (2003).
- [10] C.Wang, D.M.Blei, and L.Fei-Fei. 'Simultaneous Image Classification and Annotation', In *Proceedings of CVPR 2009*, 1903-1910, (2009).
- [11] D.Putthividhya, H.T.Attias, and S.S.Nagarajan. 'Topic Regression Multi-Modal Latent Dirichlet Allocation for Image Annotation', In *Proceedings of CVPR 2010*, 3408-3415, (2010).
- [12] N.Rasiwasia, J.C.Pereira, E.Coviello, G.Doyle, G.R.G.Lanckriet, R.Levy, and N.Vasconcelos. 'A New Approach to Cross-Modal Multimedia Retrieval', In *Proceedings of MM 2010*, 251-260, (2010).
- [13] C.T.Nguyen, N.Kaothanthong, T.Tokuyama, and X.H.Phan. 'A Feature-Word-Topic Model for Image Annotation and Retrieval', *ACM Transactions on the Web* 7(3), Article 12, (2013).
- [14] Z.X.Niu, G.Hua, X.B.Gao, and Q.Tian. 'Semi-supervised Relational Topic Model for Weakly Annotated Image Recognition in Social Media', In *Proceedings of CVPR 2014*, 4233-4240, (2014).
- [15] Y.Zheng, Y.J.Zhang, and H.Larochelle. Topic Modeling of Multimodal Data: An Autoregressive Approach. In *Proceedings of CVPR 2014*, 1370-1377, (2014).
- [16] J.Tian, Y.Huang, Z.Guo, and X.Qi. 'A Multi-Modal Topic Model for Image Annotation Using Text Analysis', In *IEEE Signal Processing Letters* 22(7): 886-890, (2014).
- [17] F.Wu, X.Jiang, X.Li, and S.Tang. Cross-Modal Learning to Rank via Latent Joint Representation, In *IEEE Transactions on Image Processing* 24(5): 1497-1509, (2015).
- [18] R.Girshick, J.Donahue, T.Darrell, and J.Malik. 'Rich feature hierarchies for accurate object detection and semantic segmentation', In *Proceedings of CVPR 2014*, 580-587, (2014).
- [19] B.Hariharan, P.Arbelaez, R.Girshick, and J.Malik. 'Simultaneous Detection and Segmentation', In *Proceedings of ECCV 2014*, 297-312, (2014).
- [20] A.Karpathy, A.Joulin, and L.Fei-Fei. 'Deep Fragment Embeddings for Bidirectional Image Sentence Mapping', In *Proceedings of NIPS 2014*, (2014).
- [21] N.Zhang, J.Donahue, R.Girshick, and T.Darrell. 'Part-Based R-CNNs for Fine-Grained Category Detection', In *Proceedings of ECCV 2014*, 834-849. (2014).
- [22] J.Sivic, and A.Zisserman. 'Video Google: A Text Retrieval Approach to Object Matching in Videos', In *Proceedings of ICCV 2003*, 2:1470-1477, (2003).
- [23] T.Mikolov, I.Sutskever, K.Chen, G.Corrado, and J.Dean. 'Distributed Representations of Words and Phrases and their Compositionality', In *Proceedings of NIPS 2013*, (2013).
- [24] D.Y.Tang, F.R.Wei, B.Qin, M.Zhou, and T.Liu. Building Large-Scale Twitter-Specific Sentiment Lexicon: A Representation Learning Approach. In *Proceedings of COLING 2014*, 172-182, (2014).
- [25] T.L. Griffiths and M. Steyvers, 'Finding scientific topics', In *Proceedings of The National Academy of Sciences of USA*, pp. 5228-5235, (2004).
- [26] T.S.Chua, J.Tang, R.Hong, H.Li, Z.Luo, and Y.T.Zheng. NUS-wide: A real-world web image database from national university of Singapore. In *Proceedings of CIVR 2009*, (2009).
- [27] J.C.Pereira, E.Coviello, G.Doyle, N.Rasiwasia, G.R.G.Lanckriet, R.Levy, and N.Vasconcelos. 'On the Role of Correlation and Abstraction in Cross-Modal Multimedia Retrieval', *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 36(3): 521-535, (2014).

Situation Calculus Game Structures and GDL

Giuseppe De Giacomo¹ and Yves Lespérance² and Adrian R. Pearce³

Abstract. We present a situation calculus-based account of multi-players synchronous games in the style of general game playing. Such games can be represented as action theories of a special form, *situation calculus synchronous game structures (SCSGSs)*, in which we have a single action *tick* whose effects depend on the combination of *moves* selected by the players. Then one can express properties of the game, e.g., winning conditions, playability, weak and strong winnability, etc. in a first-order alternating-time μ -calculus. We discuss verification in this framework considering computational effectiveness. We also show that SCSGSs can be considered as a first-order variant of the Game Description Language (GDL) that supports infinite domains and possibly non-terminating games. We do so by giving a translation of GDL specifications into SCSGSs and showing its correctness. Finally, we show how a player's possible moves can be specified in a Golog-like programming language.

1 Introduction

Many types of problems can be viewed as games, where one or more agents interact to ensure that certain objectives hold no matter how the environment and other agents behave, e.g., contingent planning, service orchestration, controller synthesis, etc. Moreover, general game playing [13], where artificial agents compete in games that are not known in advance, is an important emerging AI testbed. Logics for reasoning about game settings, e.g., [35, 16, 24], has been an active area, with Alternating-Time Temporal Logic (ATL) [1] a popular choice. Model checking techniques have been used to verify properties of games specified in ATL and to synthesize strategies that agents can use to force temporal properties to hold [21]. However, such logics are usually propositional or limited to finite domains. Moreover, the game settings are usually specified using low-level automata-like languages. One exception is the Game Description Language (GDL) [13, 22] developed for the general game playing competition, which is based on logic programming, and allows for a quite high level representation of games. Typically, however, GDL is intended to represent games with finite domains in a declarative way, with a semantics based on “negation as failure” [13, 22, 27].

Within the situation calculus (SitCalc) [23, 26], a well known formalism for reasoning about action based on first-order logic (FOL) (with a second-order axiom to specify the domain of situations), [11] proposes an expressive logical framework for specifying and solving game-like problems. Game settings are specified as a special kind of SitCalc action theory. It is assumed that in any given state, only one agent may act next, and thus the approach is concerned with

turn-taking games. Complex temporal properties of games can be expressed in a first-order (FO) variant of alternating-time μ -calculus. Methods for verification and synthesis based on fixpoint approximation and regression are developed.

In this paper, inspired by [11], we develop a SitCalc-based specification and verification framework, which deals with multi-players *synchronous games*, and is similar in spirit to GDL. Games are represented as action theories of a special form called *situation calculus synchronous game structures (SCSGSs)*, where we have a single action *tick* whose effects depend on the combination of *moves* selected by the players (see Sec. 3). A FO variant of alternating-time μ -calculus is used to specify and verify properties of the game (see Sec. 5), including winning conditions, playability, weak and strong winnability, etc.

The paper's main contributions are:

1. We develop a truly first-order framework that can be used to specify games/systems that involve infinite domains and infinite sets of states.
2. Games can be specified at a high level, using SitCalc action theories [26].
3. SCSGSs amounts to a variant of GDL where states are represented by first-order theories: we give a translation of GDL specifications into SCSGSs and show its soundness and completeness (see Sec. 4).
4. Reasoning techniques developed for the SitCalc can be used to verify properties of games, which can help in analyzing them and developing better players. These includes sound but incomplete techniques that apply to the general setting [11, 17], and techniques that are sound and complete for the decidable “bounded fluent extension” setting [8] (see Sec. 5).
5. Also agent moves can be specified procedurally in a variant of the SitCalc-based programming language Golog [19] (see Sec. 6).
6. Recent verification techniques developed for Golog and ConGolog programs, e.g., [10], can be applied (see again Sec. 6).

Like the original GDL formalism, our account assumes that agents have full observability of the state and all past moves. Handling partial observability, as in GDL-II [33, 30], is left for future work.

2 Preliminaries

The *situation calculus* (SitCalc) is a sorted predicate logic language for representing and reasoning about dynamically changing worlds [23, 26]. It includes three sorts, *Actions*, *Situations* and *Objects*. All changes to the world are the result of *actions*, which are terms in the logic. A possible world history is represented by a term called a *situation*. The constant S_0 is used to denote the initial situation where no actions have yet been done. Sequences of actions are built using the function symbol *do*, where $do(a, s)$ denotes the successor situation resulting from performing action a in situation s . Predicates and

¹ Dip. di Ingegneria Informatica, Automatica e Gestionale, Sapienza – Università di Roma, Rome, Italy, email: degiacomo@dis.uniroma1.it

² Dept. of Electrical Engineering and Computer Science, York University, Toronto, ON, Canada, email: lesperan@cse.yorku.ca

³ Dept. of Computer Science and Software Engineering, University of Melbourne, Victoria, Australia, email: adrianrp@unimelb.edu.au

functions whose value varies from situation to situation are called *fluents*, and are denoted by symbols taking a situation term as their last argument (e.g., *Holding*(x, s)). Actions and fluents (except for the last argument) can only take arguments of sort *Objects*. Notice that we allow the object domain to be infinite. Within this language, we can formulate action theories that describe how the world changes as the result of actions. Here, we concentrate on *basic action theories* as proposed in [26]. A *basic action theory* \mathcal{D} is the union of the following disjoint sets: the foundational, domain independent, axioms of the SitCalc (Σ); unique name axioms for actions; precondition axioms stating when actions can be legally performed (\mathcal{D}_{poss}); successor state axioms describing how fluents change between situations (\mathcal{D}_{ssa}); and axioms describing the initial configuration of the world (\mathcal{D}_{S_0}). A special predicate *Poss*(a, s) is used to state that action a is executable in situation s ; precondition axioms in \mathcal{D}_{poss} characterize this predicate. We say that a situation s (corresponding to a sequence of actions) is executable, written *Executable*(s), if every action performed in reaching s is possible in the situation it occurred [26]. In turn, successor state axioms encode the causal laws of the domain; they take the place of the so-called effect axioms and provide a solution to the frame problem.

3 Synchronous Game Structures

We focus on games where there are n players/agents each of whom chooses a move at every time step. All such moves are executed *synchronously* and determine the next state of the game. At each time step, the state of the game is fully observable by all agents, as are all past moves of every agent. This is in agreement with the assumptions built into GDL [13, 22]. To represent such multi-player synchronous games, we define a special class of basic action theories, called *situation calculus synchronous game structures* (SCSGSs), which are defined as follows.

Agents A SCSGS involves a finite set of n agents, and we introduce a subsort *Agents* of *Objects* which includes these finitely many agents Ag_1, \dots, Ag_n , each denoted by a constant, and for which unique names $Ag_i \neq Ag_j$ for $i \neq j$ and domain closure $Agent(x) \equiv x = Ag_1 \vee \dots \vee x = Ag_n$ hold.

Moves. We also introduce a second subsort *Moves* of *Objects*, representing the possible moves of the agents. These come in finitely many types, represented by function symbols $M_i(\vec{x})$, which are parametrized by objects \vec{x} and we have $Move(m) \equiv \bigvee_i \exists \vec{x}. m = M_i(\vec{x})$. Given that the parameters range over *Objects*, each agent may have an infinite number of possible moves at each time step. We have unique name and domain closure axioms (parametrized by objects) for these functions $M_i(\vec{x}) \neq M_j(\vec{y})$ for $i \neq j$, and $M_i(\vec{x}) = M_i(\vec{y}) \supset \vec{x} = \vec{y}$.

Actions. In SCSGSs, there is only *one action type*, $tick(m_1, \dots, m_n)$, which represents the execution of a joint move by all the agents at a given time step. The action *tick* has exactly n parameters, m_1, \dots, m_n , one per agent, which are of sort *Moves* and corresponds to the simultaneous choice of the move to perform by the n different agents.

Legal moves. A key component of a SCSGS is a characterization of the *legal* moves available to each agent in a given situation. This is specified formally using a special predicate *LegalM*, which is defined by statements of the following form (one for each agent Ag_i and move type M_i):

$$LegalM(Ag_i, M_i(\vec{x}), s) \doteq \Phi_{Ag_i, M_i}(\vec{x}, s)$$

meaning that agent Ag_i can legally perform move $M_i(\vec{x})$ in situation s if and only if $\Phi_{Ag_i, M_i}(\vec{x}, s)$ holds. Technically *LegalM* is an abbreviation for $\Phi_{Ag_i, M_i}(\vec{x}, s)$, which is a uniform formula (i.e., a formula that only refers to a single situation s).

Precondition axioms. The precondition axiom for the action *tick* is fixed and specified in terms of *LegalM* as follows:

$$Poss(tick(m_1, \dots, m_n), s) \equiv \bigwedge_{i=1, \dots, n} LegalM(Ag_i, m_i, s)$$

This states that action $tick(m_1, \dots, m_n)$, denoting the joint move of all agents, can be performed if and only if each selected move m_i is a legal move for agent Ag_i in situation s . Since we only have one action type *tick*, this is the only precondition axiom in \mathcal{D}_{poss} .

Successor state axioms. We have *successor state axioms* \mathcal{D}_{ssa} , specifying the effects and frame conditions of the joint moves $tick(m_1, \dots, m_n)$ on the fluents. Such axioms, as usual in basic action theories, are domain specific, and characterize the actual game under consideration. Within such axioms, the agent moves, which occur as parameters of *tick*, determine how fluents change as the result of joint moves.⁴

Initial situation description. Finally, the initial state of the game is axiomatized in the *initial situation description* \mathcal{D}_0 as usual, in a domain specific way.

Example 1 Consider the following example drawn from [29]. There are two guard agents, Ag_1 and Ag_2 , that cooperatively try to catch a third agent, Ag_3 , who is trying to escape, in a 5×5 grid world. Ag_3 is initially at location (5, 5) and can escape after reaching any of the other corners. Initially, Ag_1 is at location (1, 1) and Ag_2 at (1, 5). At each time step, the agents can all move synchronously to an adjacent square. Ag_3 is caught and loses the game if he ends up on the same square as one of the guards or if he crosses path with one of them in a simultaneous move. We can specify this game as follows. We have 3 possible moves, with the following definitions:

$$\begin{aligned} LegalM(ag, move(d), s) &\doteq \exists u, v, x, y. \neg Terminal(s) \wedge \\ &At(ag, u, v, s) \wedge Adj(u, v, d, x, y) \\ LegalM(ag, Stay, s) &\doteq \exists x, y. At(ag, x, y, s) \\ LegalM(ag, Exit, s) &\doteq \\ &ag = Ag_3 \wedge \neg Terminal(s) \wedge AtExit(Ag_3, s) \end{aligned}$$

Thus an agent ag may perform move $move(d)$ in s to move one step in direction d provided that the game is not yet finished and moving in direction d is possible given ag 's position in s . An agent may also perform move *Stay* in a situation s to remain where he is provided that he is on the grid in s . Finally an agent may perform move *Exit* in s to exit the grid provided he is Ag_3 , the game is not yet finished, and he is at an exit position in s . *Terminal*(s), meaning that the game is finished in situation s , holds if Ag_3 is at the same position as one of the other agents in s and is "captured", in which case Ag_1 and Ag_2 win, or if Ag_3 has "exited" the grid in s , in which case Ag_3 wins, and is defined as:

$$\begin{aligned} Terminal(s) &\doteq \exists x, y. At(Ag_1, x, y, s) \wedge At(Ag_3, x, y, s) \vee \\ &\exists x, y. At(Ag_2, x, y, s) \wedge At(Ag_3, x, y, s) \vee \\ &\neg \exists x, y. At(Ag_3, x, y) \\ Wins(ag, s) &\doteq Terminal(s) \wedge \\ &ag = Ag_3 \wedge \neg \exists x, y. At(Ag_3, x, y, s) \vee \\ &ag \neq Ag_3 \wedge \exists x, y. At(Ag_3, x, y, s) \wedge At(ag, x, y, s) \end{aligned}$$

⁴ In many cases, moves don't interfere with each other and the effects are just the union of those of each move. One can also exploit previous work on axiomatizing parallel actions to generate successor state axioms [26, 25].

$$\begin{aligned} AtExit(ag, s) &\doteq \\ &At(ag, 1, 1, s) \vee At(ag, 5, 1, s) \vee At(ag, 1, 5, s) \end{aligned}$$

The following successor state axiom specifies how the game state changes:

$$\begin{aligned} At(ag, x, y, do(a, s)) &\equiv \exists u, v. MovesTo(ag, u, v, x, y, a, s) \vee \\ &At(ag, x, y, s) \wedge \neg \exists u, v. MovesTo(ag, x, y, u, v, a, s) \wedge \\ &\neg \exists m_1, m_2. (a = tick(m_1, m_2, Exit) \wedge ag = Ag_3) \\ MovesTo(ag, u, v, x, y, a, s) &\doteq \exists m_1, m_2, m_3, d. \\ a = tick(m_1, m_2, m_3) \wedge At(ag, u, v, s) \wedge Adj(u, v, d, x, y) \wedge \\ [ag = Ag_1 \wedge m_1 = move(d) \vee ag = Ag_2 \wedge m_2 = move(d) \vee \\ ag = Ag_3 \wedge m_3 = move(d) \wedge \neg Capturing(Ag_3, m_3, Ag_1, \\ m_1, s) \wedge \neg Capturing(Ag_3, m_3, Ag_2, m_2, s)] \\ Capturing(ag, m, ag', m', s) &\doteq \exists x, y, u, v, d, d'. ag = Ag_3 \wedge \\ At(Ag_3, x, y, s) \wedge (ag' = Ag_1 \vee ag' = Ag_2) \wedge \\ At(ag', u, v, s) \wedge m = move(d) \wedge m' = move(d') \wedge \\ Adj(x, y, d, u, v) \wedge Adj(u, v, d', x, y) \end{aligned}$$

The successor state axiom for *At* essentially says that agent *ag* moves to position (x, y) in situation $do(a, s)$ if *a* is a tick joint move where *ag* performs a move in direction *d*, from a position (u, v) , which is adjacent from (x, y) in direction *d*, and moreover if *ag* is Ag_3 , then he is not “being captured” by one of the other agents (see below). Otherwise, *ag* remains at the position where he was in situation *s* except if *ag* is Ag_3 and he performs move *Exit*, in which case he will no longer be at any position on the grid. The axiom uses a defined fluent (*an abbreviation*), *Capturing*(*ag, m, ag', m', s*), meaning that *ag'* performing move *m'* is capturing agent *ag* performing move *m* in situation *s*, which holds if and only if *ag* is Ag_3 and *ag'* is one of the other agents, and *ag* and *ag'* are performing moves that overlap, i.e., where the starting position of one move is the ending position of another.

The following axioms define the non-fluent predicates that we use:

$$\begin{aligned} Agent(ag) &\equiv ag = Ag_1 \vee ag = Ag_2 \vee ag = Ag_3 \\ Move(m) &\equiv \exists d. Dir(d) \wedge m = move(d) \vee \\ &m = Stay \vee m = Exit \\ Dir(d) &\equiv d = N \vee D = E \vee \dots \text{— directions} \\ &d = S \vee d = W \\ Co(x) &\equiv x = 1 \vee x = 2 \vee \dots \text{— coordinates} \\ &x = 3 \vee x = 4 \vee x = 5 \\ Succ(x, y) &\equiv \dots \text{— } y \text{ is successor of } x \\ &x = 1 \wedge y = 2 \vee x = 2 \wedge y = 3 \vee \\ &x = 3 \wedge y = 4 \vee x = 4 \wedge y = 5 \\ Adj(x, y, d, x', y') &\equiv \dots \text{— } (x', y') \text{ is adjacent from } (x, y) \\ &d = N \wedge x' = x \wedge Co(x) \wedge Succ(y, y') \vee \text{ in direction } d \\ &d = S \wedge x' = x \wedge Co(x) \wedge Succ(y', y) \vee \\ &d = E \wedge y' = y \wedge Co(y) \wedge Succ(x, x') \vee \\ &d = W \wedge y' = y \wedge Co(y) \wedge Succ(x', x) \end{aligned}$$

The initial state is specified as follows:

$$\begin{aligned} At(ag, x, y, S_0) &\equiv ag = Ag_1 \wedge x = 1 \wedge y = 1 \vee \\ &ag = Ag_2 \wedge x = 5 \wedge y = 1 \vee ag = Ag_3 \wedge x = 1 \wedge y = 5 \end{aligned}$$

The precondition axiom for the tick action is as discussed earlier. We also have unique names for moves, agents, directions, and positions. \square

Note that it easy to obtain an infinite states version of this game, for instance, by using an infinite grid, with positions (x, y) for all $x, y \in \mathbb{N}$. Then Ag_3 can run away to avoid getting caught and the others cannot corner her. We can then say that the game ends if the

guards catch Ag_3 or if Ag_3 gets North and East of both guards past a given area (they can't catch up if he keeps going North-East).

Let's now consider another simple example which has infinite states. It is not a game in the traditional sense, but we can analyse what properties agents can enforce in it.

Example 2 We have a repair shop where items arrive, are repaired, and then shipped. Items are denoted by a countably infinite set of constants $Item_1, Item_2, \dots$, for which we have unique name axioms. Ag_1 represents the environment, Ag_2 is a repairing robot, and Ag_3 is a shipper agent. We have the following legal move axioms:

$$\begin{aligned} LegalM(ag, Wait, s) &\doteq True \\ LegalM(ag, arrive(i), s) &\doteq \\ &ag = Ag_1 \wedge Item(i) \wedge \neg InShop(i, s) \\ LegalM(ag, repair(i), s) &\doteq \\ &ag = Ag_2 \wedge InShop(i, s) \\ LegalM(ag, ship(i), s) &\doteq \\ &ag = Ag_3 \wedge Repaired(i, s) \end{aligned}$$

and the following successor state axioms:

$$\begin{aligned} InShop(i, do(a, s)) &\equiv \exists m, m'. a = tick(arrive(i), m, m') \\ &\vee InShop(i, s) \wedge \neg \exists m, m'. a = tick(m, m', ship(i)) \\ Repaired(i, do(a, s)) &\equiv \exists m, m'. a = tick(m, repair(i), m') \\ &\vee Repaired(i, s) \wedge \neg \exists m, m'. a = tick(arrive(i), m, m') \\ Shipped(i, do(a, s)) &\equiv \exists m, m'. a = tick(m, m', ship(i)) \\ &\vee Shipped(i, s) \wedge \neg \exists m, m'. a = tick(arrive(i), m, m') \end{aligned}$$

We also have initial state axioms saying that initially no items are in the shop, or have been repaired or shipped. Clearly, the domain is infinite, as is the number of moves. \square

4 Relationship with GDL

SCSGSs are closely related to GDL specifications. We show that GDL game descriptions where auxiliary predicates are “acyclic” or “hierarchical” (without direct or indirect recursion) [20] can be translated into SCGSs. Notice that formalisms in which the state description is based on first-order logic (FOL), such as the situation calculus, cannot capture predicates on state defined recursively.

We first define a translation function $\tau_{a,s}$ for translating the bodies of the rules for defining the initial situation, next situation, and legal moves; only **true**, **does** atoms and auxiliary predicates $aux(\vec{x})$ can occur in bodies:

$$\begin{aligned} \tau_{a,s}(true) &= true \\ \tau_{a,s}(true(F(\vec{t}))) &= F(\vec{t})[s] \\ \tau_{a,s}(does(R, M)) &= \exists m_1 \dots \exists m_{R-1} \exists m_{R+1} \dots \exists m_n \\ &a = tick(m_1, \dots, m_{R-1}, M, m_{R+1}, \dots, m_n) \\ \tau_{a,s}(aux(\vec{x})) &= \exists \vec{y}. \tau_{a,s}(body_{aux}(\vec{x}, \vec{y})) \\ \tau_{a,s}(\alpha_1 \wedge \alpha_2) &= \tau_{a,s}(\alpha_1) \wedge \tau_{a,s}(\alpha_2) \\ \tau_{a,s}(\neg \alpha) &= \neg \tau_{a,s}(\alpha) \end{aligned}$$

where $body_{aux}(\vec{x}, \vec{y})$ denotes the body of the rule for $aux(\vec{x})$ (which may involve disjunctions).

Initial situation. In GDL, the initial situation is specified by a set of clauses of the form $init(F(\vec{t})) \leftarrow body(\vec{t}, \vec{y})$, where $body(\vec{t}, \vec{y})$ includes only **true** atoms and auxiliary predicates (facts are represented as $init(F(\vec{t})) \leftarrow true$), involving terms \vec{t} and additional existential variables \vec{y} . In the SitCalc, we capture this through a set of FOL formulas:

$$F(\vec{x}, S_0) \equiv \bigvee_{init(F(\vec{t})) \leftarrow body(\vec{t}, \vec{y})} \vec{x} = \vec{t} \wedge \exists \vec{y}. \tau_{a,S_0}(body(\vec{t}, \vec{y}))$$

This is the familiar completion of the set of `init` clauses, which captures their semantics given that the set of clauses is acyclic. Note that since the bodies of these clauses cannot contain `does` atoms, the action parameter of τ is irrelevant. We have a complete specification, so there is a single model.

Effects. In GDL, the next state resulting from moves is specified by a set of clauses of the form $\text{next}(F(\vec{t})) \leftarrow \text{body}(\vec{t}, \vec{y})$, where body includes only `true` and `does` atoms, and auxiliary predicates. In the SitCalc, we capture this description through successor state axioms of the form:

$$F(\vec{x}, \text{do}(a, s)) \equiv \bigvee_{\text{next}(F(\vec{t})) \leftarrow \text{body}(\vec{t}, \vec{y})} \vec{x} = \vec{t} \wedge \exists \vec{y}. \tau_{a,s}(\text{body}(\vec{t}, \vec{y}))$$

Preconditions and legality. In GDL, the legality conditions for a move M by a role R are expressed by a set of clauses of the form $\text{legal}(R, M(\vec{t})) \leftarrow \text{body}(\vec{t}, \vec{y})$, where body contains only `true` and auxiliary predicates. In the SitCalc, we capture this through axioms of the form:

$$\text{Legal}M(R, M(\vec{x}), s) \equiv \bigvee_{\text{legal}(R, M(\vec{t})) \leftarrow \text{body}(\vec{t}, \vec{y})} \vec{x} = \vec{t} \wedge \exists \vec{y}. \tau_{.,s}(\text{body}(\vec{t}, \vec{y}))$$

The preconditions of the `tick` joint move action are specified by the action precondition axiom given earlier.

Goals and terminal states. For GDL goals, we have clauses of the form $\text{goal}(R, V) \leftarrow \text{body}(\vec{t}, \vec{y})$, where body includes only `true` and auxiliary predicates. In the SitCalc, we have:

$$\text{Goal}(r, v, s) \equiv \bigvee_{\text{goal}(R, V) \leftarrow \text{body}(\vec{t}, \vec{y})} r = R \wedge v = V \wedge \exists \vec{y}. \tau_{.,s}(\text{body}(\vec{t}, \vec{y}))$$

Similarly for defining termination, we have in GDL clauses of the form $\text{terminal} \leftarrow \text{body}(\vec{y})$, where body includes only `true` and auxiliary predicates. So we have:

$$\text{Terminal}(s) \equiv \bigvee_{\text{terminal} \leftarrow \text{body}(\vec{y})} \exists \vec{y}. \tau_{.,s}(\text{body}(\vec{y}))$$

Unique name and domain closure for objects. We additionally need to impose the unique name assumption and domain closure for the object sort in the SitCalc to conform to the GDL assumption that object terms are interpreted as themselves. In the SitCalc this corresponds to assuming we have standard names for objects [18].

We can now show that the above mapping is correct.

Theorem 3 *For any GDL specification that uses acyclic auxiliary predicates only, the above translation is correct, i.e., it produces a SCSGS whose only model is bisimilar to the transition system associated with the GDL specification.*

Proof (sketch). Notice that we have complete information. This means the resulting SCSGS \mathcal{D} has only one SitCalc model \mathcal{M} (up to isomorphism). We can associate to such a model \mathcal{M} a transition system $T_{\mathcal{M}} = \langle \Delta, \mathcal{S}, S_0, \rightarrow_{\mathcal{M}}, L_{\mathcal{M}} \rangle$ induced by \mathcal{M} where:

- Δ is the object domain of \mathcal{M} , which is isomorphic to the set of all ground object terms since we have unique name and domain closure for objects.
- \mathcal{S} is the set of possible states formed by all situations;
- $S_0 \in \mathcal{S}$ is the initial state, where S_0 is the initial situation;
- $\rightarrow_{\mathcal{M}} \subseteq \mathcal{S} \times \mathcal{S}$ is the transition relation s.t. $s \rightarrow_{\mathcal{M}} s'$ iff there exists some a s.t. $s' = \text{do}^{\mathcal{M}}(a, s)$ and $(a, s) \in \text{Poss}^{\mathcal{M}}$; note that a will be some instantiation of the `tick` action type for some move arguments;

- $L_{\mathcal{M}} : \mathcal{S} \mapsto \text{Int}^{\mathcal{M}}$ is the labeling function associating each state/situation s with a first-order (FO) interpretation $I = L_{\mathcal{M}}(s)$ s.t. $F^I = \{\vec{o} \mid \mathcal{M} \models F(\vec{o}, s)\}$, for every predicate fluent.

On the other hand, one can use the techniques in [29] to generate a transition system for the GDL specification G . We can associate to such a game description G a transition system $T_G = \langle \Delta, Q, q_0, \rightarrow_G, L_G \rangle$ where:

- Δ is the set of all ground object terms.
- Q is the set of possible states formed by all possible finite subsets of ground “fluent” atoms;
- $q_0 = \{F(\vec{t}) \mid G \models \text{init}(F(\vec{t}))\}$;
- $\rightarrow_G \subseteq Q \times Q$ is the transition relation s.t. $q \rightarrow_G q'$ iff there exists some ground move terms M_1, \dots, M_n s.t. $G \cup q \models \text{does}(Ag_i, M_i)$ (for $i = 1, \dots, n$) and $q' = \{F(\vec{t}) \mid G \cup q \cup \{\text{does}(Ag_1, M_1), \dots, \text{does}(Ag_n, M_n)\} \models \text{next}(F(\vec{t}))\}$;
- $L_G : Q \mapsto \text{Int}^G$ is the labeling function associating each state q with a FO interpretation $I = L_G(q)$ s.t.
 - $\text{Goal}^I = \{(Ag_i, v) \mid G \cup q \models \text{goal}(Ag_i, v)\}$,
 - $\text{Terminal}^I = \text{true}$ iff $G \cup q \models \text{terminal}$, and
 - $F^I = \{\vec{t} \mid G \cup q \models F(\vec{t})\}$ for all other predicates.

The two transition systems $T_{\mathcal{M}}$ and T_G are bisimilar. Indeed there is a relation \mathcal{B} including (S_0, q_0) such that if $(s, q) \in \mathcal{B}$ then: (i) $L_{\mathcal{M}}(s)$ is isomorphic to $L_G(q)$; (ii) for all s' such that $s \rightarrow_{\mathcal{M}} s'$, there exists a q' such that $q \rightarrow_G q'$ and $(s', q') \in \mathcal{B}$; (iii) for all q' such that $q \rightarrow_G q'$, there exists a s' such that $s \rightarrow_{\mathcal{M}} s'$ and $(s', q') \in \mathcal{B}$. One can check that one such relation is the isomorphism between state labeling of the transitions systems: i.e. $\mathcal{B} = \{(s, q) \mid L_{\mathcal{M}}(s) \text{ is isomorphic to } L_G(q)\}$. Given the bisimilarity-invariance of the μ -calculus, we get that the two transition systems satisfy the same $\mu\text{ATL-FO}$ formulas (see Sec. 5). \square

Notice that [34] shows trace-equivalence between GDL specifications and their translation into the C^+ action language [14]. Remember that bisimilarity implies trace-equivalence. GDL also requires that game specifications be stratified, “allowed”, and satisfy some restrictions on recursion that ensure that the specification is equivalent to a finite set of ground clauses [34]. In principle, when the game is finite state as assumed in [13], we could drop the acyclicity restriction and capture GDL in its entirety at the cost of compositionality.

5 Verification

To express properties about SCSGSs, we introduce a specific logic $\mu\text{ATL-FO}$, inspired by alternating-time μ -calculus, μATL , which is a well-known generalization of ATL [1]. Our logic is a first-order variant of the μ -calculus [2] that works on games, by suitably considering coalitions acting towards the realization of a temporally extended goal, as in μATL . The key building block in these kinds of logics is the so-called *force-next* operator, which in our case is:

$$\begin{aligned} \langle\langle G \rangle\rangle \circ \varphi \equiv & \exists m_{g_1}, \dots, m_{g_k} \cdot \bigwedge_{\{g_i, \dots, g_k\} = G} \text{Legal}M(g_i, m_{g_i}, \text{now}) \wedge \\ & \exists m_{g_{k+1}}, \dots, m_{g_n} \cdot \bigwedge_{\{g_{k+1}, \dots, g_n\} = \bar{G}} \text{Legal}M(g_i, m_{g_i}, \text{now}) \wedge \\ & \forall m_{g_{k+1}}, \dots, m_{g_n} \cdot \bigwedge_{\{g_{k+1}, \dots, g_n\} = \bar{G}} \text{Legal}M(g_i, m_{g_i}, \text{now}) \\ & \supset \varphi(\text{do}(\text{tick}(m_{g_1}, \dots, m_{g_n}), \text{now})) \end{aligned}$$

Above, φ is a situation suppressed formula, i.e., one with situation arguments in fluents suppressed (syntactically replaced by a placeholder *now*). We denote by $\varphi[s]$ the formula obtained by restoring

the suppressed situation argument s into all fluents in φ . Here, we quantify existentially on legal moves when the agent is in the coalition G , and universally when it is not. We are looking for some move for each agent in the coalition G , such that for all moves by the agents not in the coalition, φ becomes true next. Notice that in any case both agents in the coalition and agents outside it must have a legal move.

Then, following [11], we define the logic $\mu\text{ATL-FO}$ as:

$$\Psi \leftarrow \varphi \mid Z \mid \neg\Psi \mid \Psi_1 \wedge \Psi_2 \mid \exists x.\Psi \mid \langle\langle G \rangle\rangle \circ \Psi \mid \mu Z.\Psi(Z)$$

where φ is an arbitrary, possibly open, situation-suppressed SitCalc uniform formula, Z is a predicate variable of a given arity, and $\langle\langle G \rangle\rangle \circ \Psi$ is as defined above. $\mu Z.\Psi(Z)$ is the *least fixpoint* construct from the μ -calculus, which denotes the least fixpoint of the formula $\Psi(Z)$ (we use this notation to emphasize that Z may occur free, i.e., not quantified by μ in Ψ). Similarly $\nu Z.\Psi(Z)$, defined as $\neg\mu Z.\neg\Phi[Z/\neg Z]$ (where we denote with $\Phi[Z/\neg Z]$ the formula obtained from Φ by substituting each occurrence of Z with $\neg Z$), denotes the *greatest fixpoint* of $\Psi(Z)$. We also use the usual abbreviations for first-order logic such as disjunction (\vee) and universal quantification \forall . Moreover we denote by $[[G]] \circ \Psi$ the dual of $\langle\langle G \rangle\rangle \circ \Psi$, i.e., $[[G]] \circ \Psi \doteq \neg\langle\langle G \rangle\rangle \circ \neg\Psi$.

As usual in the μ -calculus, formulas of the form $\mu Z.\Psi(Z)$ (and $\nu Z.\Psi(Z)$) must obey the *syntactic monotonicity* of $\Psi(\cdot)$ w.r.t. Z , which states that every occurrence of the second-order variable Z in $\Psi(Z)$ must be within the scope of an even number of negation symbols. This ensures that both the least fixpoint $\mu Z.\Psi(Z)$ and the greatest fixpoint $\nu Z.\Psi(Z)$ always exist.

The least fixpoint formula $\mu Z.\Psi$ is true in a situation if and only if it belongs to the least set of situations Z that satisfy the temporal formula $\Psi(Z)$, where Z is a second-order predicate variable ranging over sets of situations (a formal semantics is given below). Similarly, the greatest fixpoint formula $\nu Z.\Psi$ holds in a situation if it belongs to the largest set of situations Z that satisfy $\Psi(Z)$. Using these least and greatest fixpoint constructs, we can express the ability of *forcing* arbitrary temporal and dynamic properties. For instance, to say that group G has a strategy to force achieving $\varphi(\vec{x})$ eventually, where $\varphi(\vec{x})$ is a situation suppressed formula with free variables \vec{x} , we use the following least fixpoint formula:

$$\mu Z. \varphi(\vec{x}) \vee \langle\langle G \rangle\rangle \circ Z$$

In a first-order ATL, this could be expressed as $\langle\langle G \rangle\rangle \diamond \varphi(\vec{x})$. Similarly, we use the greatest fixpoint construct to express the ability of a coalition G to force maintaining property φ :

$$\nu Z. \varphi(\vec{x}) \wedge \langle\langle G \rangle\rangle \circ Z$$

In a first-order ATL, this could be expressed as $\langle\langle G \rangle\rangle \square \varphi(\vec{x})$.

The formal semantics of $\mu\text{ATL-FO}$ is based on characterizing how to evaluate $\mu\text{ATL-FO}$ formulas in a SitCalc model \mathcal{M} . To do so, since $\mu\text{ATL-FO}$ contains formulas with both individual and predicate free variables, we need to introduce an individual variable valuation v , and a predicate variable valuation V , i.e., a mapping from predicate variables Z to subsets of the set of all situations \mathcal{S} . Then, we assign meaning to formulas by associating to \mathcal{M} , v , and V an *extension function* $(\cdot)_{v,V}^{\mathcal{M}}$, which maps formulas to subsets of \mathcal{S} , and is defined inductively as follows:

$$\begin{aligned} (\varphi)_{v,V}^{\mathcal{M}} &= \{s \in \mathcal{S} \mid \mathcal{M} \models \varphi[s]\} \\ (\neg\Psi)_{v,V}^{\mathcal{M}} &= \mathcal{S} - (\Psi)_{v,V}^{\mathcal{M}} \\ (\Psi_1 \wedge \Psi_2)_{v,V}^{\mathcal{M}} &= (\Psi_1)_{v,V}^{\mathcal{M}} \cap (\Psi_2)_{v,V}^{\mathcal{M}} \\ (\exists x.\Psi)_{v,V}^{\mathcal{M}} &= \{s \in \mathcal{S} \mid \text{exists } t \text{ s.t. } s \in (\Psi)_{v[x/t],V}^{\mathcal{M}}\} \\ (\langle\langle G \rangle\rangle \circ \Psi)_{v,V}^{\mathcal{M}} &= \{s \in \mathcal{S} \mid s \in \text{Pre}(G, (\Psi)_{v,V}^{\mathcal{M}})\} \\ (Z(\vec{t}))_{v,V}^{\mathcal{M}} &= V(Z) \\ (\mu Z.\Psi)_{v,V}^{\mathcal{M}} &= \bigcap \{\mathcal{E} \subseteq \mathcal{S} \mid (\Psi)_{v,V[Z/\mathcal{E}]}^{\mathcal{M}} \subseteq \mathcal{E}\} \end{aligned}$$

where:

$$\begin{aligned} \text{Pre}(G, \mathcal{E}) &= \{s \in \mathcal{S} \mid \\ &\exists m_{g_1}, \dots, m_{g_k} \cdot \bigwedge_{\{g_i, \dots, g_k\}=G} (\mathcal{M} \models \text{LegalM}(g_i, m_{g_i}, s)) \wedge \\ &\exists m_{g_{k+1}}, \dots, m_{g_n} \cdot \bigwedge_{\{g_{k+1}, \dots, g_n\}=\bar{G}} (\mathcal{M} \models \text{LegalM}(g_i, m_{g_i}, s)) \wedge \\ &\forall m_{g_{k+1}}, \dots, m_{g_n} \cdot \bigwedge_{\{g_{k+1}, \dots, g_n\}=\bar{G}} (\mathcal{M} \models \text{LegalM}(g_i, m_{g_i}, s)) \\ &\supset \text{do}(\text{tick}(m_{g_1}, \dots, m_{g_n}), s) \in \mathcal{E}\} \end{aligned}$$

Note that given a valuation V and a predicate variable Z and a set of situations \mathcal{E} we denote by $V[Z/\mathcal{E}]$ the valuation obtained from V by changing the value of Z to \mathcal{E} . Similarly for v . Notice also that when a $\mu\text{ATL-FO}$ formula Ψ is closed (w.r.t. individual and predicate variables), its extension $(\Psi)_{v,V}^{\mathcal{M}}$ does not depend on the valuations v and V , and we denote the extension of Ψ simply by $(\Psi)^{\mathcal{M}}$. We say that a closed formula Ψ holds in the SitCalc model \mathcal{M} , denoted by $\mathcal{M} \models \Psi$, if $S_0 \in (\Psi)^{\mathcal{M}}$.

Example 4 Several key properties of games [13] can easily be expressed in $\mu\text{ATL-FO}$, for example:

- Playability, i.e., at every step which is not terminal there exists a legal joint move:

$$\nu Z. \text{Terminal} \vee \langle\langle \text{ALL} \rangle\rangle \circ Z$$

- Termination, i.e., there is a way of playing the game that eventually leads to termination:

$$\mu Z. \text{Terminal} \vee \langle\langle \text{ALL} \rangle\rangle \circ Z$$

- Weak Winnability (by agent Ag), i.e., there is a way for agent Ag to win if the others cooperate:

$$\mu Z. \text{Terminal} \wedge \exists v. \text{Goal}(Ag, v) \wedge (\bigwedge_{Ag' \neq Ag} \exists v'. \text{Goal}(Ag', v') \wedge v' \leq v) \vee \langle\langle \text{ALL} \rangle\rangle \circ Z$$

- Strong Winnability (by agent Ag), i.e., there is a way for agent Ag to win no matter what the others do:

$$\mu Z. \text{Terminal} \wedge \exists v. \text{Goal}(Ag, v) \wedge (\bigwedge_{Ag' \neq Ag} \exists v'. \text{Goal}(Ag', v') \wedge v' \leq v) \vee \langle\langle \{Ag\} \rangle\rangle \circ Z$$

- Well-formed: if terminating, playable and weakly winnable.

In Example 1, one can check that the game is weakly winnable for all agents, and that it becomes strongly winnable for either Ag_3 or for the coalition $\{Ag_1, Ag_2\}$ starting from certain initial configurations (where we change the initial position of some players). In Example 2, one can check that Ag_2 can ensure that all items that arrive are eventually repaired, and that Ag_2 and Ag_3 together can ensure that all are eventually shipped. \square

Let us now discuss how one can effectively verify $\mu\text{ATL-FO}$ formulas against a SCSGS in three key cases.

Propositional case. The propositional case is the one where the object domain is assumed to be finite. If this is the case, actions are also finite (we have finite moves types and only one action type). Hence, the only domain that remains infinite is that of situations (though now the situation tree is only finitely branching). However successor state axioms ensure that fluents in a given situation depend only on the values of the fluents in the previous situation (not the history). Hence in the presence of a finite object domain, one can abstract situations into “states”, which are the interpretation of the fluents for that situation [32]. Consequently one can show that there is a finite transition system bisimilar to the SitCalc model, for example along the lines of [8]. Given the invariance with respect to bisimulation of the μ -calculus, one can use such a finite transition system for the evaluation, or model checking, of the μ ATL-FO formulas (notice also that first-order quantification can be eliminated because of the finite object domain). Thus we have the following result:

Theorem 5 *Let \mathcal{D} be a SCSGS with a finite object domain and Ψ a μ ATL-FO formula. Then checking whether $\mathcal{D} \models \Psi$ is decidable.*

In practice μ ATL-FO reduces to standard alternating-time μ -calculus (μ ATL) [1], and one can use standard algorithms and tools for the verification. In fact such tools can also be used for synthesis by considering that strategies can be extracted from the existential choices in the $\langle\langle G \rangle\rangle \circ \varphi$ operators as discussed in [1]. Verification techniques for propositional GDL descriptions have been proposed in [27].

Bounded first-order case. We say that a SitCalc theory is bounded if in spite of having an infinite object domain, it allows only a bounded number of object tuples in the extension of fluents in each situation [8]. Intuitively this is like saying that we have a bookshelf of a fixed size in which we can freely add, remove and replace books as long as we remain within the fixed size of the bookshelf. For instance, we can obtain an infinite-states bounded version of Example 2 as follows: we make the *arrive*(i) move illegal if there are already k items in the shop; we make *Repaired* become false when an item is shipped, so it is also bounded by k ; finally we replace *Shipped*(i, s) by *JustShipped*(i, s), which only holds in the situation that follows the *ship*(i) action, i.e., for at most one item (*Move*(m) and *Item*(i)) can be viewed as abbreviations, the latter standing for anything that is not an agent or move). Such bounded action theories are known to be decidable for model checking a first-order variant of the μ -calculus without first-order quantification across situations [8] as well as with quantification across [8, 15, 3]. Such results can be adapted to show that μ ATL-FO model checking against SCSGSs is decidable:

Theorem 6 *Let \mathcal{D} be a SCSGS that is a bounded action theory and Ψ a μ ATL-FO formula. Then checking whether $\mathcal{D} \models \Psi$ is decidable.*

Proof (sketch). If Ψ does not include first-order quantification across situations, we can apply the techniques in [8] which allow for building a finite transition system that is bisimilar to the one induced by the SitCalc theory model (which is essentially unique if we assume complete information). If quantification across is allowed, we cannot use the techniques in [8] in general, but we can still use a similar finite faithful abstraction if the quantification is over objects in the *active domain* and is restricted to be *persistence-preserving* [9]. Finally if we allow unrestricted quantification, we can still generate a finite faithful abstraction, which however, in this case depends on the number of variables in Ψ as well [3, 4]. The key to adapting the original proofs to our case is to suitably reformulate the preimage construction used in evaluating μ ATL-FO formulas, so has to handle the coalition existentially and the adversaries universally. \square

Synthesis can be done as in the propositional case.

General first-order case. In the general case, model checking of μ ATL-FO in SCSGSs is undecidable. For example it is immediate to reconstruct the undecidability result in [15] using very simple SCSGSs and μ ATL-FO formulas. In this case, we can adopt the approach of [11], and base the verification method on two main ingredients: (i) *regression* [26], and (ii) *fixpoint approximates* and the classical Knaster and Tarski results [31].

Regarding regression, note that with *LegalM* defined as in Section 3, if φ is regressable then $\langle\langle G \rangle\rangle \circ \varphi$ is also regressable, and in fact its (one step) regression is:

$$\begin{aligned} \mathcal{R}(\langle\langle G \rangle\rangle \circ \varphi) &\doteq \\ &\exists m_{g_1}, \dots, m_{g_k} \cdot \bigwedge_{\{g_i, \dots, g_k\}=G} \text{LegalM}(g_i, m_{g_i}, \text{now}) \wedge \\ &\exists m_{g_{k+1}}, \dots, m_{g_n} \cdot \bigwedge_{\{g_{k+1}, \dots, g_n\}=\bar{G}} \text{LegalM}(g_i, m_{g_i}, \text{now}) \wedge \\ &\forall m_{g_{k+1}}, \dots, m_{g_n} \cdot \bigwedge_{\{g_{k+1}, \dots, g_n\}=\bar{G}} \text{LegalM}(g_i, m_{g_i}, \text{now}) \\ &\supset \mathcal{R}(\varphi(\text{do}(\text{tick}(m_{g_1}, \dots, m_{g_n}), \text{now}))) \end{aligned}$$

The second element is the ability, in some cases, to compute fixpoint approximates. Suppose that we want to verify a least fixpoint formula $\mu Z. \Psi(Z)$, where Z occurs free in Ψ . We can try to evaluate this formula using the general technique of iterated fixpoint approximates, which guarantees that for some transfinite ordinal we get the fixpoint [31]. The technique goes as follows. The approximates for a least fixpoint of the form $\mu Z. \Psi(Z)$ are as follows:

$$\begin{aligned} Z_0 &\doteq \Psi(\text{False}) \\ Z_1 &\doteq \Psi(Z_0) \\ Z_2 &\doteq \Psi(Z_1) \\ &\dots \end{aligned}$$

Observe that all of these formulas Z_i are situation suppressed which means that they all talk about the same situation, say *now*.

At limit transfinite ordinals ω we have that:

$$Z_\omega = \bigvee_i Z_i$$

Notice that in order to express this approximate we need infinitary disjunction (for least fixpoint as here, and conjunctions for greatest fixpoint).⁵ However, this technique becomes effective only when such a fixpoint can be reached within a finite number of iterations. For an in-depth discussion, see [11].

6 Golog-Based Players

We can also use programs to specify the possible behaviors of the agents playing the game. In particular, we can assume that each agent Ag_i is following a program δ_i specifying her possible moves at each step. For this, we use programs in a variant of the Golog programming language [19] where instead of atomic actions, we use *moves*. Such programs cannot be run in isolation; they must be executed concurrently with all agents moving synchronously. Programs constructs are the following:

m	atomic move
$\varphi?$	test for a condition
$\delta_1; \delta_2$	sequence
if φ then δ_1 else δ_2	conditional
while φ do δ	while loop
$\delta_1 \delta_2$	nondeterministic branch
$\pi x. \delta$	nondeterministic choice of argument
δ^*	nondeterministic iteration

⁵ By the way, notice that the fixpoint formulas in the μ -calculus are not continuous, so going above limit ordinals is in general necessary.

$$\begin{aligned}
\text{TransM}(m, s, \delta', m') &\equiv m' = m \wedge \delta' = \text{nil} \\
\text{TransM}(\varphi?, s, \delta', m') &\equiv \text{False} \\
\text{TransM}(\delta_1; \delta_2, s, \delta', m') &\equiv \\
&\quad \exists \delta'_1. \text{TransM}(\delta_1, s, \delta'_1, m') \wedge \delta' = \delta'_1; \delta_2 \vee \\
&\quad \text{FinalM}(\delta_1, s) \wedge \text{TransM}(\delta_2, s, \delta', m') \\
\text{TransM}(\text{if } \varphi \text{ then } \delta_1 \text{ else } \delta_2, s, \delta', m') &\equiv \\
&\quad \varphi[s] \wedge \text{TransM}(\delta_1, s, \delta', m') \vee \\
&\quad \neg\varphi[s] \wedge \text{TransM}(\delta_2, s, \delta', m') \\
\text{TransM}(\text{while } \varphi \text{ do } \delta, s, \delta', m') &\equiv \\
&\quad \varphi[s] \wedge \exists \delta''. \text{TransM}(\delta, s, \delta'', m') \wedge \delta' = \delta''; (\text{while } \varphi \text{ do } \delta) \\
\text{TransM}(\delta_1 | \delta_2, s, \delta', m') &\equiv \\
&\quad \text{TransM}(\delta_1, s, \delta', m') \vee \text{TransM}(\delta_2, s, \delta', m') \\
\text{TransM}(\pi x. \delta, s, \delta', m') &\equiv \exists z. \text{TransM}(\delta, s, \delta', m') \\
\text{TransM}(\delta^*, s, \delta', m') &\equiv \exists \delta''. \text{TransM}(\delta, s, \delta'', m') \wedge \delta' = \delta''; \delta^* \\
\text{TransM}(\text{nil}, s, \delta', m') &\equiv \text{False} \\
\text{FinalM}(m, s) &\equiv \text{False} \\
\text{FinalM}(\varphi?, s) &\equiv \varphi[s] \\
\text{FinalM}(\delta_1; \delta_2, s) &\equiv \text{FinalM}(\delta_1, s) \wedge \text{FinalM}(\delta_2, s) \\
\text{FinalM}(\text{if } \varphi \text{ then } \delta_1 \text{ else } \delta_2, s) &\equiv \\
&\quad \varphi[s] \wedge \text{FinalM}(\delta_1, s) \vee \neg\varphi[s] \wedge \text{FinalM}(\delta_2, s) \\
\text{FinalM}(\text{while } \varphi \text{ do } \delta, s) &\equiv \\
&\quad \varphi[s] \wedge \text{FinalM}(\delta, s) \vee \neg\varphi[s] \\
\text{FinalM}(\delta_1 | \delta_2, s) &\equiv \text{FinalM}(\delta_1, s) \vee \text{FinalM}(\delta_2, s) \\
\text{FinalM}(\pi x. \delta, s) &\equiv \exists x. \text{FinalM}(\delta, s) \\
\text{FinalM}(\delta^*, s) &\equiv \text{True} \\
\text{FinalM}(\text{nil}, s) &\equiv \text{True}
\end{aligned}$$

Figure 1. Axioms specifying *TransM* and *FinalM*

In the above, m is a term that represents a move, possibly with parameters, and φ is situation-suppressed SitCalc formula. Program $\delta_1 | \delta_2$ allows for the nondeterministic choice between programs δ_1 and δ_2 , while $\pi x. \delta$ executes program δ for *some* nondeterministic choice of a legal binding for variable x (observe that such a choice is, in general, unbounded). δ^* performs δ zero or more times. Note that we leave out recursive procedures.

To assign semantics to such programs we use notions analogous to *Trans* and *Final* from ConGolog's transition semantics [6]. In particular we introduce the predicate $\text{TransM}(\delta, s, \delta', m)$ to mean that the program δ in situation s can perform move m leaving δ' as the remaining program to execute, and the predicate $\text{FinalM}(\delta, s)$ to mean that program δ can be considered terminated in situation s . The definition of these predicates appears in Figure 1. We can read these axioms as follows: A program consisting of an atomic move m can only perform move m with the remaining program being the “empty” program nil . A test program $\varphi?$ can never perform a move. A sequence $\delta_1; \delta_2$ can perform move m in situation s if δ_1 can perform it with the remaining program being what remains of δ_1 followed by δ_2 or if δ_1 can be considered completed in s and δ_2 can perform move m in s with the remaining program being what remains of δ_2 . An **if** φ **then** δ_1 **else** δ_2 can perform move m in s if δ_1 can when φ is true, and if δ_2 can when φ is false; the remaining program is what remains of the selected branch. A “while” program can perform move m in s if its condition is true in s and its body can perform m ; the remaining program is what remains of the body followed by the “while” program itself. A nondeterministic branch can perform move m in s if either one of its branches can, the remaining program being what remains of the chosen branch. A nondeterministic choice of argument $\pi x. \delta$ can perform move m in s if δ can perform it for some value of variable x , which may occur in δ ; the remaining program is what remains of δ for this value. A nondeterministic iteration δ^* can perform move m in s if δ can perform it; the remaining program is what remains of δ followed by δ^* again. Finally, the empty program nil can never perform a move.

For *FinalM* we have the following: A program consisting of an atomic move m is never considered terminated. A test program $\varphi?$ is considered terminated in situation s if and only if φ holds in s . A sequence $\delta_1; \delta_2$ is considered terminated in situation s if both δ_1 and

δ_2 are terminated in s . An **if** φ **then** δ_1 **else** δ_2 is terminated if δ_1 is when φ is true, and if δ_2 is when φ is false. A “while” program is terminated if its condition is false or if its condition is true and its body is terminated. A nondeterministic branch is terminated if either one of its branches is. A nondeterministic choice of argument $\pi x. \delta$ is terminated if δ is terminated for some value of variable x , which may occur in δ . A nondeterministic iteration δ^* can always be considered terminated, as it can execute 0 times. Lastly, the empty program nil is always considered terminated.

Actually, we require that the agents' behavior programs be *move determined*.⁶ That is, we require that the step-by-step execution of such programs be fully determined at each step by the selected move. In other words, we cannot have nondeterminism in the program once the move is selected. E.g., program $m_1; (m_2 | m_3)$ is move determined, but $(m_1; m_2) | (m_1; m_3)$ is not; with the latter, the remaining program after performing m_1 could be either m_2 or m_3 . We impose this requirement because we use programs to specify *the* set of available moves for each agent in every game state. Using *TransM* we can formalize that a program δ is *move determined in a situation* s as:

$$\begin{aligned}
\text{MoveDet}(\delta, s) &\doteq \\
&\quad \forall m, \delta', \delta''. \text{TransM}(\delta, s, \delta', m) \wedge \text{TransM}(\delta, s, \delta'', m) \supset \delta' = \delta''
\end{aligned}$$

An agent Ag_i is *move determined in a game* if its (remaining) program is move determined in every situation that the game can reach.⁷

We can then define *LegalM* in terms of such programs by introducing a special fluent $\text{CurrProg}(Ag_i, \delta_i, s)$ that stores the remaining program of each agent in the situation:

$$\begin{aligned}
\text{LegalM}(Ag_i, m, s) &\doteq \\
&\quad \text{CurrProg}(Ag_i, \delta_i, s) \wedge \exists \delta'_i. \text{TransM}(\delta_i, s, \delta'_i, m)
\end{aligned}$$

where the successor state axiom for *CurrProg* is as follows:

$$\begin{aligned}
\text{CurrProg}(Ag_i, \delta'_i, \text{do}(\text{tick}(m_i, \dots, m_n), s)) &\equiv \\
&\quad \text{CurrProg}(Ag_i, \delta_i, s) \wedge \text{TransM}(\delta_i, s, \delta'_i, m_i)
\end{aligned}$$

⁶ The notion of move-determined program is similar to that of situation-determined program from [7].

⁷ Using *CurrProg* introduced here, this can be specified as follows:
 $\forall s. \forall \delta_i. \text{Executable}(s) \wedge \text{CurrProg}(Ag_i, \delta_i, s) \supset \text{MoveDet}(\delta_i, s)$.

That is, a move m is legal for agent Ag_i in situation s if her current remaining program δ_i in s can perform m , and when a joint move $tick(m_1, \dots, m_n)$ is performed, the current remaining program of each agent Ag_i is updated to be what remains of her current program after her move m_i .

Example 7 For the game of Example 1, we can define the legal moves of Ag_3 using the following program:

$$\begin{aligned} CurrProg(Ag_3, \delta, S_0) &\equiv \delta = BehaviorAg_3 \\ \text{where } BehaviorAg_3 &\doteq \\ \text{while } \neg Terminal &\text{ do} \\ &([\exists x, y. At(ag, x, y)?; Stay] | \\ & [AtExit(Ag_3)?; Exit] | \\ & [\pi d. \exists u, v, x, y. At(Ag_3, u, v, s) \wedge Adj(u, v, d, x, y)?; move(d)]) \end{aligned}$$

For the guard agents Ag_1 and Ag_2 , the program is similar, except that the *Exit* move is not allowed:

$$\begin{aligned} \text{while } \neg Terminal &\text{ do} \\ &([\exists x, y. At(ag, x, y)?; Stay] | \\ & [\pi d. \exists u, v, x, y. At(ag, u, v) \wedge Adj(u, v, d, x, y)?; move(d)]) \end{aligned}$$

We can also specify more constrained behaviors/strategies, e.g., one where Ag_3 never moves in a direction where he may be captured:

$$\begin{aligned} BehaviorAg_3 &\doteq \\ \text{while } \neg Terminal &\text{ do} \\ &([\exists x, y. At(Ag_3, x, y)?; Stay] | \\ & [AtExit(Ag_3)?; Exit] | \\ & [\pi d. \exists u, v, x, y. At(Ag_3, u, v) \wedge Adj(u, v, d, x, y) \wedge \\ & \quad \neg \exists m. Capturing(Ag_3, move(d), Ag_1, m) \wedge \\ & \quad \neg \exists m. Capturing(Ag_3, move(d), Ag_2, m)?; \\ & \quad move(d)]) \end{aligned}$$

When we do this, we verify properties of the system under the assumption that agents behave as specified. We should of course ensure that all moves allowed by such a specialized behavior are legal and that the agent always has some move it can make until the game ends. The latter is just playability, which we discussed earlier. The former requires establishing a simulation relation between the transition systems induced by the specialized and the original program, cf. [28]; we leave this for future work.

The verification techniques of Sec. 5 can be adapted to handle *LegalM* defined through programs.

Propositional case. The propositional case is straightforward.

Theorem 8 Let \mathcal{D} be a SCSGS with a finite object domain, with *LegalM* and *CurrProg* defined through programs. Then, for every $\mu\text{ATL-FO}$ formula Ψ , checking whether $\mathcal{D} \models \Psi$ is decidable.

Proof (sketch). Since the domain is bounded, the number of possible remaining programs within every computation is finite. Hence every fluent, including *LegalM* and *CurrProg* has a finite extension (and hence can be represented propositionally). Thus we can define a finite transition system that is bisimilar to the model of the action theory and check the property Ψ over it. \square

Bounded first-order case. For the bounded first-order case, we can leverage on recent work [10]. The difficulty when the domain is infinite is that the number of possible remaining programs within a computation is infinite in general. Moreover their inductive structure guides the definition of *TransM* and *FinalM* and hence ultimately of

LegalM and *CurrProg*. The results in [10] however, show that in the absence of recursion, the source of having infinitely many program terms is the pick operator π . Now, when the number of action types is finite (and in our framework, moves (and hence actions) come in finitely many move types), we can capture such programs as a pair formed by a program schema (with pick variables uninstantiated) and a separate set of variable substitutions, one for each pick variable in the original program, which in turn can be assumed to range over objects only w.l.o.g. In this way, the number of remaining program schemas that are generated during a computation is finite (they act as a program counter) while the number of possible substitutions is infinite (they can get all possible values from the infinite object domain). The point is that the number of pick variables in a program is syntactically determined by the original program alone (not the remaining programs that can be generated) and hence is naturally bounded.

As a final result [10] shows that for *situation-determined programs*, execution, as defined by *Trans* and *Final*, can be captured using new suitable predicate fluents, which are bounded. The same kind of reasoning can indeed be applied in our case to show that for *move-determined programs*, *TransM* and *FinalM* can be captured using new predicate fluents, which are bounded. This, in turn, makes the extension of *LegalM* and *CurrProg* bounded and hence we can apply Theorem 6 to get decidability. Thus we get:

Theorem 9 Let \mathcal{D} be a SCSGS that is a bounded action theory except for *LegalM* and *CurrProg* defined through programs as above. Then for every $\mu\text{ATL-FO}$ formula Ψ , checking whether $\mathcal{D} \models \Psi$ is decidable.

General first-order case. For the general first-order case, we can only obtain sound (but generally incomplete) methods, e.g., resorting to the techniques in [11] based on program characteristic graphs [5] or compilation techniques such as [12] and in [10] for situation-determined (in our case move-determined) programs.

7 Conclusion

In this paper, we have defined a logical framework, SCSGSs, for representing synchronous games-like systems and verifying temporal properties over them. We have also shown that under some common assumptions, GDL games can be represented as SCSGSs. Perhaps more significantly our framework allows for representing first-order GDL games in standard situation calculus, and thus allows one to leverage on the wide literature on such a formalism for analyzing the game, in particular for FO-temporal verification. Indeed, a key point is that our framework is truly first-order and can be used to specify games/systems that involve infinite domains and an infinite set of states. Further, when the SCSGS is “propositional” or “bounded”, verification of large classes of temporal formulas are decidable. Even in the general case, reasoners for our framework can be developed. A prototype verifier that uses the iterated fixpoint approximation technique is discussed in [17]. Such reasoners could be used to verify properties of interest in general game playing. One important assumption in our framework is that the game state is fully observable and no agent can any have private information. We would like to generalize it to accommodate partially observable game settings.

ACKNOWLEDGEMENTS

We acknowledge the support of Sapienza 2015 project “Immersive Cognitive Environments” and the National Science and Engineering Research Council of Canada.

REFERENCES

- [1] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman, 'Alternating-time temporal logic', *J. ACM*, **49**(5), 672–713, (2002).
- [2] Julien Bradfield and Colin Stirling, 'Modal mu-calculi', in *Handbook of Modal Logic*, volume 3, 721–756, Elsevier, (2007).
- [3] Diego Calvese, Giuseppe De Giacomo, Marco Montali, and Fabio Patrizi, 'On first-order μ -calculus over situation calculus action theories', in *Proc. of KR*, (2016).
- [4] Diego Calvese, Giuseppe De Giacomo, Marco Montali, and Fabio Patrizi, 'First-order μ -calculus over generic transition systems and applications to the situation calculus'. Submitted.
- [5] Jens Claßen and Gerhard Lakemeyer, 'A logic for non-terminating Golog programs', in *Proc. of KR*, pp. 589–599, (2008).
- [6] Giuseppe De Giacomo, Yves Lespérance, and Hector J. Levesque, 'ConGolog, a concurrent programming language based on the situation calculus', *Artificial Intelligence*, **121**(1–2), 109–169, (2000).
- [7] Giuseppe De Giacomo, Yves Lespérance, and Christian J. Muise, 'On supervising agents in situation-determined ConGolog', in *Proc. of AAMAS*, pp. 1031–1038, (2012).
- [8] Giuseppe De Giacomo, Yves Lespérance, and Fabio Patrizi, 'Bounded situation calculus action theories and decidable verification', in *Proc. of KR*, (2012).
- [9] Giuseppe De Giacomo, Yves Lespérance, and Fabio Patrizi, 'Bounded situation calculus action theories', *Artificial Intelligence*, **237**, 172–203, (2016).
- [10] Giuseppe De Giacomo, Yves Lespérance, Fabio Patrizi, and Sebastian Sardina, 'Verifying ConGolog programs on bounded situation calculus theories', in *Proc. of AAI*, (2016).
- [11] Giuseppe De Giacomo, Yves Lespérance, and Adrian R Pearce, 'Situation calculus based programs for representing and reasoning about game structures.', in *Proc. of KR*, (2010).
- [12] Christian Fritz, Jorge A. Baier, and Sheila A. McIlraith, 'ConGolog, Sin Trans: Compiling ConGolog into basic action theories for planning and beyond', in *Proc. of KR*, pp. 600–610, (2008).
- [13] Michael R. Genesereth, Nathaniel Love, and Barney Pell, 'General game playing: Overview of the AAI competition', *AI Magazine*, **26**(2), 62–72, (2005).
- [14] Enrico Giunchiglia, Joohyung Lee, Vladimir Lifschitz, Norman McCain, and Hudson Turner, 'Nonmonotonic causal theories', *Artificial Intelligence*, **153**(1–2), 49–104, (2004).
- [15] Babak Bagheri Hariri, Diego Calvese, Giuseppe De Giacomo, Alin Deutsch, and Marco Montali, 'Verification of relational data-centric dynamic systems with external services', in *Proc. of PODS*, pp. 163–174, (2013).
- [16] Andreas Herzig, Emiliano Lorini, Frédéric Moisan, and Nicolas Troquard, 'A dynamic logic of normative systems', in *Proc. of IJCAI*, (2011).
- [17] Slawomir Kmiec and Yves Lespérance, 'Infinite states verification in game-theoretic logics: Case studies and implementation', in *Proc. of EMAS*. Springer, (2014).
- [18] Hector J. Levesque and Gerhard Lakemeyer, *The Logic of Knowledge Bases*, MIT Press, 2001.
- [19] Hector J. Levesque, Ray Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl, 'GOLOG: A logic programming language for dynamic domains', *J. Logic Programming*, **31**, 59–84, (1997).
- [20] John W. Lloyd, *Foundations of Logic Programming, 2nd Edition*, Springer, 1987.
- [21] Alessio Lomuscio, Hongyang Qu, and Franco Raimondi, 'MCMAS: A model checker for the verification of multi-agent systems', in *Proc. of CAV*, pp. 682–688, (2009).
- [22] Nathaniel Love, Timothy Hinrichs, David Haley, Eric Schkufza, and Michael Genesereth, 'General game playing: Game description language specification'. Tech. Rept. LG-2006-01, Stanford University, 2006.
- [23] J. McCarthy and P. J. Hayes, 'Some Philosophical Problems From the Standpoint of Artificial Intelligence', *Machine Intelligence*, **4**, 463–502, (1969).
- [24] Fabio Mogavero, Aniello Murano, and Moshe Y. Vardi, 'Reasoning about strategies', in *Proc. of FSTTCS*, pp. 133–144, (2010).
- [25] Javier Pinto, 'Concurrent actions and interacting effects', in *Proc. of KR*, pp. 292–303, (1998).
- [26] Ray Reiter, *Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems*, MIT Press, 2001.
- [27] Ji Ruan, Wiebe van der Hoek, and Michael Wooldridge, 'Verification of games in the game description language', *J. Log. Comput.*, **19**(6), 1127–1156, (2009).
- [28] Sebastian Sardina and Giuseppe De Giacomo, 'Composition of ConGolog programs', in *Proc. of IJCAI*, pp. 904–910, (2009).
- [29] Stephan Schiffel and Michael Thielscher, 'A multiagent semantics for the game description language', in *Agents and Artificial Intelligence*, volume 67 of *CCIS*, pp. 44–55. Springer, (2010).
- [30] Stephan Schiffel and Michael Thielscher, 'Representing and reasoning about the rules of general games with imperfect information', *J. Artif. Intell. Res. (JAIR)*, **49**, 171–206, (2014).
- [31] Alfred Tarski, 'A lattice-theoretical fixpoint theorem and its applications', *Pacific J. of Mathematics*, **5**(2), 285–309, (1955).
- [32] Eugenia Ternovskaia, 'Automata theory for reasoning about actions', in *Proc. of IJCAI*, pp. 153–159, (1999).
- [33] Michael Thielscher, 'A general game description language for incomplete information games', in *Proc. of AAI*, (2010).
- [34] Michael Thielscher, 'Translating general game descriptions into an action language', in *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning*, volume 6565 of *LNCS*, pp. 300–314. Springer, (2011).
- [35] Wiebe van der Hoek and Michael Wooldridge, 'On the logic of cooperation and propositional control', *Artif. Intell.*, **164**(1-2), 81–119, (2005).

The Game of Reciprocation Habits

Gleb Polevoy and Mathijs de Weerd and Catholijn Jonker¹

Abstract. People often have reciprocal habits, almost automatically responding to others' actions. A robot who interacts with humans may also reciprocate, in order to come across natural and be predictable. We aim to facilitate decision support that advises on utility-efficient habits in these interactions. To this end, given a model for reciprocation behavior with parameters that represent habits, we define a game that describes what habit one should adopt to increase the utility of the process. This paper concentrates on two agents. The used model defines that an agent's action is a weighted combination of the other's previous actions (reacting) and either i) her innate kindness, or ii) her own previous action (inertia). In order to analyze what happens when everyone reciprocates rationally, we define a game where an agent may choose her habit, which is either her reciprocation attitude (i or ii), or both her reciprocation attitude and weight. We characterize the Nash equilibria of these games and consider their efficiency. We find that the less kind agents should adjust to the kinder agents to improve both their own utility as well as the social welfare. This constitutes advice on improving cooperation and explains real life phenomena in human interaction, such as the societal benefits from adopting the behavior of the kindest person, or becoming more polite as one grows up.

1 Introduction

Interaction is central in human behavior, e.g., at school, in file sharing over networks, and in business cooperation. While interacting, people tend to reciprocate, i.e., react on the past actions of others [9, 11, 14]. Imagine software agents owned by individuals repeatedly competing with the same people online. People expect reciprocal behavior and tend to behave so themselves. Virtual assistants also need to be reciprocal in order to be credible. Countries at an arms race or arguing friends also tend to be nicer if the other side is nicer [7, 27, 13]. In these and other cases of repeated interaction, we can help people and artificial agents obtain more from the interaction by providing decision support. The decision is how to reciprocate. Reciprocating efficiently includes defining to one's software agent or other artificial agents how to reciprocate with humans. In order to help people strategically choose efficient approaches for reciprocating, and to predict that strategic choice of how to reciprocate, a model is needed that is amenable to analytical analysis and has enough predictive power.

Consider the following example of an arms race.

Example 1 Consider n countries $1, 2, \dots, n$; each country can put a certain arsenal of weapons at the border with its

neighbors. What a country approximately does with respect to another country at a given year is what was done in the previous year, adjusted to react to what the other countries did. If they armed themselves against us, we also will, and if the others aimed at us less, so shall we. This process is often reciprocal with linear reactions [7, 27]. Perhaps, one reason for that is that politicians can explain a reciprocal action as a proper reaction to the nation. A crucial question is how to make this process efficient, so that one's country, and, preferably, everyone incurs the least possible cost.

In this example, an action had a negative influence on the other country. We can also consider a positive influence on the other side in this context; for instance, a concession.

Software agents can reciprocate automatically.

Example 2 Consider software agents running on computers in a cloud. They need to agree on how much resources each is allocated. Since their owners may want to be nice to others reciprocally, it is reasonable to make them reciprocate. Everyone wants her agent to reciprocate as efficiently as possible, and also the society can save much money by efficient reciprocation.

Companies can reciprocate while achieving mutual gain.

Example 3 Reciprocation is useful in business life [25]. Reciprocating means helping the other, for example, by redirecting potential clients to another company. It is definitely economically important to make this reciprocation efficient.

The existing studies of reciprocation (sometimes repeated) either attempt to explain why reciprocation is there in the first place [4, 3, 26, 10], or, given that reciprocation exists, they analyze what happens in a short interaction where being reciprocal pays off [5, 9, 23]. We, on the other hand, consider a lengthy interaction, that is (naturally) bound to be reciprocal, but changing the approach of reciprocation is possible, in order to receive more and do less.

To study such interactions, we employ the model from Polevoy, de Weerd and Jonker [22]², which formally defined and analyzed repeated intrinsic reciprocation, to understand how reciprocity makes interaction evolve with time. We briefly summarize the model. Actions, which are influences of an agent on another one, are represented by *weight*, where a higher value means a more desirable contribution to its recipient. That model was mainly inspired by arms race models [7, 27] and a model of spouses arguments [13]. Given the model, the paper [22] analyzes the interaction it engenders.

¹ Delft University of Technology, email: g.polevoy@tudelft.nl

² The full version can be found at <http://arxiv.org/abs/1601.07965>.

This model consists of two reciprocation attitudes, where the action of an agent is a convex combination³ between i) one's own kindness or ii) one's own last action (mental inertia), and the other's last action (reaction). The combination is determined by the agent's reciprocation coefficient. Since the last own action is, recursively, a product of previous actions, it represents the agent at a given time, including her history. Attitude i), which is connected to kindness, is called *fixed*, and ii) depending on one's own last action is called *floating*.

A reciprocation process converges, and in many cases, the actions in the limit are found in [22]. We aim to provide decision support and predict the strategic reciprocation. A natural question to ask here is in what way the agents can strategically influence the reciprocation process for their own good, and what the social welfare will become when every individual behaves strategically. Setting one's way of reciprocating resembles Mastenbroek's [17, Chapter 14] recommendation to know one's own negotiating style and adjust it. Assuming people strategically choose each action is unrealistic, since people usually act on habits [15], and a strategic choice consists of choosing a habit for the reciprocal interaction. Here, the habit, chosen after deliberation, can be the balance between reacting and being faithful to oneself, as defined in the model. It is also easy to prescribe a "habit" to a robot.

Choosing habits resembles bounded rationality, especially that of procedures of choice [24, Chapter 2]. Indeed, our agent follows the procedure of rationally choosing among the possible habits. The difference is that choosing a habit does include a rational step, and is, therefore, amenable to a standard game-theoretic analysis, like NE and price of anarchy and stability. Choosing habits resembles metagames as well, when an agent chooses a representative to play the underlying game for her. For instance, Rubinstein [24, Chapter 8] and [21, Chapter 9] define a *machine game*, where an agent wants a well-paying strategy that is simple to implement. This trade-off is modeled by choosing a finite deterministic automaton to play the repeated game, where the agent's utility increases in the utility of the underlying game and decreases in the number of the states of the chosen automaton. The equilibria in this game are found for the case of the utility of the repeated game being defined as the limit-of-means or with discounting in [6]. A player in a machine game chooses a finite automaton, while our player chooses a habit. Choosing an automaton, however, considers the bounding effect of finiteness and attempt to minimize the automaton's state space, while we simply consider a best possible habit, all habits being equally simple. Therefore, our model neither generalizes theirs nor is our model generalized by theirs. Additionally, no finite automaton is able to model reciprocation, though it is possible to approximate it arbitrarily.

To model strategically setting one's habits, we define the utility of an agent and then we consider the one-shot game of setting one's own reciprocation attitude or coefficient, each of which represents a habit. We analyze changing reciprocation attitude for a pairwise interaction. Pairwise interactions still allow for many agents provided assuming that the agents do not mix one relationship with the other ones.

All the agents choose their reciprocation habits and then the reciprocation process plays itself. Our contributions in-

clude a characterization of this game's Nash equilibria (NE) and a discussion of their efficiencies. We consider only pure NE in this paper. Analyzing this game provides an insight into how people and machines could change their behavior to achieve a more desirable behavior in the limit of the interaction process. This desirability can be to themselves or to the society. In addition to predicting the strategic reciprocation and advising on what to do, the analysis explains the following known phenomena. First, in reciprocation, we often notice that when the example of the kindest person is followed by others, it makes the group more successful [2]. We also notice, that people tend to become more polite as they grow up [12], which is yet another example of the utility of learning from the behavior of the kindest.

We present the model in Section 2. To make this paper self-contained, Section 2.3 provides the necessary background. We consider the game of choosing the reciprocation attitude in sections 3, 4 and 5, proving the central Theorems 4 and 5. We also model in Section 6 what happens if an agent can choose both own attitude and reciprocation coefficient. The answers are given in the key Theorems 6 and 7.

We briefly describe the model and the results for n agents in Section 7. We deal with convergence of the best response dynamics to a NE in Section 8 and conclude in Section 9.

2 Modeling Reciprocation

We first model agents, times and actions. We conclude the section by sharpening the model and providing explanatory examples. Let $N = \{1, 2\}$ be $n = 2$ interacting agents. Time is modeled by a set of discrete moments $t \in T \triangleq \{0, 1, 2, \dots\}^4$, defining a time slot when the agents act.

Denote the weight of an action by agent $i \in N$ on another agent $j \in N$ at moment $t \in T$ by $x_{i,j} : T \rightarrow \mathbb{R}$. For example, when interacting by file sharing, the actions of sending a valid piece of a file, nothing, or a piece with a virus are decreasing in weight. Since only the weight of an action is relevant, we usually write "action" while referring to its weight.

We now define two reciprocation attitudes, which define how an agent reciprocates. We need the following notions. The kindness of agent i is denoted by $k_i \in \mathbb{R}$; w.l.o.g., $k_2 \geq k_1$ throughout the paper. Kindness models inherent inclination to help others; in particular, it determines the first action of an agent, before others have acted. We model agent i 's inclination to mimic the other agent's action by reciprocation coefficients $r_i \in [0, 1]$. Here, r_i is the fraction of $x_{i,j}(t)$ that is determined by the last action of j upon i . Conceptually, reacting to last actions, one reacts to the actor, since "who you are is what you do" [18].

Intuitively, with the *fixed* attitude, actions always depend on the agent's kindness, while the *floating* attitude moves freely in the reciprocation process, and kindness directly influences such behavior only at $t = 0$. In both cases $x_{i,j}(0) \triangleq k_i$.

Definition 1 For the fixed reciprocation attitude, agent i 's action on another agent j is determined by j 's last action weighted by r_i and by the agent's kindness weighted by $1 - r_i$.

³ A combination is convex if it has nonnegative weights that sum up to 1.

⁴ Allowing agents to be non-synchronized is possible, but we assume synchronicity for the sake of clarity.

That is, for $t \in T$,

$$x_{i,j}(t) \triangleq (1 - r_i) \cdot k_i + r_i \cdot x_{j,i}(t - 1).$$

Definition 2 In the floating reciprocation attitude, agent i 's action is a weighted average of that of the other agent j , and of her own last action. To be precise, for $t \in T$,

$$x_{i,j}(t) \triangleq (1 - r_i) \cdot x_{i,j}(t - 1) + r_i \cdot x_{j,i}(t - 1).$$

The relations are (usually inhomogeneous) linear recurrences with constant coefficients, but many variables. We could express the dependence of $x_{i,j}(t)$ only on $x_{i,j}(t')$ with $t' < t$, but then the coefficients would not be constant, besides the case of two *fixed* agents. We are not aware of a method to use the general recurrence theory to improve our results.

2.1 Context and Examples

Compared to the other reciprocation models, our model takes reciprocal actions as given and looks at the process, while other models either consider how reciprocation originates, such as the evolutionary model of Axelrod [4], or take it as given and consider specific games, such as in [5, 8, 9, 23].

In Example 1, let the reciprocation coefficients be $r_1 = 0.2, r_2 = 0.7$. Assume the kindness to be $k_1 = 0$ and $k_2 = 0.5$. At $t = 0$, every country's action on every other country is equal to her kindness value, so $x_{i,2} = 0$ and $x_{2,1} = 0.5$. If all countries rely on their previous action, meaning that they are *floating*, then, at $t = 1$ they act as follows: $x_{1,2}(1) = (1 - 0.2) \cdot 0 + 0.2 \cdot 0.5 = 0.1$, $x_{2,1}(1) = (1 - 0.7) \cdot 0.5 + 0.7 \cdot 0 = 0.15$. Theorem 2 implies they converge to the common limit $\frac{0.7}{0.2+0.7} \cdot 0 + \frac{0.2}{0.2+0.7} \cdot 0.5 = 1/9$. This is closer to k_1 than to k_2 , since country 1 is less responsive, in the sense that $r_1 < r_2$.

Consider modeling tit for tat [3]:

Example 4 In our model, a tit for tat agent with two options: cooperate or defect is easily modeled with $r_i = 1, k_i = 1$, meaning that the original action is cooperating (1) and the next action is the current action of the other agent. If one of two tit-for-tat agents makes a mistake and begins with defection ($k_2 = 0$), then they will alternate.

If the agents are human, this example predicts an indefinitely long alternation, which seems unrealistic to us. Similarly, an agent that sticks to his actions regardless the other seems highly implausible. This provides evidence that extreme values of the reciprocation coefficients are uncommon in life.

2.2 Utility Definition

An agent's utility at a given time moment is the action one receives minus the effort incurred by the action one performs. Colloquially, this is what the agent gets minus what she gives. This classical way of defining utility is expressed, for instance, in the quasilinear preferences of auction theory [20, Chapter 9.3]. Formally, we define as follows.

Definition 3 The utility of agent i at moment t , $u_{i,t}: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, is defined as

$$u_{i,t}(x_{i,j}(t), x_{j,i}(t)) \triangleq x_{j,i}(t) - \beta_i x_{i,j}(t),$$

where β_i is the relative importance of the effort incurred by performed actions for i 's utility. The personal price of acting is higher, equal or lower than of receiving an action, if β is bigger, equal or smaller than 1, respectively.

Denote $x(t) \triangleq x_{1,2}(t)$ and $y(t) \triangleq x_{2,1}(t)$. Thus, at time t , agent 1's utility is $y(t) - \beta_1 x(t)$ and 2's utility is $x(t) - \beta_2 y(t)$.

We take acting with a minus sign, to account for the effort it takes (a negative β_i would mean that the agent enjoys making effort). According to this formula, when $\beta_i > 0$, a negative action would suddenly contribute to the utility; we needed to take the absolute value. Instead, we will assume that actions are always non-negative, which is equivalent to all kindness values being non-negative. We still can have negative influence, we have simply mathematically transformed all the original kindness values by adding a sufficiently large number so that they all have become nonnegative.

To model the utility in the long run, we give the following Define the asymptotic utility, or just the *utility*, of agent i , $u_i: \mathbb{R}^\infty \times \mathbb{R}^\infty \rightarrow \mathbb{R}$, as $u_i(\bigcup_{t=0}^\infty \{x_{i,j}(t), x_{j,i}(t)\}) \triangleq \lim_{t \rightarrow \infty} u_{i,t}(x_{i,j}(t), x_{j,i}(t))$. When the parameters in the parentheses are clear from the context, we may omit them.

This is the utility we consider in the paper. The utility might be defined otherwise, like a discounted sum, though since we have an exponential convergence, it is possible to simplify it to looking at the limit, assuming that the discounting is not extremely quick. It can be proven that our definition is also equivalent to the other models from Osborne and Rubinstein [21, Chapter 8.3], which are limit of arithmetic means and overtaking. We omit this for lack of space.

2.3 Background

In order to analyze utility in the long run, we use the following convergence theorems from [22], representing what takes place once the actions have stabilized. For two *fixed* agents, they prove:

Theorem 1 If the reciprocation coefficients are not both 1, which means $r_1 r_2 < 1$, then we have, for $i \in N$: $\lim_{t \rightarrow \infty} x_{i,j}(t) = \frac{(1-r_i)k_i + r_i(1-r_j)k_j}{1-r_i r_j}$.

For two agents, in the *floating* case, they show:

Theorem 2 If the reciprocation coefficients are neither both 0 and nor both 1, which means $0 < r_1 + r_2 < 2$, then, as $t \rightarrow \infty$, $x(t)$ and $y(t)$ converge to a common limit, which is

$$\frac{1}{2} \left(k_1 + k_2 + (k_2 - k_1) \frac{r_1 - r_2}{r_1 + r_2} \right) = \frac{r_2}{r_1 + r_2} k_1 + \frac{r_1}{r_1 + r_2} k_2.$$

For a *fixed* and a *floating* agent, the following holds:

Theorem 3 If agent i employs fixed reciprocation and the other agent j employs the floating one, assume that $r_i < 1$ and $r_j > 0$. Then, both limits exist and are equal to k_i . The convergence is geometrically fast.

The following holds for two agents with any attitudes:

Proposition 1 If $k_1 \leq k_2$ and both action sequences converge, then $\lim_{t \rightarrow \infty} x_{i,j}(t) \leq \lim_{t \rightarrow \infty} x_{j,i}(t)$.

3 Utility Maximization

As a first step to analyzing strategic choices, consider how an agent can maximize her utility by choosing either her reciprocation coefficient or reciprocation attitude, before the interaction begins. This can be expected from a rational agent, who reciprocates, but chooses her reciprocation habits. In the case of Example 1, this models a country setting a smart foreign policy with respect to arming. Since in reality the behavioral parameters of others are unknown, choosing an optimal behavior will probably be harder, through trial and error, and the theory predicts the trend of these choices. Some (parts of) proofs are omitted for lack of space.

First, suppose that the only available option of agent i to modify the reciprocation process is by setting its reciprocation coefficient r_i . We therefore analyze how i 's utility depends on r_i . In the results of this section, the asymmetry of the agents stems from $k_2 \geq k_1$.

For the *fixed* reciprocation attitude, we prove:

Proposition 2 *In the fixed reciprocation attitude, the following holds: If $r_2 < 1$ and agent 1 wants to maximize his utility by choosing his reciprocation coefficient r_1 , then he should*

$$\text{set } r_1 \text{ to be } \begin{cases} 1 & \text{if } r_2 > \beta_1, \\ \text{anything} & r_2 = \beta_1, \\ 0 & r_2 < \beta_1. \end{cases}$$

If $r_1 < 1$ and agent 2 wants to maximize his utility by choosing his reciprocation coefficient r_2 , then he should set r_2

$$\text{to be } \begin{cases} 0 & \text{if } r_1 > \beta_2, \\ \text{anything} & r_1 = \beta_2, \\ 1 & r_1 < \beta_2. \end{cases}$$

These choices are the only utility maximizing ones.

The idea of the proof is to express the utility of an agent and differentiate it by her reciprocation coefficient, to find candidates for the extrema.

Proof. Let us prove for agent 1 choosing r_1 . We first express 1's utility and then maximize it. Since $r_2 < 1$, we have $r_1 r_2 < 1$, and from Theorem 1,

$$\begin{aligned} \lim_{t \rightarrow \infty} x(t) &= \frac{(1-r_1)k_1 + r_1(1-r_2)k_2}{1-r_1 r_2}, \\ \lim_{t \rightarrow \infty} y(t) &= \frac{(1-r_2)k_2 + r_2(1-r_1)k_1}{1-r_1 r_2} \Rightarrow u_1 = \\ &= \frac{(1-r_2)k_2 + r_2(1-r_1)k_1}{1-r_1 r_2} - \beta_1 \frac{(1-r_1)k_1 + r_1(1-r_2)k_2}{1-r_1 r_2}. \end{aligned}$$

To find a maximum point of this utility as a function of r_1 , we differentiate:

$$\frac{\partial(u_1)}{\partial(r_1)} = \dots = \frac{(r_2 - \beta_1)(1-r_2)}{(1-r_1 r_2)^2} (k_2 - k_1).$$

Therefore, if $r_2 = \beta_1$, then the derivative is zero, and the utility is constant. Otherwise, the maximum is attained at an endpoint: at the right endpoint, if the $r_2 > \beta_1$, and at the left endpoint if $r_2 < \beta_1$.

The case of agent 2 choosing r_2 is proven by analogy. \square

For the *floating* reciprocation attitude, we prove:

Proposition 3 *In the floating reciprocation attitude, the following holds: If $r_2 < 1$ and agent 1 wants to maximize his utility by choosing his reciprocation coefficient r_1 , then he should*

$$\text{set } r_1 \text{ to be } \begin{cases} 1 & \text{if } r_2 > 0 \text{ and } \beta_1 < 1, \\ 0 & \text{if } r_2 > 0 \text{ and } \beta_1 > 1, \\ \text{anything} & \text{if } r_2 > 0 \text{ and } \beta_1 = 1, \\ 0 & \text{if } r_2 = 0 \text{ and } \beta_1 > 0, \\ \text{anything positive} & \text{if } r_2 = 0 \text{ and } \beta_1 < 0, \\ \text{anything} & \text{if } r_2 = 0 \text{ and } \beta_1 = 0. \end{cases}$$

If $r_1 < 1$ and agent 2 wants to maximize his utility by choosing his reciprocation coefficient r_2 , then he should set r_2

$$\text{to be } \begin{cases} 0 & \text{if } r_1 > 0 \text{ and } \beta_2 < 1, \\ 1 & \text{if } r_1 > 0 \text{ and } \beta_2 > 1, \\ \text{anything} & \text{if } r_1 > 0 \text{ and } \beta_2 = 1, \\ \text{anything positive} & \text{if } r_1 = 0 \text{ and } \beta_2 > 0, \\ 0 & \text{if } r_1 = 0 \text{ and } \beta_2 < 0, \\ \text{anything} & \text{if } r_1 = 0 \text{ and } \beta_2 = 0. \end{cases}$$

These choices are the only utility maximizing ones.

The idea of the proof is as in the previous proof.

If the kindness values and reciprocation coefficient are set, and an agent may only choose between *fixed* or *floating* reciprocation, we prove:

Proposition 4 *If $0 < r_1, r_2 < 1$, then, if agent 1 wants to maximize her utility, and she may only choose whether to employ fixed or floating reciprocation, then she should choose*

$$\begin{cases} \text{fixed} & \text{if } (2 \text{ is fixed} \wedge \{\beta_1 \geq r_2\}) \\ \vee (\text{agent 2 is floating} \wedge \{\beta_1 \geq 1\}), \\ \text{floating} & \text{if } (2 \text{ is fixed} \wedge \{\beta_1 \leq r_2\}) \\ \vee (\text{agent 2 is floating} \wedge \{\beta_1 \leq 1\}). \end{cases}$$

If agent 2 wants to maximize his utility by choosing fixed or floating reciprocation, then he should choose

$$\begin{cases} \text{floating} & \text{if } (1 \text{ is fixed} \wedge \{\beta_2 \geq r_1\}) \\ \vee (\text{agent 1 is floating} \wedge \{\beta_2 \geq 1\}), \\ \text{fixed} & \text{if } (1 \text{ is fixed} \wedge \{\beta_2 \leq r_1\}) \\ \vee (\text{agent 1 is floating} \wedge \{\beta_2 \leq 1\}). \end{cases}$$

Supposing $k_1 < k_2$, an attitude choice given in this proposition is the only best one if and only if the relevant inequality on the right-hand side of the conditions holds strictly.

The idea of the proof is to compare the possibilities, to see when which option is best. For $\beta_1 = \beta_2 = 0$, which is when both agents want only to receive more, all the results from this section are intuitive, since a less kind agent should choose to be very reciprocating, while the other agent should choose to be completely non-reciprocating, thereby remaining kind and pulling the other agent to act more.

In Example 1, if countries 1 and 2 have $r_1 = r_2 = 0.5$, $\beta_1 = 0, \beta_2 = 0.2$ (acting is cheap), then, whatever attitude 2 employs, 1 should employ *floating*, to maximize its utility.

We have prepared the analysis of the game of choosing reciprocation habits. To prepare the ground for analyzing the efficiency of NE, our next step will be finding how the social welfare can be maximized.

4 Maximizing Social Welfare

Maximizing the social welfare is relevant for analyzing the whole interaction of agents maximizing their own utilities as a game, to see how good equilibria are for the society relatively to the best possible social welfare. Regardless of the game, the manager (say, the boss of a group of interacting workers) wants to maximize the social welfare by influencing agents' behavior through propaganda or an incentive mechanism.

We now define the social welfare.

Definition 4 *The social welfare at time t ($SW_t: \mathbb{R}^2 \rightarrow \mathbb{R}$) is defined as the sum of utilities at time t , i.e.,*

$$SW_t \triangleq u_{1,t} + u_{2,t} = (1 - \beta_1)x(t) + (1 - \beta_2)y(t). \quad (1)$$

For the whole process, we define the (asymptotic) *social welfare*, $SW: (\mathbb{R}^2)^\infty \rightarrow \mathbb{R}$, as $SW \triangleq \lim_{t \rightarrow \infty} SW_t$.

In Example 1, changing the behavioral parameters to increase the social welfare models the United Nations trying to spread good practices among countries.

We first suppose that the only available option to influence the interaction network is through choosing the reciprocation coefficients of the agents, and ask what is the most efficient setup of the r_1, r_2 parameters. To this end, we now analyze how the asymptotic social welfare depends on these parameters. Recall that $k_2 \geq k_1$. For given reciprocation attitudes (not necessarily the same attitudes for both agents), we prove

Proposition 5 *We can maximize the social welfare by setting r_1 and r_2 to*

$$\begin{cases} r_1 = 1, r_2 = 0 & \text{if } \max\{\beta_1, \beta_2\} \leq 1, \\ r_1 = 0, r_2 = 1 & \text{if } \min\{\beta_1, \beta_2\} \geq 1, \\ r_1 = r_2 = 0 & \text{if } \beta_1 \geq 1, \beta_2 \leq 1, \\ r_1 = 1, r_2 = 0 & \text{if } \beta_1 \leq 1, \beta_2 \geq 1, \beta_1 + \beta_2 \leq 2, \\ r_1 = 0, r_2 = 1 & \text{if } \beta_1 \leq 1, \beta_2 \geq 1, \beta_1 + \beta_2 \geq 2. \end{cases} \quad (2)$$

The idea of the proof is to consider, what limits should be maximized, to maximize the social welfare.

Proof. If $\max\{\beta_1, \beta_2\} \leq 1$, then if we maximize both $\lim_{t \rightarrow \infty} x(t)$ and $\lim_{t \rightarrow \infty} y(t)$, we maximize the social welfare. For $r_1 = 1, r_2 = 0$, we obtain⁵ $\lim_{t \rightarrow \infty} x(t) = k_2$ and $\lim_{t \rightarrow \infty} y(t) = k_2$, which are the maximum possible. Thus, $r_1 = 1, r_2 = 0$ maximizes the social welfare.

We skip the easy cases, concentrating on the hard one.

If $\beta_1 \leq 1, \beta_2 \geq 1$, we first express the social welfare in a handier form, and subsequently show how we can maximize it. Denote $\delta \triangleq 1 - \beta_1 \Rightarrow \delta \geq 0$ and $\epsilon \triangleq 2 - \beta_1 - \beta_2$. Then, we have $1 - \beta_2 = -(\delta - \epsilon)$ and $SW = (1 - \beta_1) \lim_{t \rightarrow \infty} x(t) + (1 - \beta_2) \lim_{t \rightarrow \infty} y(t) = \delta \lim_{t \rightarrow \infty} x(t) - (\delta - \epsilon) \lim_{t \rightarrow \infty} y(t) = \delta(\lim_{t \rightarrow \infty} x(t) - \lim_{t \rightarrow \infty} y(t)) + \epsilon \lim_{t \rightarrow \infty} y(t)$.

Now, if $\beta_1 + \beta_2 \leq 2$, then $\epsilon \geq 0$ and thus, if we maximize $\lim_{t \rightarrow \infty} x(t) - \lim_{t \rightarrow \infty} y(t)$ and $\lim_{t \rightarrow \infty} y(t)$, we maximize the social welfare. For $r_1 = 1, r_2 = 0$, we obtain⁵ $\lim_{t \rightarrow \infty} x(t) = \lim_{t \rightarrow \infty} y(t) = k_2$, thus maximizing the first (since by Proposition 1, $\lim_{t \rightarrow \infty} x_{i,j}(t) \leq \lim_{t \rightarrow \infty} x_{j,i}(t)$, the first is non-positive) and the second. Thus, $r_1 = 1, r_2 = 0$ maximizes the social welfare.

⁵ This is evident from the definition of *fixed* or *floating* reciprocation, without a convergence theorem.

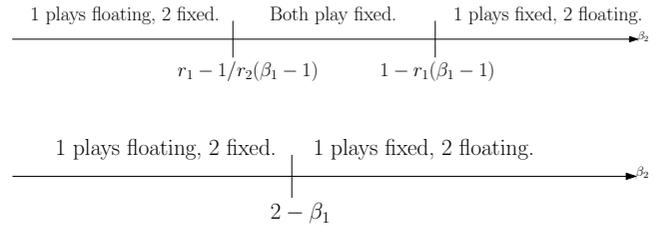


Figure 1: The upper figure is for $\beta_1 - 1 \geq 0$, and the lower figure is for $\beta_1 - 1 < 0$. The strategy profile written above denotes a profile to maximize the social welfare, based on where the value of β_2 resides.

Now, if $\beta_1 + \beta_2 \geq 2$, then $\epsilon \leq 0$ and thus, if we maximize $\lim_{t \rightarrow \infty} x(t) - \lim_{t \rightarrow \infty} y(t)$ and minimize $\lim_{t \rightarrow \infty} y(t)$, we maximize the social welfare. For $r_1 = 0, r_2 = 1$, we obtain⁵ $\lim_{t \rightarrow \infty} x(t) = \lim_{t \rightarrow \infty} y(t) = k_1$, thus maximizing the first and minimizing the second. Thus, $r_1 = 0, r_2 = 1$ maximizes the social welfare. \square

Note that this proposition holds also if we may influence both r_1, r_2 and the attitudes of the agents, since the proof maximizes and minimizes expressions for any possible attitudes.

Suppose now that the reciprocation coefficients are set, and the manager only chooses whether the agents employ *fixed* or *floating* reciprocation.

Proposition 6 *If $0 < r_1, r_2 < 1$, then the social welfare is maximal by reciprocating as follows:*

$$\begin{cases} 1 \text{ floating, } 2 \text{ fixed.} & \text{if } \beta_2 \leq 1 - \max\left\{\frac{1}{r_2}(\beta_1 - 1), \beta_1 - 1\right\}, \\ 1 \text{ fixed, } 2 \text{ fixed.} & \text{if } 1 - \frac{1}{r_2}(\beta_1 - 1) \leq \beta_2 \leq 1 - r_1(\beta_1 - 1), \\ 1 \text{ fixed, } 2 \text{ floating.} & \text{if } \beta_2 \geq 1 - \min\{r_1(\beta_1 - 1), \beta_1 - 1\}. \end{cases}$$

The statement of the proposition can be expressed geometrically. We can maximize the social welfare depending on the real interval where β_2 is: Figure 1 shows a profile to maximize the social welfare, based on the segment where the value of β_2 belongs.

The omitted proof compares the various options.

For $\beta_1 = \beta_2 = 0$, this result (agent 1 plays *floating*, 2 *fixed*) is intuitive, since the less kind agent aligns to the kinder one. Also the previous results of this section show that for $\beta_1 = \beta_2 = 0$, the less kind agent should align to the kinder one, to maximize the social welfare. By now, the preparation for analyzing the whole interaction as a game are completed, so we proceed to define and to analyze the game.

5 Reciprocation Attitude Game

We have considered an agent choosing her reciprocation coefficient or her *fixed* or *floating* reciprocation attitude, each choice yielding certain (asymptotic) utility to the agent. This situation is naturally modeled as a game where the strategies of each agent are the above choices and the utility is the asymptotic utility of the interaction. Recall that the utility of agent i is $\lim_{t \rightarrow \infty} \{x_{j,i}(t) - \beta_i x_{i,j}(t)\}$. This is a one-shot game, the attitude being chosen once, before the interaction commences. Analyzing this game allows predicting the situation, supplying some advice to an external party (such as the boss who wants to influence her employees) or to the agents

themselves. As explained after Example 4, human agents usually neither completely mimic the others' behavior, nor do they completely ignore it, which means $0 < r_1, r_2 < 1$. We call this game the *reciprocation attitude game (RAG)*. Theorems 4 and 5 summarize our findings about RAG.

We first characterize the existence of pure NE in this game and subsequently look into their efficiency. Then, we consider how stable these NE are with respect to the best response dynamics. We assume that $k_2 > k_1$ (strictly) in this section; when the kindness is equal, everyone always keeps acting with this equal value.

Theorem 4 *The NE of RAG are characterized as follows:*

$$\begin{aligned} (\text{fixed, fixed}) \text{ is an NE} &\iff \beta_1 \geq r_2 \text{ and } \beta_2 \leq r_1. \\ (\text{float, fixed}) \text{ is an NE} &\iff \beta_1 \leq r_2 \text{ and } \beta_2 \leq 1. \\ (\text{fixed, float}) \text{ is an NE} &\iff \beta_1 \geq 1 \text{ and } \beta_2 \geq r_1. \\ (\text{float, float}) \text{ is an NE} &\iff \beta_1 \leq 1 \text{ and } \beta_2 \geq 1. \end{aligned}$$

The proof utilizes Proposition 4 about utility maximization to see when no deviation is profitable.

Proof. Assume that $\beta_1 \geq r_2$ and $\beta_2 \leq r_1$. If the strategy profile is *(fixed, fixed)*, then, according to Proposition 4, no agent will have an incentive to unilaterally deviate, meaning this strategy profile is indeed an NE.

Assume now that *(fixed, fixed)* is an NE. We prove that $\beta_1 \geq r_2$ and $\beta_2 \leq r_1$ by contradiction. If $\beta_1 < r_2$, then Proposition 4 would imply that agent 1 would like to deviate, contradictory to the profile being an NE. If $\beta_2 > r_1$, Proposition 4 would imply that 2 would like to deviate, contradictory to the NE.

The remaining 3 cases are proven by analogy. \square

Remark 1 (Existence of NE) *If no characterizing condition holds, then no NE exists. For example, no characterizing condition holds when $\beta_1 = 0.8, \beta_2 = 0.9, r_1 = 0.5, r_2 = 0.2$, so no pure NE exists in this case. Since the game is finite, a mixed NE always exists by the classical result by Nash [19].*

We now illustrate the theorem for certain parameter values.

Example 5 *Let $\beta_1 = 0.3, \beta_2 = 0.6$. Theorem 4 states that*

$$\begin{aligned} (\text{fixed, fixed}) \text{ is an NE} &\iff 0.3 \geq r_2 \text{ and } 0.6 \leq r_1. \\ (\text{float, fixed}) \text{ is an NE} &\iff 0.3 \leq r_2. \end{aligned}$$

No other Nash equilibria exist.

5.1 PoA and PoS

The manager or the government may want to know how far the social welfare in an equilibrium is from the maximum possible social welfare. To this end, we consider the famous measures of the efficiency of an equilibrium, namely price of anarchy [16] (PoA) and price of stability [1] (PoS). PoA is the smallest ratio of a social welfare in an NE to the optimum social welfare, and PoS is the largest such ratio.

Theorem 4 provides all the NE, for each set of parameters. Using Proposition 6, we know for each set of parameters what the maximum social welfare is. Calculating the social welfare at each of the Nash equilibria and finding its ratio to the optimum social welfare enables us to find the price of anarchy and stability in the following theorem.

Conditions:	PoA = PoS :
$1 + r_2 - r_2\beta_2 > \beta_1 > r_2$	$\frac{\sum_{i=1}^2 (1-\beta_i) \frac{(1-r_i)k_i + r_i(1-r_j)k_j}{1-r_i r_j}}{(2-\beta_1-\beta_2)k_2}$
$\wedge \{\beta_2 < r_1\}$	
$\{\beta_1 > 1 + r_2 - r_2\beta_2\} \wedge \{\beta_2 < r_1\}$	1
$\wedge \{1 + 1/r_1 - \beta_2/r_1 > \beta_1\}$	
$\beta_1 > 1 + 1/r_1 - \beta_2/r_1$	
$\wedge \{\beta_2 < r_1\}$	$\frac{\sum_{i=1}^2 (1-\beta_i) \frac{(1-r_i)k_i + r_i(1-r_j)k_j}{1-r_i r_j}}{(2-\beta_1-\beta_2)k_1}$
$\{\beta_1 < r_2\} \wedge \{\beta_2 < 1\}$	1
$\{\beta_1 > 1\} \wedge \{1 + r_1 - \beta_1 r_1 > \beta_2\}$	
$\wedge \{\beta_2 > \max\{1 + 1/r_2 - \beta_1, r_1\}\}$	$\frac{(2-\beta_1-\beta_2)k_1}{\sum_{i=1}^2 (1-\beta_i) \frac{(1-r_i)k_i + r_i(1-r_j)k_j}{1-r_i r_j}}$
$\{\beta_1 > 1\} \wedge$	
$\beta_2 > \max\{1 + r_1 - \beta_1 r_1, r_1\}$	1
$\{\beta_1 < 1\} \wedge \{2 - \beta_1 > \beta_2 > 1\}$	$\frac{r_2}{r_1+r_2} \frac{k_1}{k_2} + \frac{r_1}{r_1+r_2}$
$\{\beta_1 < 1\} \wedge \{\beta_2 > 2 - \beta_1\}$	$\frac{r_2}{r_1+r_2} + \frac{r_1}{r_1+r_2} \frac{k_2}{k_1}$

Table 1: The efficiency of NE in reciprocation attitude game.

Theorem 5 *The efficiency of the equilibria is given in Table 1. In the case of equality in the conditions, the highest entry from our conditions that border the equal value is the price of stability, and the lowest entry is the price of anarchy.*

In particular, if $\beta_1 < r_2, \beta_2 < 1$, then PoA = PoS = 1. We now illustrate the efficiency ranges on Example 5.

Example 5 (Continued) *Recall that $\beta_1 = 0.3, \beta_2 = 0.6$. For these values, Theorem 5 implies the following.*

Conditions:	Price of anarchy and stability:
$\{0.3 > r_2\} \text{ and } \{0.6 < r_1\}$	$\frac{\sum_{i=1,2;j \neq i} (1-\beta_i) \frac{(1-r_i)k_i + r_i(1-r_j)k_j}{1-r_i r_j}}{1.1k_2}$
$\{0.3 < r_2\}$	1

Consider Example 2. If agents 1 and 2 have $r_1 = r_2 = 0.5, \beta_1 = 0, \beta_2 = 0.2$ (acting is cheap), then, as just mentioned, PoA = PoS = 1 and the only NE is *(float, fixed)*. This is intuitive, since agent 1 will align to the kinder 2, thereby each agent maximizes the total action and, since acting is cheap, also her own utility and the social welfare.

This completes the analysis of the agents setting their own reciprocation attitudes. The next section considers agents who set both their own reciprocation attitudes and coefficients.

6 Reciprocation Attitude and Coefficient Game

In the previous section we looked at the game of choosing a reciprocation attitude. It is also natural to consider what happens when the other habit, namely, the reciprocation coefficient, is chosen as well. Analyzing this game allows predicting the situation of more choice than the situation analyzed in RAG; for instance, the participants have more willpower or knowledge than in RAG. As before, this is a one-shot game, the attitude and reciprocation coefficient being chosen once, before the interaction commences. As we did for RAG, since people usually neither completely mimic the others' behavior, nor do they completely ignore it, we assume $0 < r_1, r_2 < 1$. We call this game the *reciprocation attitude and coefficient game (RACG)*. This game is analyzed in Theorems 6 and 7.

We first characterize the existence of pure NE in this game and then look into their efficiency, by finding the price of anarchy and stability. This section assumes that $k_2 > k_1$ (strictly).

Theorem 6 *The only Nash equilibria of RACG are characterized as follows:*

<i>An equilibrium profile :</i>	<i>Condition :</i>
(fixed, fixed, $r_1 = \beta_2, r_2 = \beta_1$)	$\iff 0 < \beta_1, \beta_2 < 1.$
(float, fixed, $0 < r_1, r_2 < 1, \beta_1 \leq r_2$)	$\iff \beta_1 < 1 \wedge \beta_2 \leq 1.$
(fixed, float, $0 < r_1, r_2 < 1, r_1 \leq \beta_2$)	$\iff \beta_1 \geq 1 \wedge \beta_2 > 0.$
(float, float, $0 < r_1, r_2 < 1$)	$\iff \beta_1 = \beta_2 = 1.$

The proof is based on Theorem 4, which narrows down the set of possible Nash equilibria, on Proposition 2 and Proposition 3 about utility maximization, and on convergence results from [22] (See Section 2.3.)

Proof. We go over all the NE for RAG from Theorem 4 and look at all the possible choices of r_1 and r_2 to have an equilibrium in the new game. No other equilibria exist, since if no condition of Theorem 4 is satisfied, then even deviating by changing only the attitude is possible.

We begin with (*fixed, fixed*), an NE in RAG if and only if $\beta_1 \geq r_2$ and $\beta_2 \leq r_1$. Given these reciprocation attitudes, Proposition 2 implies that to prevent the only best choice of r_1 being 0 or 1, we must have $(r_2 - \beta_1) = 0$, and to avoid the situation where the only best choice of r_2 is 0 or 1, we must have $(\beta_2 - r_1) = 0$. This implies the necessity of the conditions for an NE with *fixed* attitudes. Theorem 4 and Proposition 2 imply that these conditions are also sufficient to prevent deviations of only the attitude or only the reciprocation coefficient. If agent j simultaneously deviates to another attitude and r_j , then Theorem 3 implies that any $r_j > 0$ yields the same utility, and therefore, this deviation may be considered to consist of attitude only, which is known to be not profitable. This proves the sufficiency.

Consider the profile (*float, fixed*) now, an NE in RAG if and only if $\beta_1 \leq r_2$ and $\beta_2 \leq 1$. Since $r_2 < 1$ implies that $\beta_1 \leq r_2 < 1$, we have the necessity of the conditions for a NE with *floating* and *fixed* attitudes. Theorem 4 implies that deviating in attitude only is not profitable. By Theorem 3, any $r_1, r_2 \in (0, 1)$ suffice for a best response, and so deviating in reciprocation coefficient only is not profitable as well. Consider a deviation of an agent to another attitude and reciprocation coefficient simultaneously. Unless this includes r_2 becoming less than β_1 , we still know from what we have just proven that for this new profile, a deviation by the attitude only would not benefit agent 2, and since changing r_2 has not been profitable, the whole deviation is not profitable. The only remaining option is agent 2 becoming *floating* and changing r_2 to be less than β_1 . This would yield agent 2 the utility of $(1 - \beta_2)(\frac{r_2}{r_1 + r_2}k_1 + \frac{r_1}{r_1 + r_2}k_2)$, by Theorem 2, while he previously had, by Theorem 3, $(1 - \beta_2)k_2$. Since $1 - \beta_2 \geq 0$ and $k_2 > k_1$, the previous profit is not smaller than the new one.

The two remaining cases are similar. \square

Remark 2 (Existence of NE) *When no characterizing condition holds, no NE exists. For instance, if $\beta_1 < 1 < \beta_2$, no characterizing condition holds, and therefore, no (pure) NE exists.*

6.1 PoA and PoS

We now look at the efficiency of these equilibria, proving

Theorem 7 *The efficiency of the NE is given in Table 2.*

We find the possible NE from Theorem 6, and compare their social welfare with the optimal social welfare, found based on the proof of Proposition 5. We only use the ideas of what one should minimize or maximize to maximize the social welfare from the proof of Proposition 5, since the proposition sets reciprocation coefficients to 0 and 1, so we cannot use it directly. To calculate the social welfare, we use the definition of utility and the limit values from Theorems 1, 2, and 3.

Proof. If $0 < \beta_1, \beta_2 < 1$, Theorem 6 implies that there exist exactly two Nash equilibria, namely (*fixed, fixed*, $r_1 = \beta_2, r_2 = \beta_1$) and (*float, fixed*, $0 < r_1, r_2 < 1, \beta_1 \leq r_2$). For the optimal social welfare, we need to maximize both $\lim_{t \rightarrow \infty} x(t)$ and $\lim_{t \rightarrow \infty} y(t)$, as does, for instance, the second NE above, yielding the social welfare of $(2 - \beta_1 - \beta_2)k_2$. Taking the ratios of the social welfare values gives row one in the table from the statement of the theorem.

The remaining cases are proven using the same idea. \square

For an RAG, Theorem 5 implies that small enough β_1, β_2 guarantee that all the NE are optimal. In RACG, however, when $0 < \beta_1, \beta_2 < 1$, the proof of Theorem 7 shows that along with a socially optimal NE, the social welfare of the NE (*fixed, fixed*, $r_1 = \beta_2, r_2 = \beta_1$) relative to the optimum is $\frac{\sum_{i=1,2;j \neq i} (1 - \beta_i) \frac{(1 - \beta_j)k_i + \beta_j(1 - \beta_i)k_j}{1 - \beta_j \beta_i}}{(2 - \beta_1 - \beta_2)k_2}$. When the efforts of acting approach zero for both agents, this expression approaches

$$\begin{aligned} \lim_{\beta_1 \rightarrow 0, \beta_2 \rightarrow 0} & \frac{\sum_{i=1,2;j \neq i} (1 - \beta_i) \frac{(1 - \beta_j)k_i + \beta_j(1 - \beta_i)k_j}{1 - \beta_j \beta_i}}{(2 - \beta_1 - \beta_2)k_2} \\ & = \frac{\sum_{i=1,2;j \neq i} k_i}{2k_2} = \frac{k_1 + k_2}{2k_2} = \frac{1}{2} \left(\frac{k_1}{k_2} + 1 \right). \end{aligned}$$

That is, allowing more freedom (setting own reciprocation attitude and coefficient), we may lose up to half of the efficiency, if k_1/k_2 is small. However, Theorem 7 leaves a sparkle of hope: if at least one agent acts completely effortlessly or even enjoys it, meaning that $\beta_i \leq 0$, then all the NE are socially optimal.

We now turn to the case of n agents, being done with 2.

7 Arbitrarily Many Agents

The original model of [22] is defined for any number $n \geq 2$ of reciprocating agents, where every agent has both r_i and r'_i , the second reciprocation coefficient being the fraction of the action, determined by reacting to the average of all the other agents' actions. They prove convergence, but find the limit only when all the agents are *floating*, and that is the technical obstacle to generalize this paper to n agents. We can, however, assume n *floating* agents and analyze the game of choosing only the reciprocation coefficient, called *reciprocation coefficient game*, by finding its equilibria and their efficiency.

In this case, we discover again that when acting is easy ($\beta_i = 0$), then the kinder agents should pull the less kind ones to act more while not reacting much to the actions they receive by acting less. The results also imply that if all the $1 - \beta_i$ s have the same sign, then PoA = 1.

Conditions:	Price of anarchy:	Price of stability:
$0 < \beta_1, \beta_2 < 1$	$\frac{\sum_{i=1,2; j \neq i} (1-\beta_i) \frac{(1-\beta_j)^{k_i} + \beta_j (1-\beta_i)^{k_j}}{1-\beta_j \beta_i}}{(2-\beta_1-\beta_2)^{k_2}}$	1
$\beta_1 < 1$ and $\beta_2 \leq 1$ but not $0 < \beta_1, \beta_2 < 1$	1	1
$\beta_1 \geq 1$ and $0 < \beta_2 \leq 1$ but not $\beta_1 = \beta_2 = 1$	$\frac{(1-\beta_1-\beta_2)k_1}{(1-\beta_1)k_1+(1-\beta_2)k_2}$	$\frac{(1-\beta_1-\beta_2)k_1}{(1-\beta_1)k_1+(1-\beta_2)k_2}$
$\beta_1 \geq 1$ and $\beta_2 > 1$	1	1
$\beta_1 = \beta_2 = 1$	1	1

Table 2: The efficiency of NE for a reciprocation attitude and coefficient game.

In addition to the just described game, there exist many other variations, even for two agents. For instance, for two agents we are able to analyze the game of choosing the reciprocation coefficient for the not *floating* case too. Another variation would be choosing the reciprocation coefficient in a closed segment $[a, b]$, for any $0 < a < b < 1$. This would limit the domain, but the compactness of the domain may facilitate existence of NE. On the other hand, allowing the extreme points $r_i = 0$ or 1 with a proper handling of the cases of no convergence is also an alternative. We can never cover every possible model, but we believe our model sheds light on the general phenomena.

8 Converging to NE

To analyze the stability of Nash equilibria, we recall the famous best response dynamics [21, Section 2.2], where each agent best responds to the current profile of the others. A reasonable question is when and whether this process converges to a NE. For reciprocation attitude games, we prove that given a NE and any profile, we can let each agent simultaneously choose her reciprocation attitude to maximize her utility, such that it ends up in this NE. The same can be proven for reciprocation coefficient games, described in Section 7. For reciprocation attitude and coefficient game, however, the non-compactness of the domain does not allow a best response to always exist. Therefore, the best response process may be undefined. Details are omitted for lack of space.

9 Conclusions and Future Work

We aim to predict and advise on strategic behavior in reciprocation, in both human-human and human-machine interactions. A reciprocal action is modeled as a balance between the inner self and a reaction to others' actions. We define an agent's utility asymptotically. We then consider choosing the reciprocation attitude or coefficient to maximize her own utility. We finally model the strategic behavior of the reciprocating agents in several games, characterize the NE and their efficiency. We also show that NE may always be achieved by a natural process, the best response dynamics [21, Section 2.2], besides in a RACG. This gives hope for achieving a situation that is stable to unilateral deviations without any regulation.

Our main advice is that both for maximizing own utility and for maximizing the social welfare, if contributing is cheaper than receiving, then, both in choosing the reciprocation attitude and coefficient, the kinder agent should be most stable (be *fixed* or have the reciprocation coefficient $r_i = 0$), and the opposite should be done if contributing is costlier than receiving. When contributing is much cheaper than receiving (β_i s are smaller than all the other parameters), then,

for the reciprocation attitude game and for the reciprocation coefficient game, the price of anarchy is 1, so rationally reciprocating agents will play socially optimally. In such equilibria, the kinder agents are stable and the less kind agents follow the kinder ones. For the reciprocation attitude and coefficient game, the price of stability is 1, but the price of anarchy is positive, meaning that rationally reciprocating agents may play socially optimally, but may also play suboptimally, so that coordination would be useful.

Comparing Theorem 5 for choosing only the reciprocation attitudes to Theorem 7 for choosing the coefficients as well, we observe that more freedom of choice allows for a socially suboptimal equilibrium, achieving as little as about half of the optimal social welfare, if the kindness values are very different. This pitfall emphasizes the importance of cooperation, if more freedom and power lies at our disposal. Like Churchill said⁶: "Where there is great power there is great responsibility".

The analysis also relates to some real-life phenomena. Our results regarding maximizing utility and social welfare show why in life, if acting is not too hard, then following the example of the kindest makes the individuals and the society thrive, which has already been observed [2]. Since being polite usually consists of words and simple gestures, and is therefore quite easy for many people, this explains why people choose this strategy with experience, becoming more polite, as is indeed observed [12]. In diplomacy (Example 1), these results predict that diplomats will be polite to each other, since this does not take much effort. Being polite benefits the individual and the society by making people feel better easily.

Many interesting directions for further research exist: a) Modeling changes in the reciprocity coefficients, attitudes, or β s during the interaction and not only before it starts. b) Modeling probabilistic reaction. c) Looking how the manager can really influence the behavior of the agents. d) Real agents often join and leave the interaction dynamically. For example, people get born and immigrate to a country, some die and emigrate. Therefore, dynamic interaction is very interesting. e) We used others' research, based on real data, as a basis for the model. Therefore, verifying the model on relevant data, like the arms race actions, would be interesting.

Our analysis provides behavioral advice and predicts reciprocation phenomena. It lays the foundation for further modeling of reciprocation, required to even better anticipate and improve the individual utilities and the social welfare.

ACKNOWLEDGEMENTS

This work has been supported by the project SHINE, the flagship project of DIRECT (Delft Institute for Research on ICT at Delft University of Technology).

⁶ This quote is from the French National Convention, 08/05/1793.

REFERENCES

- [1] E. Anshelevich, A. DasGupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden, 'The price of stability for network design with fair cost allocation', in *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pp. 295–304, (Oct 2004).
- [2] R. Axelrod and WD Hamilton, 'The evolution of cooperation', *Science*, **211**(4489), 1390–1396, (1981).
- [3] R.M. Axelrod, *The evolution of cooperation*, Basic books, Basic Books, 1984.
- [4] Robert Axelrod, 'The emergence of cooperation among egoists', *American Political Science Review*, **75**, 306–318, (jun 1981).
- [5] James C. Cox, Daniel Friedman, and Steven Gjerstad, 'A tractable model of reciprocity and fairness', *Games and Economic Behavior*, **59**(1), 17 – 45, (2007).
- [6] Ariel Rubinstein Dilip Abreu, 'The structure of nash equilibrium in repeated games with finite automata', *Econometrica*, **56**(6), 1259–1281, (1988).
- [7] William J. Dixon, 'Reciprocity in united states-soviet relations: Multiple symmetry or issue linkage?', *American Journal of Political Science*, **30**(2), pp. 421–445, (1986).
- [8] Martin Dufwenberg and Georg Kirchsteiger, 'A theory of sequential reciprocity', *Games and Economic Behavior*, **47**(2), 268 – 298, (2004).
- [9] Armin Falk and Urs Fischbacher, 'A theory of reciprocity', *Games and Economic Behavior*, **54**(2), 293 – 315, (2006).
- [10] Ernst Fehr, Urs Fischbacher, and Simon Gächter, 'Strong reciprocity, human cooperation, and the enforcement of social norms', *Human Nature*, **13**(1), 1–25, (2002).
- [11] Ernst Fehr and Simon Gächter, 'Fairness and retaliation: The economics of reciprocity', *Journal of Economic Perspectives*, **14**(3), 159–181, (2000).
- [12] Maria Rosa Baroni Giovanna Axia, 'Linguistic politeness at different age levels', *Child Development*, **56**(4), 918–927, (1985).
- [13] John Gottman, Catherine Swanson, and James Murray, 'The mathematics of marital conflict: Dynamic mathematical non-linear modeling of newlywed marital interaction', *Journal of Family Psychology*, **13**, 3–19, (1999).
- [14] Werner Güth, Rolf Schmittberger, and Bernd Schwarze, 'An experimental analysis of ultimatum bargaining', *Journal of Economic Behavior & Organization*, **3**(4), 367 – 388, (1982).
- [15] D. Kahneman, *Thinking, Fast and Slow*, Farrar, Straus and Giroux, 2011.
- [16] E. Koutsoupias and C. Papadimitriou, 'Worst-case equilibria', in *16th Annual Symposium on Theoretical Aspects of Computer Science*, pp. 404–413, Trier, Germany, (4–6 March 1999).
- [17] W.F.G. Mastenbroek, *Onderhandelen*, Het Spectrum, 1992.
- [18] Heather McAllister, *Who You Are Is What You Do: Making Choices About Life After School*, Wilkins Farago Pty Ltd, illustrated edn., February 2013.
- [19] John Nash, 'Non-Cooperative Games', *The Annals of Mathematics*, **54**(2), 286–295, (September 1951).
- [20] N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, 2007.
- [21] Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory*, volume 1 of *MIT Press Books*, The MIT Press, April 1994.
- [22] G. Polevoy, M.M. de Weerdt, and C.M. Jonker, 'The convergence of reciprocation', in *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '16*, Richland, SC, (2016). International Foundation for Autonomous Agents and Multiagent Systems.
- [23] Matthew Rabin, 'Incorporating fairness into game theory and economics', *The American Economic Review*, **83**(5), pp. 1281–1302, (December 1993).
- [24] Ariel Rubinstein, *Modeling Bounded Rationality*, volume 1, The MIT Press, 1 edn., 1997.
- [25] Hayley Spencer, 'Why reciprocation is key to building business relationships', *Business 2 Community*, (2012).
- [26] Robert L. Trivers, 'The evolution of reciprocal altruism', *The Quarterly Review of Biology*, **46**, 35–57, (mar 1971).
- [27] M. Don Ward, 'Modeling the USA-USSR arms race', *Transactions of The Society for Modeling and Simulation International*, **43**, 196–203, (1984).

Randomized Distribution Feature for Image Classification

Hongming Shan and Junping Zhang*,¹

Abstract.

Local image features can be assumed to be drawn from an unknown distribution. For image classification, such features are compared through the histogram-based model or the metric-based model. By quantizing these local features into a set of histograms, the histogram-based model is convenient and has vectorial representation of image but information could be lost in vector quantization. Unlike the histogram-based model, the metric-based model estimates the metrics over the underlying distribution of local features immediately, achieving better predictive performance. However, the model requires higher computational cost and loses the benefit of vectorial representation of image.

To retain the advantages of these two models, this paper proposes the (doubly) randomized distribution features that represent the underlying distribution of local features in each image as a vectorial feature by utilizing random Fourier feature. We prove the convergences of the similarity and distance based on the randomized distribution feature. Remarkable advantages of the randomized distribution feature are that it has vectorial representation and thus computes efficiently as the histogram-based model. Besides, it provides rigorous theory guarantee and competitive performance as the metric-based model. Compared with several state-of-the-art algorithms, experiments in three real-world datasets justify that our proposed approaches attain competitive classification accuracy with faster computational speed. Furthermore, we indicate that our proposed features can utilize the methods in learning based on vectors, which are broadly studied in traditional machine learning domain, to deal with the problems in learning based on distribution.

1 Introduction

Image representation plays a crucial role in computer vision domains. Generally, images could be represented by a set of high-dimensional, unordered and finite local features. For example, the shapes of object are characterized by a set of local descriptors at edges and corner points [11], and facial expressions are represented by a set of local image patches containing action units [8]. To some extent, these features in each image can be assumed to be drawn from an unknown distribution [25, 34], leading to a learning task based on distribution such as distribution regression with scalar response [32] and distribution to distribution regression [30].

Under this assumption, existing approaches to image classification are roughly categorized into two types: the histogram-based model

and the metric-based one. The histogram-based model usually represents each image by the empirical, one-dimensional histogram that enumerates the occurrence probability of each point set in the bag of visual words. Here, the collection of these words is called a codebook or dictionary. The disadvantages of this method are that the size of codebook is difficult to select, and the computational cost of generating the codebook by the quantization algorithms is expensive. Besides, the information will be lost in the quantization process [34]. In contrast, the metric-based model estimates statistical metrics over the underlying distribution of images with higher accuracy. The advantage of this model is that it does not require quantization techniques and selecting the size of codebook, each of which could result in the loss of performance in image classification. However, these metrics suffer from high computational cost since they operate over pairwise samples. Another drawback of the model is that the matrices obtained by these metrics are only suitable for some specific learning algorithms, *e.g.*, kernel-based algorithms, but cannot be amenable for off-the-shelf use with any standard learning algorithm [22].

In this paper, we propose the (doubly) **R**andomized **D**istribution **F**eature (**RDF**) that could characterize the underlying distribution of local image features of each image as a vector. In this way, the proposed approaches achieve a vectorial representation of distribution, and thus inherit the property of high efficiency of the histogram-based model. Meanwhile, it can approximate the metrics defined on distribution as the metric-based model. Specifically, the distribution of local features is characterized as the mean of random Fourier features which are a low-dimensional embedding representation of kernel mapping function. As a result, the proposed approaches retain advantages from both the histogram-based model and the metric-based model. We also prove the convergences of the similarity and distance based on the randomized distribution feature in this paper. The experimental results show that the proposed methods could achieve competitive performance and reduce computational cost significantly.

The contributions of this paper are summarized here:

- We propose the (doubly) randomized distribution features that represent the distribution of local features extracted from images as a vector;
- We analyze the convergences of the similarity and distance based on the randomized distribution feature;
- Experimental results show that the (doubly) randomized distribution features work better than BoW in vectorial representation, and have competitive performance as the metrics defined over distributions directly. Most importantly, it is easy to implement and computes much faster;
- The proposed method could make learning problems on distribution where each input is a distribution become our traditional machine learning problems, where each input is a vector.

¹ Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai 200433, China.
Emails: {hmshan, jpzhang}@fudan.edu.cn.

* Corresponding author: Junping Zhang.

The paper is organized as follows. Section 2 briefly surveys the associated algorithms that learn from the distribution. Section 3 presents the preliminaries of the metric-based model, especially the similarity and distance between distributions. Section 4 introduces the proposed (doubly) randomized distribution features, and theoretically analyzes the convergences of the proposed approaches. Experiments in Section 5 demonstrate a comprehensible comparison between the performances of the proposed approaches and several recently published methods. Finally, Section 6 presents a conclusive summary.

2 Related works

Associated algorithms that deal with distributions could be roughly divided into two categories: the histogram-based model and the metric-based model. The most popular method in the histogram-based model is the bag of word (BoW) [9]. By quantizing each local feature into one of visual words by using K -means, BoW represents an image as one-dimensional histogram that enumerates the occurrence probability of each local feature of images in the bag of visual words. BoW suffers from high computational cost of generating codebook by K -means and the quantization process that the information could be lost. Therefore, some recent researches are devoted to accelerating quantization process, such as hierarchical K -means [28], KD-tree and random projection tree [7] and so on. To alleviate the loss of information in the quantization process, several researches attempt to learn more discriminant information from images by aggregating local descriptors [1, 16], learning a discriminant codebook [28], and keeping fisher information [31], etc.

Alternatively, the metric-based model defines various metrics such as similarity, distance and divergence between the distributions for avoiding information loss of the histogram-based model. Specifically, mean map kernel (set kernel) [12, 40] measures similarities among pairwise points. Distance metrics between two distributions such as maximum mean discrepancy (MMD) [13, 25] and nonparametric divergence [33, 34, 45] are commonly-used in machine learning and computer vision domains. Unlike the histogram-based model where features must be quantized and vectorized, the metric-based model can achieve better predictive performance since the comparison is done over the underlying distribution of local features. However, the metric-based model requires preserving the whole data sets and calculating metric between training sets and a new unseen set, which makes it infeasible even for a moderate-size problem. Moreover, the metric-based model is only suitable for some special learning algorithms that could use similarity/distance matrix between samples. Though condensing local features of each image could improve the speed and accuracy [45], the aforementioned problem has not been addressed in essence.

To address these issues, we propose an alternative way that represents the underlying distribution of images by random distribution feature, more concretely, by averaging random Fourier feature [36, 37]. Currently, two related works in literature employ random Fourier feature to characterize the probability distribution in cause-effect inference [22], and to construct match kernel heuristically [2]. Compared to these previous studies, the major difference in this paper is that our work provides a theoretical analysis on the convergences of the similarity and distance between images when using random distribution feature. We also propose a doubly randomized distribution feature to represent images for further promoting performance.

3 Preliminary

In this section, we will introduce kernel embedding of the distribution and two metrics defined on distribution in details.

Following [22], the notations used in this paper are summarized in Table 1. Assume that two images are represented by unknown distributions P and Q separately, their local feature sets are $S = \{x_i\}_{i=1}^n \sim P$ and $T = \{z_j\}_{j=1}^m \sim Q$, respectively. Note that the local feature x and z reside in a d -dimensional space and $S \cup T \in \mathcal{X}$. In fact, S and T could construct their empirical distributions P_S and Q_T respectively, each of which is a set of local descriptors. For example, these features can be extracted from the local regions of images by histogram of gradient (HOG) [6] or scale-invariant feature transform (SIFT) [23].

Table 1. Notations used in this paper

$\mathbb{E}[\xi]$	Expected value of random variable ξ
P	True distribution
$S = \{x_i\}_{i=1}^n$	Point set randomly drawn from P
P_S	Empirical distribution of S
\mathcal{X}	Domain of random variable sampled from P and Q
κ	Kernel function from $\mathcal{X} \times \mathcal{X}$ to \mathbb{R}
\mathcal{H}_κ	RKHS induced by κ
$\mu_\kappa(P)$	Kernel embedding of the distribution P
$\mu_\kappa(P_S)$	Empirical kernel embedding of P_S
κ^F	Low-D representation of κ
$\mu_\kappa^F(P)$	Low-D representation of $\mu_\kappa(P)$
$\mu_\kappa^F(P_S)$	Low-D representation of $\mu_\kappa(P_S)$

3.1 Kernel embedding of the distribution

Let P denote the probability distribution of some random variable X taking value in a separable topological space $(\mathcal{X}, \tau_\mathcal{X})$. Then *kernel embedding of distribution P* associated with a continued, bounded, and positive-definite kernel function $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is described as follows:

$$\mu_\kappa(P) := \int_{\mathcal{X}} \kappa(x, \cdot) dP(x), \quad (1)$$

where $\mu_\kappa(P)$ is an element in the reproducing kernel Hilbert space (RKHS) \mathcal{H}_κ associated with kernel function κ [39].

Interestingly, a kernel function κ is said to be *characteristic* if the mapping μ_κ is injective [43], i.e., $\|\mu_\kappa(P) - \mu_\kappa(Q)\|_{\mathcal{H}_\kappa} = 0$ iff $P = Q$. In other words, kernel embedding of the distribution does not lose any information about the distribution when equipped with a characteristic kernel. An example of this kernel is the Gaussian kernel. It will be used throughout this paper and is defined as follows:

$$\kappa(x, x') = \exp\left(-\gamma \|x - x'\|_2^2\right), \gamma > 0. \quad (2)$$

Since it is unrealistic to get both the true distribution P and true embedding $\mu_\kappa(P)$ in practice, we utilize a sample set $S = \{x_i\}_{i=1}^n \sim P$ to construct the empirical distribution P_S instead. As a result, we approximate the empirical kernel embedding $\mu_\kappa(P_S)$ through P_S :

$$\mu_\kappa(P_S) := \frac{1}{n} \sum_{i=1}^n \kappa(x_i, \cdot) \in \mathcal{H}_\kappa. \quad (3)$$

As summarized in [26], the estimator in eq. (3) has several nice properties: 1) kernel embedding of distribution could preserve all the

information about distribution with characteristic kernel; 2) basic operation on distribution can be done by means of inner products in RKHSs; 3) no intermediate density estimation is required. Therefore, many algorithms benefit from eq. (3) such as maximum mean discrepancy [13], kernel dependency measure [14], Hilbert space embedding of HMMs [41] and kernel Bayes' rule [10]. Despite that the estimator in eq. (3) can be improved by utilizing Stein's phenomenon [26], this estimator is commonly used in practice. Furthermore, the convergence of empirical kernel embedding $\mu_\kappa(P_S)$ to the embedding of its population $\mu_\kappa(P)$ in RKHS norm has been proven in [22].

3.2 Mean map kernel

Note that kernel embedding of distribution does not result in any loss of information by using characteristic kernel. The similarity between distribution P and Q , called *mean map kernel* (MMK), is defined as inner product in RKHS [25]:

$$K^{\text{MMK}}(P, Q) := \langle \mu_\kappa(P), \mu_\kappa(Q) \rangle_{\mathcal{H}_\kappa} = \mathbb{E}_{x, z}[\kappa(x, z)], \quad (4)$$

where $x \sim P$ and $z \sim Q$.

When we have the empirical distribution P_S and Q_T of images, similarly, the empirical mean map kernel is calculated as follows [25]:

$$K^{\text{MMK}}(P_S, Q_T) = \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m \kappa(x_i, z_j). \quad (5)$$

It can be seen that MMK is a way of essentially aggregating the pairwise similarity over two local feature sets. It possesses many nice theoretical properties, e.g., it is a positive-definite kernel [12, 25]. However, the computational complexity of this estimator in eq. (5) is $O(mnd)$ where d is the dimension of local feature.

3.3 Maximum mean discrepancy kernel

An alternative metric between two distributions, *maximum mean discrepancy* (MMD) [12], is to measure the distance between two distributions. Based on the same property of characteristic kernel, the distance between two distributions, referred as two-sample problem [13], is defined as a RKHS norm:

$$D(P, Q) := \|\mu_\kappa(P) - \mu_\kappa(Q)\|_{\mathcal{H}_\kappa} \quad (6)$$

$$= \left[\mathbb{E}_{x, x'} \kappa(x, x') + \mathbb{E}_{z, z'} \kappa(z, z') - 2 \mathbb{E}_{x, z} \kappa(x, z) \right]^{1/2},$$

where two independent random variables x and x' are drawn from P , and the other two independent random variables z and z' are drawn from Q . Furthermore, x is independent of z .

When we have the empirical distribution P_S and Q_T , a biased (but asymptotically unbiased) estimator of MMD is obtained based on the law of large numbers:

$$D(P_S, Q_T) = \left[\frac{1}{n^2} \sum_{i, j=1}^n \kappa(x_i, x_j) + \frac{1}{m^2} \sum_{i, j=1}^m \kappa(z_i, z_j) - \frac{2}{mn} \sum_{i, j=1}^{n, m} \kappa(x_i, z_j) \right]^{1/2}. \quad (7)$$

If combining MMD with a level-2 kernel [25], an alternative similarity between two distributions, called MMD-based kernel

(MMDK) or Gaussian-type RBF kernel [5] will be obtained when the Gaussian kernel defined in eq. (2) is used again. It is formulated as a universal kernel [5]:

$$K^{\text{MMD}}(P_S, Q_T) = \exp(-\gamma' \|\mu_\kappa(P_S) - \mu_\kappa(Q_T)\|_{\mathcal{H}_\kappa}^2) \\ = \exp(-\gamma' D^2(P_S, Q_T)), \quad (8)$$

where γ' is a parameter for the level-2 kernel [25]. It is worth mentioning that although the combination of two level kernels makes MMD kernel more flexible on learning procedure, tuning these two bandwidths is very costly since the computational complexity of estimator in eq. (7) is $O((m+n)^2d)$.

4 Randomized Distribution Feature

Since the kernel embeddings $\mu_\kappa(P_S) \in \mathcal{H}_\kappa$ are infinite dimensional for some characteristic kernel functions, kernel matrices are often used for dealing with the dual optimization problem. However, the construction of kernel matrices needs at least $O(n^2)$ computational and memory requirement, prohibitive for large n . Therefore, we employ the random Fourier feature to obtain a low-dimensional representation of $\mu_\kappa(P_S)$ [22] in order to avoid invoking the dual optimization. Easy to implement, the proposed method possesses a lot of additional advantages including vectorial representation, efficient computation, nice theory guarantee and competitive performance.

Assume that kernel function κ is real-valued and shift-invariant, Bochner's theorem [38] shows that for any $x, z \in \mathcal{X}$:

$$\kappa(x, z) = 2C_\kappa \mathbb{E}_{w, b} [\cos(\langle w, x \rangle + b) \cos(\langle w, z \rangle + b)], \quad (9)$$

where $w \sim \frac{1}{C_\kappa} p_\kappa$, $b \sim \mathcal{U}[0, 2\pi]$, $p_\kappa : \mathcal{X} \rightarrow \mathbb{R}$ is the positive and integrable Fourier transform of κ , and $C_\kappa = \int_{\mathcal{X}} p_\kappa(w) dw$ [22]. In this paper, Gaussian kernel in eq. (2) which is a shift-invariant kernel is approximated by eq. (9), if setting $p_\kappa(w) = \mathcal{N}(w|0, 2\gamma I)$ and $C_\kappa = 1$ [22].

Sampling t times from $p_\kappa(w)$ and $\mathcal{U}[0, 2\pi]$, concretely, we have the parameters $\{(w_l, b_l)\}_{l=1}^t$. The kernel mapping $\kappa(x, \cdot)$ is then approximated by the following formula

$$\kappa^{\text{F}}(x, \cdot) = \sqrt{\frac{2}{t}} \left[\cos(\langle w_1, x \rangle + b_1), \dots, \cos(\langle w_t, x \rangle + b_t) \right]^T \in \mathbb{R}^t, \quad (10)$$

which is the low-dimensional representation of kernel mapping function $\kappa(x, \cdot)$ in a t -dimensional space through random Fourier feature [36, 37]. This random Fourier feature has been widely used to approximate kernel function in many applications [4, 20, 21] since its computation is more efficient than those of kernel methods.

By eq. (10), the empirical kernel embedding $\mu_\kappa(P_S)$ is further approximated by

$$\mu_\kappa^{\text{F}}(P_S) = \frac{1}{n} \sum_{i=1}^n \kappa^{\text{F}}(x_i, \cdot) \in \mathbb{R}^t. \quad (11)$$

This estimator has been studied in cause-effect inference [22] and heuristically used in match kernel [2]. Since it represents a distribution by random Fourier feature into a vector, we call it the *randomized distribution feature* (RDF) in this paper. It is noticeable that this estimator is efficient because its computational complexity is $O(ndt)$.

In the following two subsections, we will show how to use RDF to approximate the MMK and MMD between two distributions.

4.1 RDF based similarity

Given the two local feature sets S and T from two images and the sampled parameters $\{(w_l, b_l)\}_{l=1}^t$, vectorial feature I^{RDF} of image is represented by eq. (11). The similarity between two RDFs of images could be formulated as inner product:

$$K^{\text{RDF}} = \langle \mu_{\kappa}^{\text{F}}(P_S), \mu_{\kappa}^{\text{F}}(Q_T) \rangle. \quad (12)$$

It is obvious that the similarity well approximates to MMK and is easy to implement. The computational complexity of this similarity is $O((m+n)dt)$ since it is linear with respect to the size of sample sets. The convergence of the similarity based on RDF to MMK is justified in the following theorem.

Theorem 1 *For any shift-invariant kernel κ , for the given two empirical distributions P_S of P and Q_T of Q on \mathcal{X} , respectively, and any $\delta > 0$, we have*

$$\begin{aligned} & \left| K^{\text{MMK}}(P, Q) - K^{\text{RDF}}(P_S, Q_T) \right| \\ & \leq 2\sqrt{2\log\left(\frac{2}{\delta}\right)\left(\frac{1}{n} + \frac{1}{m} + \frac{1}{t}\right)}, \end{aligned} \quad (13)$$

with the probability greater than $1 - \delta$ over $\{x_i\}_{i=1}^n, \{z_j\}_{j=1}^m$, and $\{(w_l, b_l)\}_{l=1}^t$.

Furthermore, the expected absolute error is

$$\begin{aligned} & \mathbb{E} \left| K^{\text{MMK}}(P, Q) - K^{\text{RDF}}(P_S, Q_T) \right| \\ & \leq 2\sqrt{2\pi\left(\frac{1}{m} + \frac{1}{n} + \frac{1}{t}\right)}. \end{aligned} \quad (14)$$

Proof to this theorem is attached in Appendix. Theorem 1 shows that the similarity based on RDF converges to MMK at a rate of $O(m^{-\frac{1}{2}})$ ($O(n^{-\frac{1}{2}})$) with respect to the size of samples and $O(t^{-\frac{1}{2}})$ with respect to the dimension of low-dimensional embedding space.

4.2 Doubly RDF based similarity

This subsection introduces how to approximate the MMD by RDF. Similar to the introduction of MMD at Sec 3.3, the distance between two distributions represented by RDF is formulated as the Euclidean distance:

$$D^{\text{RDF}}(P_S, Q_T) = \left\| \mu_{\kappa}^{\text{F}}(P_S) - \mu_{\kappa}^{\text{F}}(Q_T) \right\|. \quad (15)$$

Compared to MMD in eq. (7), this distance can be computed efficiently since the computational complexity of this distance is $O((m+n)dt)$, which is linear with respect to the size of sample sets. The convergence of $D^{\text{RDF}}(P_S, Q_T)$ to the MMD $D(P, Q)$ is shown in the following theorem.

Theorem 2 *For any shift-invariant kernel κ , s.t., $\sup_{x \in \mathcal{X}} \kappa(x, x) \leq 1$, for the given two empirical distributions P_S of P and Q_T of Q on \mathcal{X} , respectively, and any $\delta > 0$, we have*

$$\begin{aligned} & D^{\text{RDF}^2}(P_S, Q_T) - D^2(P, Q) \\ & \leq \left[\frac{1}{n} + \frac{1}{m} \right] + 4\sqrt{\log\left(\frac{1}{\delta}\right)\left(\frac{9}{n} + \frac{9}{m} + \frac{16}{t}\right)}, \end{aligned} \quad (16)$$

with the probability greater than $1 - \delta$ over $\{x_i\}_{i=1}^n, \{z_j\}_{j=1}^m$ and $\{(w_l, b_l)\}_{l=1}^t$.

Furthermore, the expected error is

$$\begin{aligned} & \mathbb{E} \left[D^{\text{RDF}^2}(P_S, Q_T) - D^2(P, Q) \right] \\ & \leq \left[\frac{1}{n} + \frac{1}{m} \right] + \sqrt{2\pi\left(\frac{9}{n} + \frac{9}{m} + \frac{16}{t}\right)}. \end{aligned} \quad (17)$$

The proof for this can be seen in the Appendix. Theorem 2 implies that the distance based on RDF converges to MMD at a rate of $O(m^{-\frac{1}{2}})$ ($O(n^{-\frac{1}{2}})$) with respect to the size of samples and $O(t^{-\frac{1}{2}})$ with respect to the dimension of low-dimensional embedding space.

Once RDFs of images are constructed, the similarity between two images is also formulated as

$$\kappa'(\mu_{\kappa}^{\text{F}}(P_S), \mu_{\kappa}^{\text{F}}(Q_T)) = \exp\left(-\lambda' D^{\text{RDF}^2}(P_S, Q_T)\right), \quad (18)$$

which approximates to MMD kernel.

However, there is still a level-2 kernel κ' contained in this similarity. It is observed that eq. (18) can also be represented in a low-dimensional embedding space again by using the random Fourier feature, we thus propose an alternative way to represent the image as a vector by using the random Fourier feature twice, called *doubly randomized distribution feature* (DRDF) in this paper. It is defined as follows:

$$I^{\text{DRDF}} = \kappa'^{\text{F}}(\mu_{\kappa}^{\text{F}}(P_S), \cdot) \in \mathbb{R}^{t'}, \quad (19)$$

where $\kappa'^{\text{F}}: \mathbb{R}^t \rightarrow \mathbb{R}^{t'}$ and t' is the dimension of low-dimensional embedding space for approximating the RHKS $\mathcal{H}_{\kappa'}$ associated with kernel function κ' . In this way, the similarity in eq. (18) can be easily calculated by following inner product:

$$K^{\text{DRDF}}(P_S, Q_T) = \langle I_{P_S}^{\text{DRDF}}, I_{Q_T}^{\text{DRDF}} \rangle. \quad (20)$$

where the computational complexity of this estimator is $O((m+n)dt + tt')$. Compared to eq. (8), there is no parameter to be tuned in eq. (20), resulting in high efficiency of computing DRDF.

4.3 Summary of the proposed methods

To facilitate the understanding of the proposed methods, a workflow is shown in Figure 1. As depicted in the figure, the local features of sample image are assumed to be drawn from an unknown mixture distribution which contains the scene features and human activity features to describe the event [19]. From bottom to top, the similarity would be more accurate as increasing dimension of low-dimensional embedding space into infinity, but the computational cost will become more expensive. From left to right, level of kernel increases in the methods with more flexibility. Technically, the level of kernel can be more than 2. Most importantly, each distribution is well represented by a vector.

5 Experiment

This section presents the application of proposed methods on image classification and distribution regression with scalar response.

5.1 Image Classification

In this section, we show the empirical performance of the proposed (doubly) randomized distribution feature in three real-world image classification tasks.

For image classification tasks, the images are represented as ‘‘sets of features’’(SOF), e.g., sets of unordered local feature vectors. The

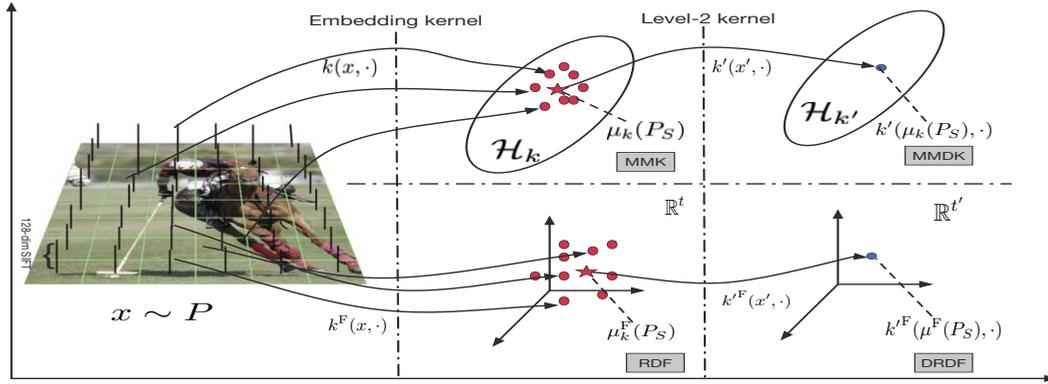


Figure 1. Workflow of the proposed methods. From bottom to top, the similarity would be more accurate and computational cost would be more expensive. From the left to right, level of kernel increases in the methods. Note that the red star denotes the mean point of red points.

proposed methods convert SOF into a vectorial representation like BoW. Therefore, it can be used off-the-shelf in conjunction with any learning algorithm for subsequent image classification. In this paper, we take multi-class SVM as the learning algorithm. For comparison, several algorithms are chosen from the histogram-based model and the metric-based model as follows:

The histogram-based model The BoW model is taken as the baseline algorithm. When we employ linear kernel and Gaussian kernel, the methods are called **BoW_L** and **BoW_G** respectively. For the fair comparison with other methods, the Euclidean distance is used in **BoW_G** method. The number of visual words is set as 1000 unless noted otherwise.

The metric-based model Three algorithms of the metric-based model, **MMK** [25], **MMDK** [25] and the state-of-the-art nonparametric divergence estimator **NPKL** [34], are employed for comparison. **MMK** has only one parameter λ to be decided, and **MMDK** has two parameters λ and λ' for embedding kernel and level-2 kernel respectively. As for **NPKL** [34], the nonparametric Rényi- α divergence between two distributions is used to approximate the KL divergence by setting $\alpha = 0.99$. Compared to **MMD**, the divergence estimated by **NPKL** is non-symmetric. Therefore, the kernel matrix based on nonparametric divergence should be projected to be a symmetric positive semi-definite matrix by symmetrizing the estimated Gram matrix and then projecting to the core of positive semi-definite matrices [15].

The RDF-based model The proposed **RDF** and **DRDF** are calculated based on our proposed (doubly) randomized distribution feature. Both dimensions of low-dimensional embedding space t for approximating embedding kernel and t' for approximating level-2 kernel are set as 1000 in this paper unless noted otherwise. γ and γ' in the Fourier transform p_κ and $p_{\kappa'}$ are calculated using the median trick separately [22]. For a given t (and t' when used), the similarity matrix we used in experiments is the average of 10 times repetition considering the random sample of w and b .

Parameter setting For **BoW_L**, **RDF** and **DRDF**, we use their similarity matrices directly. For other methods, γ in Gaussian kernel defined in eq. (2) is chosen from $\gamma_0 \times \{2^{-9}, 2^{-8}, \dots, 2^9\}$, where γ_0 is estimated by median trick. The penalty to points within the margin C is chosen from $\{2^{-7}, 2^{-6}, \dots, 2^4\}$. C and (when used) γ are chosen through joint 3-fold cross-validation on the training set. Note that there are two γ for different level Gaussian kernel in **MMDK**, it is pretty hard to tune these two γ by cross-validation because of the high computational cost of **MMDK**. According to the strategy used in [25], the best γ in **MMK** obtained by cross-validation is used for

embedding kernel in **MMDK**, the γ' in level-2 kernel is then tuned by cross-validation. Finally, the 5th nearest neighbor in these estimators is used according to the suggestion in [34].

Feature extraction Local features are extracted as follows. The SOF representation of an image is based on the *dense* SIFT descriptors where step size 10 is used to sample image patches and the size of each patch is 12 in this paper unless noted otherwise. We only use the grayscale images to extract SIFT features and each image is represented by a set of 128-dimensional feature vectors. In order to reduce computational cost of the metric-based model, the dimension of SIFT is reduced by principal component analysis in our experiments, preserving 80% variance [34]. Note that each SOF may have different size, depending on the size of image.

Assessing running time For assessing the computational efficiency, each method was implemented in MATLAB[®] 2014b and executed on a server which has a total RAM of 512 GB and four AMD Opteron 6378 processors, each of which contains 16 cores. The running time of each algorithm for constructing the similarity/divergence matrix is assessed in this paper.

Algorithm implementation Multi-class SVM in LibSVM package [3] is employed for image classification tasks in this paper. Besides, feature extraction of image and K -means use the *PHOW* and *kmeans* functions of the VLFEAT package [44] respectively. Furthermore, the code of **NPKL** is provided by [34] and the codes of **MMK**, **MMDK** and the proposed **(D)RDF** are implemented in *MEX* C++ files which are invoked by MATLAB.

5.1.1 Description of three benchmarked datasets

In this subsection, we will describe three benchmarked datasets for different image classification tasks.

ETH-80 dataset [17] is widely used for *object classification*. This dataset contains 8 categories of objects. Each category has 10 different objects, and each object has 41 images from different view angles. Here we can suppose that the images with different view angles are drawn from the same distribution of that object. Moreover, all the images from one category could empirically describe the distribution of that category. Following [34], we extract dense SIFT descriptors in each patch of size 6 for the whole 3280 images in our experiments. The purpose of this experiment is to classify these objects into the 8 categories. In order to save the computational cost of the metric-based model, SIFT features are reduced to 29 dimensions by PCA in this experiment. Thus, each image is represented by a set of 576 29-dimensional features and this dataset produces 1, 889, 280

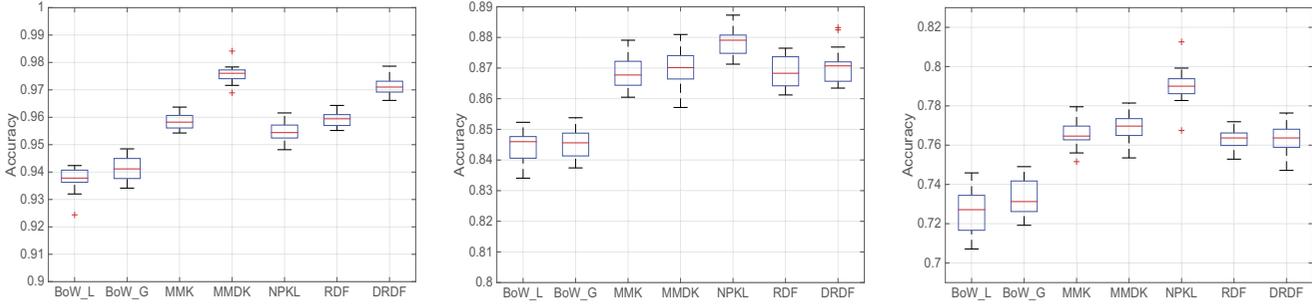


Figure 2. Accuracies on (a) ETH-80 dataset, (b) OT dataset and (c) SE dataset.

SIFT features in total.

OT data set [29] we consider here is a widely used benchmark for *scene classification*. In general, a scene image can be described by a distribution of local features, *e.g.*, the proportion of sky, water, tree, etc. OT dataset includes 8 outdoor scene categories: coast, forest, highway, inside city, mountain, open country, street and tall building. There are 2688 images in total, and each image is in 256×256 pixels. The purpose of this dataset is to classify test images into one of the categories. The original SIFT features are reduced to 30-dimension by using PCA. A typical image is thus represented by 484 30-dimensional local features, which means a total of 1,300,992 SIFT features are extracted from this dataset.

UIUC Sport Event (SE) datasets [19] is considered in the third experiment since the various foreground activities of this dataset make it more difficult than other traditional scene classification, *e.g.*, the OT dataset we used above. This dataset contains Internet images of 8 sport event categories: badminton, bocce, croquet, polo, rock climbing, rowing, snowboarding, and sailing. Each image can be viewed as a *mixture* distribution of scene features and human activity feature to describe the event [19]. The number of images in each category varies from 137 to 250. We use all the 1574 images in experiments. As the size of images varies, the number of local features in each SOF varies from 88 to 484. As a result, there are totally 535,678 SIFT features, each of which is reduced to 34 dimension.

5.1.2 Classification accuracy

For fair comparison and saving computational cost of metric based model, we employ 2-fold cross-validation to split data, which means 50% of data set for training and remaining 50% for testing. The average performance of 20 random runs is reported in Figure 2. From these three experimental results, it can be seen that the metric- and RDF-based methods outperform **BoW** model since the quantization in **BoW** results in the loss of information—potentially a lot of information. As the similarities based on **RDF** and **DRDF** are the approximators to **MMK** and **MMDK** respectively, it is not difficult to see that **DRDF** and **RDF** perform slightly worse compared with **MMK** and **MMDK**. These results justify that our proposed (**D**)**RDF** achieve competitive performance with the metric-based model and better performance than that of **BoW**.

In order to show whether the differences between the proposed methods and their corresponding versions in metric-based model are significant, a paired *t-test* at the significant level 5% is performed on these three real-world datasets. With this significant test, the result shows that **RDF** and **MMK** achieve statistically same performance on the ETH-80 and OT datasets. Meanwhile, **DRDF** and **MMDK** are

statistically significant on these three datasets. A possible reason is that random Fourier feature is used twice in **DRDF**, leading to the loss of much more information for prediction when compared with **RDF**. Note that the *t-test* relies on the pre-specified dimension of vectorial representation in the proposed methods. Theorem 1 and 2 indicate that **RDF** and **DRDF** converge to **MMK** and **MMDK** respectively as the dimension of vectorial representation increases.

Comparisons between algorithms in each model show that non-linear feature, *i.e.* mapped into kernel feature space, achieves higher accuracy than original feature space. It can be noticed that **NPKL** achieves best performances on two of three datasets since its non-parametric estimation of divergence based on *k*-nearest neighbor. Remember that the metric-based model suffers from the expensively computational cost.

5.1.3 Effect of parameters

We examine the effect of parameters upon the performance of the proposed (doubly) randomized distribution feature on ETH-80 dataset.

Dimension of vectorial representation For fair comparison, the number of visual words in **BoW**, the dimension of embedding space t in **RDF** and another dimension t' in **DRDF** are set as the same value since this is the dimension of vectorial representation of each image. To analyze the influence of this value, we vary it from 10 to 10000 and report the results in Figure 3(a). It can be seen that 1) the (doubly) randomized distribution feature work better than histogram representation of **BoW** and 2) all the algorithms converge when the dimension is greater than 1000. Note that **BoW** has not degenerated in performance as the dimension increases because 10000 is still small compared to the number of the all SIFT features extracted from this dataset.

Running time as the dimension increases Running time for constructing the similarity/distance matrices versus the dimension of image representation is reported in Figure 3(b). Since **BoW_L** and **BoW_G** spend almost the same time constructing similarity matrix and distance matrix, we combine them as one to show their running time. From Figure 3(b) it can be seen that the **BoW** needs higher computational cost than **RDF** and **DRDF**, especially when the number of vectorial representation gets large. **DRDF** needs more time than **RDF** slightly since the random Fourier feature is applied twice in **DRDF**.

Effect of two dimensions t and t' in DRDF To show the effect of **DRDF** caused by two dimensions t and t' , a subset of 400 images is used to tune these two dimensions in order to reduce storage size. The effect of **DRDF** with various t and t' is shown in Figure 3(c). It

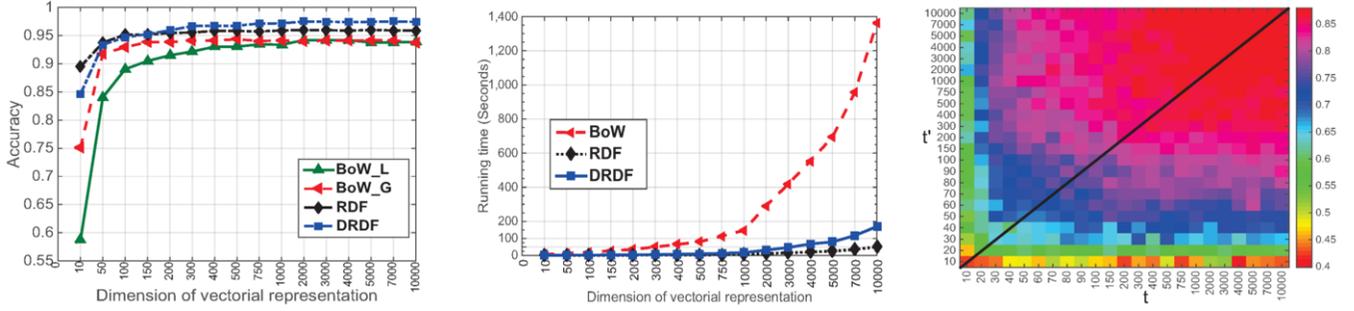


Figure 3. Figures are (a) Varying the dimension of vectorial feature of images; (b) Running time for constructing the similarity matrices; and (c) Sensitivity of DRDF with respect to t and t' .

can be seen that both dimension t and t' are important to the performance of DRDF, and dimension t tends to have more impact on the predictive performance compared to dimension t' . This conclusion coincides with effect of the two bandwidths in MMDK [25].

Influence of dimension reduction In order to investigate the influence of dimension reduction via PCA, we also perform experiments on raw SIFT features which are 128-D feature and show the experimental results in Table 2. Due to expensively computational cost of the metric-based model, the performances of MMK, MMDK and NPKL are not included here.

Table 2. Classification accuracies and their standard deviations (in brackets) on three benchmarked datasets with raw SIFT features.

Datasets	BoW_L	BoW_G	RDF	DRDF
ETH-80	0.9464 (0.0165)	0.9508 (0.0127)	0.9612 (0.0091)	0.9730 (0.0113)
OT	0.8506 (0.0117)	0.8569 (0.0103)	0.8701 (0.0145)	0.8749 (0.0146)
SE	0.7444 (0.0245)	0.7553 (0.0261)	0.7807 (0.0229)	0.7880 (0.0160)

By comparing classification accuracies on raw SIFT features as shown in Table 2 and accuracies with pre-proceeding by keeping 80% variance as reported in Figure 2, we can see that each algorithm gains slightly improvement on its raw SIFT feature. On average, BoW-based algorithms improve about 1.5% while RDF-based ones improve only about 0.5%. We notice that RDF-based algorithms still achieve better performance than that of BoW-based algorithms on raw SIFT features. This means that although reducing the dimension of SIFT features is not a necessary step, it is worth doing this step so that the proposed algorithms can attain lower computational cost in the dimension-reduced space.

5.1.4 Running time over three datasets

In this subsection, we compare the running time of each algorithm for constructing the similarity/distance/divergence matrices over the aforementioned four datasets by using their whole samples.

Running time of each algorithm is reported in Table 3. We can see that the metric-based algorithms consume more computational time than other algorithms do for attaining good performance. Even though BoW saves more time than the metric-based models, it has the worst performance among these algorithms we used since a lot of information may be lost in the quantization process. To conclude, our proposed algorithms require less computational time yet achieve

Table 3. Running time among different algorithms (seconds).

Dateset	BoW	NPKL	MMK	MMDK	RDF	DRDF
ETH-80	146	10277	4066	11924	5.9	17.5
OT	114	5171	1812	5227	3.8	11.0
SE	70	1022	352	1045	2.0	4.7
Average	110	5490	2076	6065	3.9	11.0

competitive predictive performance as the metric-based models do. More specifically, our proposed RDF and DRDF are at least 10 times faster than BoW in vectorial representation with achieving higher accuracies, and at least 500 times faster than the metric-based models with competitive performance on average.

5.2 Application on learning problems on distribution

Besides image classification, we also apply randomized distribution feature to learning problems on distribution. Distribution regression with scalar response [35] is considered here where each input is distribution and output is the scalar response. The setup of this experiment is to learn the skewness of Beta distribution when given their sample set. We generated 300 sample sets from Beta(a, b) distributions where a was varied between [3, 20] randomly and b was fixed to be 3. We used 200 sample sets for training and 100 for testing. Each sample set consisted of 500 distributed i.i.d. points drawn from Beta($a, 3$). Note that the skewness of Beta(a, b) can be calculated as

$$\frac{2(b-a)}{(2+a+b)} \sqrt{\frac{1+a+b}{ab}}$$

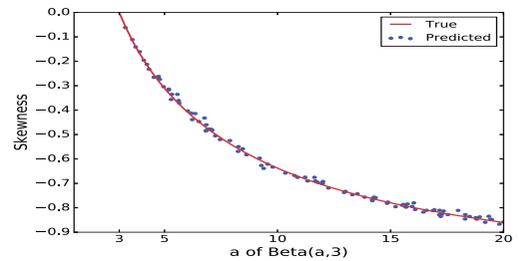


Figure 4. Skewness of Beta distribution

In this experiments, we used a 20-dimensional random distribution feature to represent a Beta distribution and regressed this vectorial representation to its skewness by least squared regression method. Figure 4 displays the predicted values for the 100 test sample sets.

Here we only report the result of RDF, because RDF has provided an accurate representation of distribution, and DRDF is to some extent a nonlinear regression with RDF. This experiment shows the proposed method could make learning problem on distribution become a traditional machine learning problem whose input is vector. More real-world applications can be concluded as the learning problems on distribution and benefited from our proposed features, such as counting the pedestrian or cells from a given image [18] and detecting anomaly group [27].

6 Conclusion and Discussion

In this paper, we introduce the randomized distribution feature to represent distribution. In this manner, the underlying distribution of local features extracted from images can be represented as a vector in image classification. Furthermore, we propose an alternative way to represent image by a doubly randomized distribution feature for further improving predictive performance. We also justify the convergences of the similarity and distance based on RDF. Our recommended feature representation of images inherits the advantages of both the histogram-based model and the metric-based model. It has vectorial representation and computes efficiently like BoW model, and has nice theory guarantee and competitive performance as the metric-based model. Experiments in three benchmark datasets justify these strengths of our proposed approaches. Furthermore, the proposed features could make learning problems on distribution become traditional machine learning problems where each input is a vector.

Compared with VLAD [1] / FV [31] that attempt to learn discriminant information for image classification task, our proposed method focuses on a general representation of distribution that could suit for not only image classification, but also other tasks such as distribution regression. To consider the data structure of distribution, a data-dependent random distribution feature based on Nyström method [46] deserves further studying. Theoretically, it is also of interest to derive tight error bound of convergence of similarity and distance based on RDF according to [42].

7 Acknowledgements

This work has been sponsored by the National Science Foundation of China (No. 61273299).

APPENDIX

Proof 1 (to Theorem 1) The similarity based on RDF is calculated as follows:

$$K^{\text{RDF}}(P_S, Q_T) = \frac{2}{nmt} \sum_{i,j,l=1}^{n,m,t} \left[\cos(\langle w_l, x_i \rangle + b_l) \cos(\langle w_l, z_j \rangle + b_l) \right].$$

Taking expectation over $x_i, z_j, (w_l, b_l)$, we derive the following equality

$$\begin{aligned} & \mathbb{E}_{x_i, z_j, w_l, b_l} K^{\text{RDF}}(P_S, Q_T) \\ &= \frac{1}{nmt} \sum_{i,j,l=1}^{n,m,t} \mathbb{E}_{x_i, z_j} \mathbb{E}_{w_l, b_l} 2 \left[\cos(\langle w_l, x_i \rangle + b_l) \cos(\langle w_l, z_j \rangle + b_l) \right] \\ &= \mathbb{E}_{x, z} \kappa(x, z) = K^{\text{MMK}}(P, Q), \end{aligned} \quad (21)$$

where Bochner's theorem in eq. (9) is applied here. Eq. (21) indicates that $K^{\text{RDF}}(P_S, Q_T)$ is an unbiased estimator of $K^{\text{MMK}}(P, Q)$.

By introducing a variable Δ to measure the difference between $K^{\text{RDF}}(P_S, Q_T)$ and $K^{\text{MMK}}(P, Q)$, we have

$$\begin{aligned} \Delta &= K^{\text{RDF}}(P_S, Q_T) - K^{\text{MMK}}(P, Q) \\ &= \frac{1}{nmt} \sum_{i,j,l=1}^{n,m,t} \left[2 \cos(\langle w_l, x_i \rangle + b_l) \cos(\langle w_l, z_j \rangle + b_l) - \mathbb{E}_{x, z} \kappa(x, z) \right]. \end{aligned} \quad (22)$$

We first provide an upper bound on the difference between Δ and its expectation. Note that changing either of $x_i, z_j, (w_l, b_l)$ in eq. (22) results in changes in magnitude of at most $\frac{4}{n}, \frac{4}{m},$ or $\frac{4}{t}$, respectively. We can then apply McDiarmid's theorem [24], given a denominator in the exponent of $n(\frac{4}{n})^2 + m(\frac{4}{m})^2 + t(\frac{4}{t})^2 = 16 \frac{mn+nt+mt}{mnt}$, to obtain

$$P \left[\left| \Delta - \mathbb{E}_{x_i, z_j, w_l, b_l} \Delta \right| \geq \epsilon \right] \leq 2 \exp \left(\frac{-mnt\epsilon^2}{8(mn+nt+mt)} \right).$$

Let $\delta = 2 \exp \left(\frac{-mnt\epsilon^2}{8(mn+nt+mt)} \right) > 0$, we get $\epsilon = 2 \sqrt{2 \log(\frac{2}{\delta}) \left(\frac{1}{n} + \frac{1}{m} + \frac{1}{t} \right)}$. Remember that $\mathbb{E}_{x_i, z_j, w_l, b_l} \Delta = 0$ as shown in Eq. (21), thus at least $1 - \delta$, we have

$$|\Delta| \leq 2 \sqrt{2 \log(\frac{2}{\delta}) \left(\frac{1}{n} + \frac{1}{m} + \frac{1}{t} \right)}. \quad (23)$$

So we derive the first inequality of theorem. Next we will derive the second inequality, i.e., the expected absolute error between $K^{\text{RDF}}(P_S, Q_T)$ and $K^{\text{MMK}}(P, Q)$. The expected absolute error is

$$\begin{aligned} \mathbb{E}|\Delta| &= \int_0^\infty P[|\Delta| \geq \epsilon] d\epsilon \\ &\leq \int_0^\infty 2 \exp \left(\frac{-mnt\epsilon^2}{8(mn+nt+mt)} \right) d\epsilon = 2 \sqrt{2\pi \left(\frac{1}{m} + \frac{1}{n} + \frac{1}{t} \right)}. \end{aligned} \quad (24)$$

Here eq. (24) is from the fact that expectation over non-negative probability distribution, i.e., $\mathbb{E}[X] = \int_0^\infty x f_X(x) dx = \int_0^\infty P[X \geq x] dx, \forall x \geq 0$. ■

Proof 2 (to Theorem 2) This proof resembles Proof 1. We first bound the difference between $D^{\text{RDF}^2}(P_S, Q_T)$ and $D^2(P, Q)$ by introducing a variable Δ as follows

$$\Delta = D^{\text{RDF}^2}(P_S, Q_T) - D^2(P, Q)$$

Similarly, changing either of x_i, z_j or (w_l, b_l) results in changes in magnitude of at most $\frac{12}{n}, \frac{12}{m},$ or $\frac{16}{t}$, respectively. Applying McDiarmid's theorem [24] gives a denominator in the exponent of $n(\frac{12}{n})^2 + m(\frac{12}{m})^2 + t(\frac{16}{t})^2 = \frac{256mn+144t(n+m)}{mnt}$, to obtain

$$P \left[\Delta - \mathbb{E}_{x_i, x_{i'}, z_j, z_{j'}, w_l, b_l} \Delta \geq \epsilon \right] \leq \exp \left(- \frac{mnt\epsilon^2}{128mn+72t(n+m)} \right).$$

Different from Proof 1, $D^{\text{RDF}^2}(P_S, Q_T)$ is asymptotically unbiased estimator and the expectation over the difference is bounded by

$$\mathbb{E}_{x_i, x_{i'}, z_j, z_{j'}, w_l, b_l} \Delta \leq \frac{1}{n} + \frac{1}{m}.$$

Thus, we have the following inequality

$$P \left[\Delta \geq \left[\frac{1}{n} + \frac{1}{m} \right] + \epsilon \right] \leq \exp \left(- \frac{mnt\epsilon^2}{128mn+72t(n+m)} \right).$$

The two inequalities in Theorem 2 can be derived from above inequality similarly to Proof 1. ■

A detailed version of proof to theorems 1 and 2 can be found at http://www.iipl.fudan.edu.cn/~zhangjp/supp/rdf_sup.pdf.

REFERENCES

- [1] Relja Arandjelovic and Andrew Zisserman, 'All about VLAD', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1578–1585, (2013).
- [2] Liefeng Bo and Cristian Sminchisescu, 'Efficient match kernel between sets of features for visual recognition', in *Advances in Neural Information Processing Systems*, pp. 135–143, (2009).
- [3] Chih-Chung Chang and Chih-Jen Lin, 'Libsvm: A library for support vector machines', *ACM Transactions on Intelligent Systems and Technology*, **2**(3), 27, (2011).
- [4] Radha Chitta, Rong Jin, and Anubhav K Jain, 'Efficient kernel clustering using random fourier features', in *IEEE 12th International Conference on Data Mining*, pp. 161–170, (2012).
- [5] Andreas Christmann and Ingo Steinwart, 'Universal kernels on non-standard input spaces', in *Advances in Neural Information Processing Systems*, pp. 406–414, (2010).
- [6] Navneet Dalal and Bill Triggs, 'Histograms of oriented gradients for human detection', in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pp. 886–893, (2005).
- [7] Sanjoy Dasgupta and Yoav Freund, 'Random projection trees for vector quantization', *IEEE Transactions on Information Theory*, **55**(7), 3229–3242, (2009).
- [8] Paul Ekman and Wallace V Friesen, *Facial Action Coding System*, Consulting Psychologists Press, 1978.
- [9] Li Fei-Fei and Pietro Perona, 'A bayesian hierarchical model for learning natural scene categories', in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 524–531, (2005).
- [10] Kenji Fukumizu, Le Song, and Arthur Gretton, 'Kernel Bayes' rule', in *Advances in Neural Information Processing Systems*, pp. 1737–1745, (2011).
- [11] Kristen Grauman and Trevor Darrell, 'The pyramid match kernel: Efficient learning with sets of features', *The Journal of Machine Learning Research*, **8**, 725–760, (2007).
- [12] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola, 'A kernel method for the two-sample problem', in *Advances in Neural Information Processing Systems*, pp. 513–520, (2007).
- [13] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola, 'A kernel two-sample test', *The Journal of Machine Learning Research*, **13**(1), 723–773, (2012).
- [14] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf, 'Measuring statistical dependence with Hilbert-Schmidt norms', in *Algorithmic Learning Theory*, pp. 63–77. Springer, (2005).
- [15] Nicholas J Higham, 'Computing the nearest correlation matrix—a problem from finance', *IMA Journal of Numerical Analysis*, **22**(3), 329–343, (2002).
- [16] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez, 'Aggregating local descriptors into a compact image representation', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3304–3311, (2010).
- [17] Bastian Leibe and Bernt Schiele, 'Analyzing appearance and contour based methods for object categorization', in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pp. II–409, (2003).
- [18] Victor Lempitsky and Andrew Zisserman, 'Learning to count objects in images', in *Advances in Neural Information Processing Systems*, pp. 1324–1332, (2010).
- [19] Li-Jia Li and Li Fei-Fei, 'What, where and who? classifying events by scene and object recognition', in *Proceedings of IEEE International Conference on Computer Vision*, pp. 1–8, (2007).
- [20] D Lopez-Paz, S Sra, A Smola, Z Ghahramani, and B Schölkopf, 'Randomized nonlinear component analysis', in *Proceedings of the 31st International Conference on Machine Learning*, pp. 1359–1367, (2014).
- [21] David Lopez-Paz, Philipp Hennig, and Bernhard Schölkopf, 'The randomized dependence coefficient', in *Advances in Neural Information Processing Systems*, pp. 1–9, (2013).
- [22] David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Iliya Tolstikhin, 'Towards a learning theory of cause-effect inference', in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1452–1461, (2015).
- [23] David G Lowe, 'Object recognition from local scale-invariant features', in *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pp. 1150–1157, (1999).
- [24] Colin McDiarmid, 'On the method of bounded differences', *Surveys in combinatorics*, **141**(1), 148–188, (1989).
- [25] Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf, 'Learning from distributions via support measure machines', in *Advances in Neural Information Processing Systems*, pp. 10–18, (2012).
- [26] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Arthur Gretton, and Bernhard Schölkopf, 'Kernel mean estimation and Stein effect', in *Proceedings of the 31st International Conference on Machine Learning*, pp. 10–18, (2014).
- [27] Krikamol Muandet and Bernhard Schölkopf, 'One-class support measure machines for group anomaly detection', in *Uncertainty in Artificial Intelligence*, pp. 449–458. Citeseer, (2013).
- [28] David Nister and Henrik Stewenius, 'Scalable recognition with a vocabulary tree', in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 2161–2168, (2006).
- [29] Aude Oliva and Antonio Torralba, 'Modeling the shape of the scene: A holistic representation of the spatial envelope', *International Journal of Computer Vision*, **42**(3), 145–175, (2001).
- [30] Junier Oliva, Barnabás Póczos, and Jeff Schneider, 'Distribution to distribution regression', in *Proceedings of the 30th International Conference on Machine Learning*, pp. 1049–1057, (2013).
- [31] Florent Perronnin, Jorge Sánchez, and Thomas Mensink, 'Improving the fisher kernel for large-scale image classification', in *European Conference on Computer Vision*, 143–156, (2010).
- [32] Barnabás Póczos, Aarti Singh, Alessandro Rinaldo, and Larry Wasserman, 'Distribution-free distribution regression', *International Conference on Artificial Intelligence and Statistics*, 507–515, (2013).
- [33] Barnabás Póczos, Liang Xiong, and Jeff Schneider, 'Nonparametric divergence estimation with applications to machine learning on distributions', *arXiv:1202.3758*, (2012).
- [34] Barnabás Póczos, Liang Xiong, Dougal J Sutherland, and Jeff Schneider, 'Nonparametric kernel estimators for image classification', in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2989–2996, (2012).
- [35] Barnabás Póczos, Liang Xiong, Dougal J Sutherland, and Jeff Schneider, 'Support distribution machines', Technical report, Carnegie Mellon University, (2012).
- [36] Ali Rahimi and Benjamin Recht, 'Random features for large-scale kernel machines', in *Advances in Neural Information Processing Systems*, pp. 1177–1184, (2007).
- [37] Ali Rahimi and Benjamin Recht, 'Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning', in *Advances in Neural Information Processing Systems*, (2008).
- [38] Walter Rudin, *Fourier analysis on groups*, number 12, John Wiley & Sons, 1990.
- [39] Bernhard Schölkopf and Alexander J Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2002.
- [40] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf, 'A hilbert space embedding for distributions', in *Algorithmic learning theory*, pp. 13–31. Springer, (2007).
- [41] Le Song, Byron Boots, Sajid M Siddiqi, Geoffrey J Gordon, and Alex J Smola, 'Hilbert space embeddings of hidden markov models', in *Proceedings of the 27th International Conference on Machine Learning*, pp. 991–998, (2010).
- [42] Bharath Sriperumbudur and Zoltán Szabó, 'Optimal rates for random fourier features', in *Advances in Neural Information Processing Systems*, pp. 1144–1152, (2015).
- [43] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet, 'Hilbert space embeddings and metrics on probability measures', *The Journal of Machine Learning Research*, **11**, 1517–1561, (2010).
- [44] Andrea Vedaldi and Brian Fulkerson, 'VLFeat: An open and portable library of computer vision algorithms', in *Proceedings of ACM international conference on Multimedia*, pp. 1469–1472, (2010).
- [45] Liang Xiong, Barnabás Póczos, and Jeff Schneider, 'Efficient learning on point sets', in *Proceeding of the International Conference on Data Mining*, pp. 847–856, (2013).
- [46] Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou, 'Nyström method vs random fourier features: A theoretical and empirical comparison', in *Advances in Neural Information Processing Systems*, pp. 476–484, (2012).

ShapeLearner: Towards Shape-Based Visual Knowledge Harvesting

Huayong Xu^{†1}, Yafang Wang^{†1,2}, Kang Feng[†], Gerard de Melo[‡], Wei Wu[†], Andrei Sharf^{II}, Baoquan Chen[†]
[†]Shandong University, China; [‡]Rutgers University, USA; ^{II}Ben-Gurion University, Israel

Abstract. The deluge of images on the Web has led to a number of efforts to organize images semantically and mine visual knowledge. Despite enormous progress on categorizing entire images or bounding boxes, only few studies have targeted fine-grained image understanding at the level of specific shape contours. For instance, beyond recognizing that an image portrays a cat, we may wish to distinguish its legs, head, tail, and so on. To this end, we present ShapeLearner, a system that acquires such visual knowledge about object shapes and their parts in a semantic taxonomy, and then is able to exploit this hierarchy in order to analyze new kinds of objects that it has not observed before. ShapeLearner jointly learns this knowledge from sets of segmented images. The space of label and segmentation hypotheses is pruned and then evaluated using Integer Linear Programming. Experiments on a variety of shape classes show the accuracy and effectiveness of our method.

1 Introduction

Motivation. Over the last decade, we have observed an explosion in the number of images uploaded online. Sharing platforms like Flickr have long been driving forces in turning previously undistributed digital images into an abundant resource with billions of images online. This vast amount of data holds great potential to revolutionize the way computers organize and understand images. Deng et al. [9] introduced ImageNet, a hierarchical organization of images, enabling major advances in object recognition, to the point of current deep convolutional neural networks being able to outperform humans in certain respects [27].

Still, current object recognition systems mostly operate at the coarse-grained level of entire images or of rectangular bounding boxes, while segmentation algorithms tend to consider abstract distinctions (e.g., foreground/background).

In this work, we consider the next level of image understanding and knowledge mining, aiming at a more fine-grained understanding of images by automatically identifying specific shape contours and the parts of objects that they portray. One of the major challenges for this is that there is only limited relevant training data. While it is possible to collect millions of images with social media tags [34] and it is feasible to obtain bounding boxes via crowdsourcing [19], obtaining training data with fine-grained hierarchical image information is much more challenging. Analysis of objects with respect to their parts draws from cognitive research of the human vision systems. Shapes of parts play an important role in the lower stages of object recognition [23]. Given a relatively small object part, humans can recognize the object

when the part is sufficiently unique [4, 3]. Unlike deep convolutional neural networks, humans appear to be able to acquire new categories from very few training examples.

Thus, fine-grained image understanding has remained an open problem in AI, as it requires considerable background knowledge about the objects. Progress on this challenging task has the potential to benefit numerous applications in AI, e.g. in robotics and for self-driving cars to interpret their environment, or in photography and graphics for selective image manipulation (removing or replacing a part of an object).

Contribution. We introduce *ShapeLearner*³, a system that learns the shapes of families of objects, together with their parts and their geometric realization, making the following contributions.

1. ShapeLearner requires only a small number of manually annotated seed shapes for bootstrapping and then progressively learns from new images. It achieves this by jointly performing shape classification, segmentation, and annotation to transfer information from seen to unseen images.
2. ShapeLearner can automatically analyse entirely new kinds of shapes, relying on its inference mechanism based on soft constraints.
3. Rather than learning mere enumerations, the system acquires hierarchical knowledge about the objects and their parts (Figures 1c and d). This hierarchical organization is critical for jointly analysing families of objects.

2 Related Work

Image Knowledge Harvesting. In recent years, several new methods have appeared to organize the growing amount of images on the Web [10]. The most prominent of these is ImageNet [9], a hierarchically organized image knowledge base intended to serve as the visual counterpart to WordNet [11]. While ImageNet merely provides image-level labels, subsequent research attempted to localize individual objects within those images using bounding boxes [13]. The SUN Attribute dataset [24] provides coarse-grained crowd-sourced attributes of scenes (e.g. man-made, enclosed). LabelMe [28] crowd-sourced large amounts of polygon labels, but the system does not support any transfer learning. Moreover, the labels can be arbitrary words and thus require significant cleaning and organization. Our work differs from previous work by learning specific shape contours and subparts of objects and then being able to transfer this knowledge to new images and even new types of objects.

Other types of data have been organized as well. For videos, hierarchical taxonomies have been used to train classifiers [32]. For 3D

¹ The two authors contributed equally to the paper.

² The corresponding author: yafang.wang@sdu.edu.cn

³ <http://irc.cs.sdu.edu.cn/ShapeLearner/>

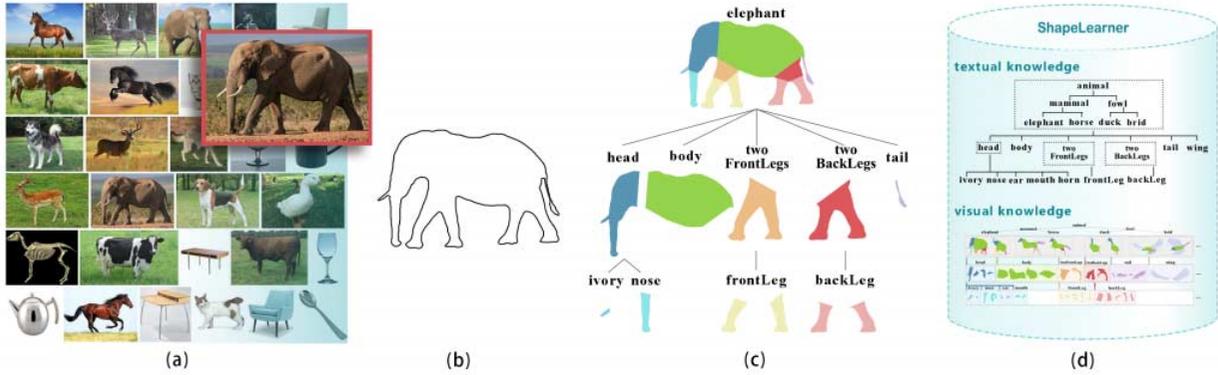


Figure 1: The proliferation of images on the Web (a) enables us to extract shapes to train ShapeLearner (b), a 2D shape learning system that acquires knowledge of shape families, geometrical instances of their inner parts and their inter-relations. Given an unknown shape (c), the system automatically determines a classification, segmentation, and hierarchical part annotation (d).

shapes, ShapeNet [5] and 3DNet [37] organize 3D (CAD) models following WordNet.

Stock graph based shape matching approaches [31] also build a hierarchy, but do not segment the shape into semantically meaningful parts. The goal is to model a complex shape using a hierarchical tree according to geometric features of the shape, which can then be used to compare the similarity of two different shapes. In our method, we use the more recent inner-distance method [20] to model the shape context for shape matching.

Segmentations and Semantic Relationships. Zhang et al. [40] observe that semantic relations of parts be shared among objects in a class and learn a set of classifiers for verb-object relationships within a class. Similarly, graph structures have been introduced for representing semantic relations of parts acquired from sets of images [22, 6, 39]. These methods focus on processing general images and scenes, while we believe to be the first to focus on the inner parts and geometries of individual shape classes.

Grammar-like descriptors for visual words and visual phrases may be defined to enhance image processing and recognition [38]. Recently, Chen et al. [7] presented a method for harvesting large amounts object relationships from images based on their probabilistic structural patterns and geometric characteristics. While their analysis is at the level of object relationships, our method focuses on a fine-grained sub-part analysis. Multiple instances of objects and parts within a class provide important contextual information that can be utilized for joint learning and segmentation [1, 35, 18]. Huang et al. [17] recently presented a data-driven approach for simultaneous segmentation and annotation of free-hand sketches. Although this problem is quite different, we compare our algorithm with theirs later in Section 5.

Deep convolutional neural networks [36, 14] can be trained to produce segmentations, but they do not address our task setting, as they depend on the existence of very large numbers of training examples per label. Related work in this area [16, 36, 14] assumes a standard supervised setting: given a large training dataset for a given class, these methods learn new segmentations. Thus, existing approaches have been limited to very small numbers of object classes, often even just a single one such as human bodies. ShapeLearner, in contrast, is aimed at learning new part segmentations for many classes, given much more limited supervision and relying on knowledge transfer from related classes.

3 Overview and Knowledge Model

High-Level Perspective. ShapeLearner constructs a relational hierarchy that indexes 2D shapes by utilizing taxonomic knowledge of object shape classes and their inner parts. Our goal is to progressively acquire such knowledge by transferring information about indexed shapes onto new ones.

We bootstrap the system by providing labeled seed images in several categories (e.g., mammals, fowl, home appliances). This involves segmenting images collected via Google Images to separate the objects from their environment. Objects are then manually segmented further into meaningful parts and labeled following the WordNet taxonomy. ShapeLearner captures this information about parts and their relations in a tree-like hierarchy by connecting parts to their siblings and ancestors. This can be viewed as a knowledge base with *isA*, *isPartOf*, and *hasShape* relationships.

ShapeLearner includes a knowledge transfer algorithm for understanding unknown shapes. It accounts for both shape geometry and high-level semantic relations from its previously acquired knowledge to infer the correct classification and segmentation of the new object shape. This is illustrated in Figure 2: Given an unknown shape, we compute a raw set of segmentation candidates considering merely the shape’s geometry. We determine additional candidates by matching with geometrically similar shapes and transferring their segmentation. This yields a set of segmentation hypotheses about the unknown shape. ShapeLearner then transfers its knowledge onto the shape by relying on an inference step to remove false hypotheses and select a valid segmentation that complies with the shape’s hierarchical taxonomy. Finally, ShapeLearner transfers this knowledge back by indexing the new shape and progressively updating its store of visual knowledge.

In the final part of the paper, we highlight some applications based on ShapeLearner. We describe the ShapeExplorer system, which supports image retrieval based on partial shape queries and shape morphing, among other things. We also describe our system for keyword-based image retrieval with special support for attributes.

ShapeLearner’s Knowledge. ShapeLearner is directly linked to the WordNet [11] taxonomy, which provides a semantic organization of classes. Focusing on a subset of this taxonomy, we adopt its *isA* class hierarchy and additionally harvest knowledge for *isPartOf* and *hasShape* facts (e.g. *isPartOf*(leg, human), *hasShape*(baseball, round)). Thus, ShapeLearner acquires knowledge of an object’s *shape*, its *parts*, and *shapes of the parts* (see Figure 3).

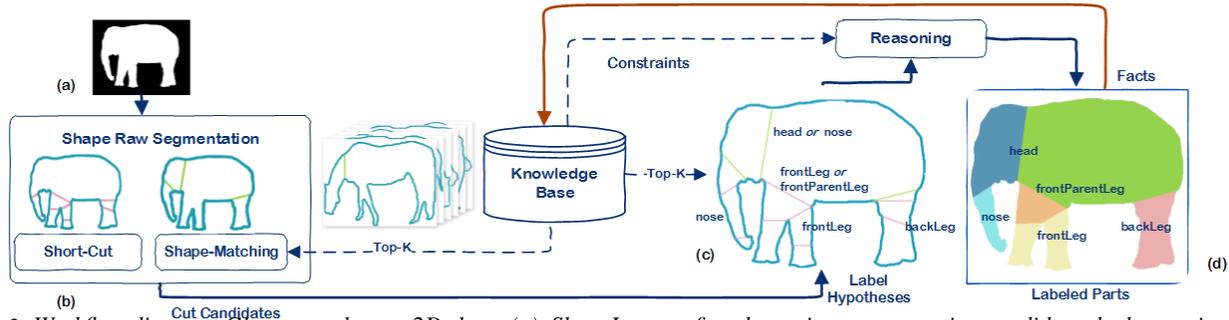


Figure 2: Workflow diagram. Given an unknown 2D shape (a), ShapeLearner first determines segmentation candidates by leveraging cut and shape matching information (b). The system uses its acquired knowledge to label candidates (c). Finally, it makes use of reasoning to prune false hypotheses and infer a classification and semantic segmentation of the shape (d).

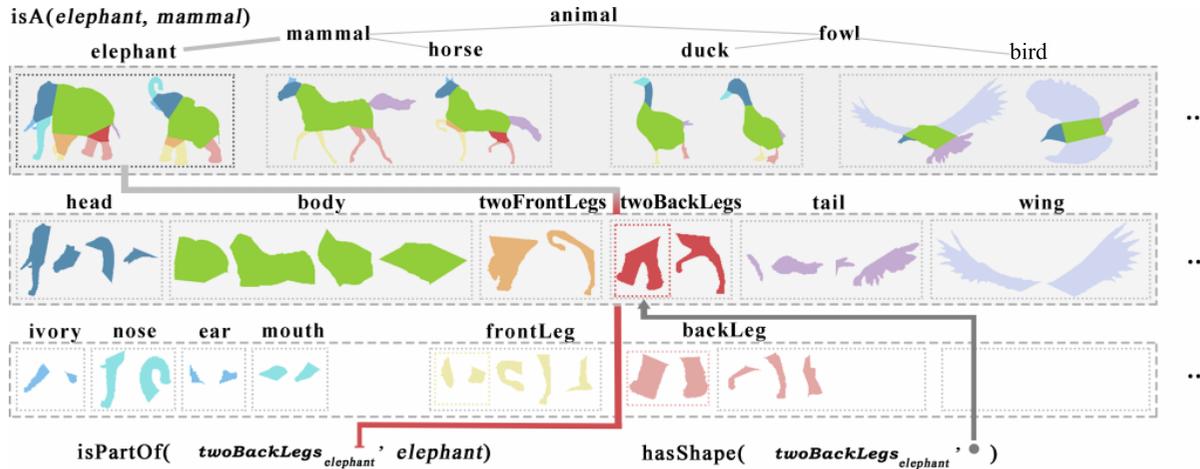


Figure 3: A snapshot of the knowledge in ShapeLearner's hierarchy, zooming in on mammals and fowl. We show also a subset of the relational facts *isA*, *isPartOf*, and *hasShape*.

We begin by defining the four basic concepts that ShapeLearner relies on:

- Shapes $\mathcal{S} = \{s_0, s_1, \dots, s_{n_S}\}$ define the contour of independent 2D objects in an image.
- Classes $\mathcal{C} = \{c_0, c_1, \dots, c_{n_C}\}$ define a category (e.g., species) of objects in the data repository.
- Parts $\mathcal{P} = \{p_0, p_1, \dots, p_{n_P}\}$ define a decomposition of a shape into meaningful components.
- Labels $\mathcal{L} = \{l_0, l_1, \dots, l_{n_L}\}$ define the textual annotations for each part.

Initially, a seed set of parts is manually preprocessed and transferred into ShapeLearner. In this step, the user manually annotates parts in shapes with labels from WordNet (e.g., *head*, *tail*, etc.) as well as semantic relations such as `hasShape(elephant, elephantShape)`, `isA(elephant, mammal)`, and `isPartOf(tail, elephant)`. ShapeLearner stores this information in a hierarchical structure (see Figure 3).

Next, we use ShapeLearner to statistically infer the following knowledge based on available evidence:

- **Part number:** the number of parts per class may be fixed or bounded (e.g., a horse has 2 front legs, an elephant has 1 trunk).
- **Part distinctiveness:** Shape classes may have discriminate parts defined by the frequency of a part in all classes (e.g., the *elephant* class has trunks as a distinct part within the class of *mammals*).

Part distinctiveness is at the core of shape classification and disambiguation. The part distinctiveness score for a part p in class $c \in \mathcal{C}$ is calculated as the inverse fraction of classes containing this part: $\frac{|C|}{|p \in C|} \geq \epsilon$, where ϵ refers to the threshold for acknowledging a part as distinctive. In our experiments, we use $\epsilon = |C|$, which means that the part occurs in just a single class.

4 Shape Analysis

Classification and semantic segmentation of an unknown object shape typically pose a chicken-egg problem: we may require information about one in order to solve the other. Given an unknown 2D shape, ShapeLearner jointly solves for both classification and semantic segmentation by relying on an inference procedure to reason from its knowledge in accordance with statistical constraints and the shape geometry. In fact, it jointly optimizes classification, segmentation, as well as part annotation. We next provide the technical details of this process.

4.1 Shape Segmentation Hypotheses

Given an unknown shape of an object, we compute a set of possible part candidates specified by different cuts in the shape (see cuts in Figure 5(c)). Initially, we compute cuts accounting merely for the shape geometry, applying the short-cut rule of [21], which is motivated by the human vision system. This method yields somewhat consistent cuts tracking the geometric features of the shape contour. Nevertheless, our algorithm does not require an exact segmentation

into meaningful parts but only a loose approximation. A somewhat reasonable segmentation is sufficient at this step.

Next, ShapeLearner transfers additional segment hypotheses from its existing knowledge to further enrich the candidate set. Shape matching plays an important role in adding new cuts that further enrich segmentation and compensate when the short-cut geometry-based method is insufficient. For instance, in Figure 5(a), the smooth elephant head could not be segmented by the short-cut method.

To accomplish this, ShapeLearner finds the best matching shapes in its existing collection and transfers their segmentation onto the input shape. Shape matching is performed using the inner-distance similarity metric [20]. We found this method suitable as it is computationally efficient, rotation-invariant, and robust with respect to other state-of-the-art 2D contour matching techniques (e.g., [2]).

Following the inner distance metric [20], we define $C(\pi(A, B))$ as the matching cost value for two shapes A and B . In a nutshell, given two shapes A and B , described by their contour point sequences p_1, p_2, \dots, p_n and q_1, q_2, \dots, q_m , respectively, we define the cost value $c(p_i, q_j)$ as the χ^2 statistic assessing the similarity of the corresponding point histograms. We compute the optimal matching between A and B , denoted as $\pi : (p_i, q_{\pi(i)})$, using dynamic programming. According to the inner distance approach [20], the mapping from shape A to B should minimize the cost. This is based on dynamic programming to solve the sequence matching problem. We define the minimum cost value by $C(\pi) = \sum_{i=1}^n c(i, \pi(i))$ and the number of matching points is $M(\pi) = \sum_{i=1}^n \delta(i)$, where $\delta(i) = 1$ if $\pi(i) \neq 0$, and 0 if $\pi(i) = 0$.

Next, we define a cut, i.e. $\text{cut}_A(p_i, p_j)$, as the 2D line connecting contour points p_i, p_j in shape A . Thus, to transfer $\text{cut}_A(p_i, p_j)$ from shape A in ShapeLearner onto the input shape B , we simply use the computed shape matching π and transfer $\text{cut}_A(p_i, p_j)$ to $\text{cut}_B(q_{\pi(i)}, q_{\pi(j)})$ (Figure 5).

To reduce noise in the segmentation candidates, ShapeLearner considers only the top $k_1 = 5$ best matching shapes in its collection. Additionally, it relies on the following constraints to remove noisy cuts (Figure 4):

- Cuts should be located in the interior of the shape.
- When cuts intersect each other, only the one corresponding to the longest contour is kept.
- If two cuts are too close together, specifically $\|\text{cut}_B(d) - \text{cut}_B(e)\|_2 \leq \epsilon$, where $\epsilon = 0.01 \times |\text{shape_points}|$, they are merged together.

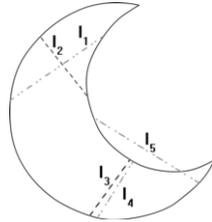


Figure 4: Cut constraints remove all dot dashed cuts (l_1, l_4, l_5).

4.2 Shape-Class and Part-Label Hypotheses

At this point, ShapeLearner has an unknown shape and a set of unlabeled segments, so the shape may belong to different classes and a cut may have different labels. Thus, ShapeLearner next annotates segments with possible label hypotheses from its knowledge and computes a valid segmentation that conforms to its acquired knowledge, by cleaning false segments and label hypotheses.

We assign a unique ID for each cut in the shape and denote an hypothesis as the pair $\text{label}(\text{cut}, \text{label}) [.]$. Additionally, we define class hypotheses as $\text{class}(\text{shape}, \text{class}) [.]$. A hypothesis may become a fact $\text{label}(\text{cut}, \text{label}) [1]$ or be evaluated as false, i.e. $\text{label}(\text{cut}, \text{label}) [0]$, following an inference

process (e.g., $\text{label}(\text{cut}@9, \text{nose}) [1], \text{class}(\text{shape}@1, \text{elephant}) [0]$).

Note that each cut corresponds to a part, so $\text{label}(\text{cut}@9, \text{nose}) [1]$ equals $\text{label}(\text{part}@9, \text{nose}) [1]$. Actually, each cut produces two parts (e.g., body and leg), but here we only consider the leg part. ShapeLearner matches the input shape against its knowledge and selects the top $k = 5$ best matching shapes using the inner distance metric. This yields multiple class and label assignments for the hypotheses.

We define the cut confidence weight with respect to the top k resulting set as follows. Given a cut c_j , label l_i , and hypotheses: $\text{label}(\text{cut}@j, l_i) [.]$, the confidence weight of cut c_j with label l_i is calculated as $w_{c_j, l_i} = \alpha \times p_1 + (1 - \alpha) \times p_2$, ($\alpha = 0.6$ in our experiments), based on two factors:

- p_1 : the confidence of assigning label l_i to cut c_j is $\frac{h_l}{k}$, where h_l is the frequency of label l_i in the top k result set.
- p_2 : A cut may match to more than one similar class. If a cut has many possible label hypotheses (say l_1, l_i, \dots, l_m), the confidence for each part is defined by the part shape matching $w'_{c_j, l_i} = M_{l_i}(\pi) / C_{l_i}(\pi)$. Then $p_2 = \frac{w'_{c_j, l_i}}{\sum_l w'_{c_j, l}}$.

Similarly, we define the class confidence weight with respect to the top k result set as follows. Given the unknown part-shape s_j , class c_i and hypothesis $\text{class}(\text{shape}@j, c_i) [.]$, the confidence of class c_i with respect to the top k result set is calculated as $w_{s_j, c_i} = \frac{h_c}{k}$, where h_c is the number of hits for class c_i .

4.3 Shape Inference

ShapeLearner jointly solves for a consistent classification and labeling by pruning noisy hypotheses and searching for the optimum class and label assignment with respect to its knowledge constraints. We formulate this problem as an Integer Linear Programming (ILP) that considers both cut labels and shape classes to yield a consistent set of truth value hypotheses.

We formulate the ILP variables as follows:

- $x_{p, l} \in \{0, 1\}$ denotes $\text{label}(\text{part}, \text{label})$ hypothesis $l \in \mathcal{L}$ for part $p \in \mathcal{P}$.
- $y_{s, c} \in \{0, 1\}$ denotes $\text{class}(\text{shape}, \text{class})$ hypothesis $c \in \mathcal{C}$ for shape $s \in \mathcal{S}$.

For each shape s , the objective function maximizes the overall confidence of hypotheses (where $w_{x_{p, l}}$ and $w_{y_{s, c}}$ are the confidence weights for cut and class hypotheses respectively, $w_{x_{p, l}} = w_{x_{c, l}}$ in the previous step):

$$\max \sum_{p \in \mathcal{P}, l \in \mathcal{L}} w_{x_{p, l}} x_{p, l} + \sum_{c \in \mathcal{C}} w_{y_{s, c}} y_{s, c}$$

subject to the following constraints derived statistically from the knowledge collection.

Class Constraints.

- A shape s can be assigned to one class at most:

$$\sum_{c \in \mathcal{C}} y_{s, c} \leq 1$$

- **Part-distinctiveness-1:** A shape class assignment should conform to its distinctive parts (if any). Denoting $(l, c) \in D_P$ as the pair set (distinctive part, class), then:

$$\forall p \in \mathcal{P} \wedge c \in \mathcal{C} \wedge (l, c) \in D_P, x_{p, l} - y_{s, c} \leq 0$$

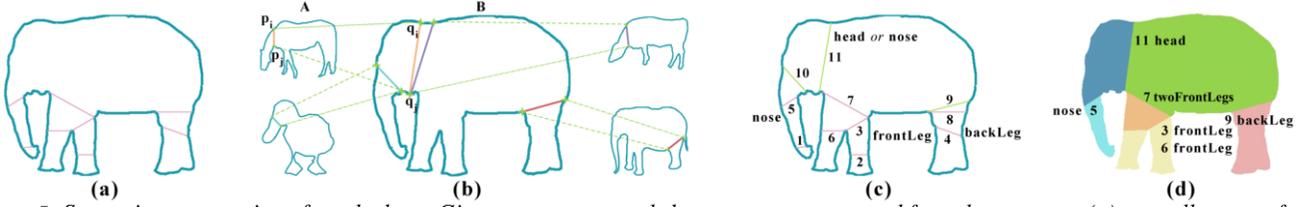


Figure 5: Semantic segmentation of an elephant. Given an unsegmented shape, cuts are computed from the geometry (a), as well as transferred from similar shapes (b). This yields multiple class hypotheses (c) which are pruned, yielding a correct semantic segmentation and annotation of the shape (d).

- **Part-distinctiveness-2:** A shape class should not consist of parts that do not belong to the class (according to `isPartOf`).

$$\forall p \in \mathcal{P} \wedge c \in \mathcal{C} \wedge (l, c) \notin \text{isPartOf}(p, c), x_{p,l} + y_{s,c} \leq 1$$

Label Constraints.

- **Part-inclusion** should conform to ShapeLearner’s part hierarchy. Denoting H_P as the set of inclusion part pairs (i.e., $(l, l') \in H_P$ if and only if l' includes l and $\text{isPartOf}(p, p')$), and using \subset for “included by”, we require

$$\forall p, p' \in \mathcal{P}, l, l' \in \mathcal{L} \wedge (l, l') \in H_P \wedge p \not\subset p', x_{p,l} + x_{p',l'} \leq 1,$$

$$\forall p, p' \in \mathcal{P}, l, l' \in \mathcal{L} \wedge (l, l') \notin H_P \wedge p \subset p', x_{p,l} + x_{p',l'} \leq 1.$$

- **Part-number:** The number of parts in a shape class should conform to the class. Denoting the number of parts as n_P , we add the constraint that

$$\sum_{p \in \mathcal{P}} x_{p,l} \leq n_P c, l.$$

Note that we require the number of parts to be less than or equal to n_P due to possible occlusions of the shape in the image (cf. the back leg in Figure 5).

After this inference step, the accepted clean facts (i.e., those of the form `label(part, label)` [1] or `class(shape, class)` [1]) are integrated into ShapeLearner’s knowledge base. The shape of each part is added as `hasShape(part, part-shape)`. Given a shape of a new class not yet in ShapeLearner, parts of the new class are identified via knowledge transfer. If the new class name is X , new facts are added as `isPartOf(part, X)` and `hasShape(part, part-shape)`.

5 Results

We now present a thorough set of experiments to evaluate ShapeLearner.

Dataset. To compile a dataset for seeding and evaluating ShapeLearner, we collected images from Google Images, Flickr, as well as public domain data used by Ren et al. [25]. We manually collect and sort these images, removing noise, frontal views, and heavily occluded shapes. We then segment the shape from its background with the aid of the open-source tool GrabCut [26]. This segmentation does not need to be precise. Instead, we account for the multiplicity of parts instances to average out the results and remove outliers. The ground truth data was labeled by 3 people. We only keep cuts or draw new cuts agreed by the majority. We extract the shape’s contour and segment it into meaningful parts simply by drawing straight lines inside the contour.

Shape and subparts are classified and annotated before being provided to ShapeLearner. Taxonomy relations (`isA`, `isPartOf`) are taken

from WordNet and textual sources [15, 33] and are used to create the hierarchy.

In total, our dataset consists of 2,020 images in 50 shape families in 7 broad classes (as shown in Table 1). Examples include humans, vases, kangaroos, mammal skeletons, handbags, umbrellas, goblets, and mushrooms. Based on these diverse seeds, our system can classify a wide range of objects if they are somewhat similar to seed images.

Labeling Accuracy. To quantify ShapeLearner’s output quality, we rely on a pixel-based metric to evaluate the part segmentation [17]. Given a segmented part, we measure its overlap with the ground-truth part as the number of pixels that are correctly labeled in the overlap vs. the incorrect ones. A part is considered adequately labeled if a reasonable percentage (precision $> 75\%$) of pixels are in the overlap. The terminology is as follows.

- **True Positive (TP):** correct cut/pixel label
- **True Negative (TN):** correct removed cut/pixel label
- **False Positive (FP):** a cut/pixel label supposed to be removed but not removed
- **False Negative (FN):** a cut/pixel supposed to be labeled, but removed.

Given these, we can use the standard definition of precision as $\frac{TP}{TP+FP}$, recall as $\frac{TP}{TP+FN}$, and $F_1 = \frac{2TP}{2TP+FP+FN}$. The class precision scores given in Table 1 and Table 2 refer to the precision of inferring class labels for the shape.

Baselines. Given all part hypotheses, we evaluate our method (both class constraints and label constraints) against two simpler baselines. However, we experiment with baselines that omit the Part-inclusion constraint and optionally the Part-distinctiveness constraint to highlight the importance of our algorithm’s advanced inference:

- **N:** the inference includes Part-number constraints and class constraints, but not Part-distinctiveness and Part-inclusion constraints.
- **N+D:** the inference includes Part-number, Part-distinctiveness constraints and class constraints, but not Part-inclusion constraints.

Comparison. For an experimental comparison, we used 20 images per family as seed data. The remaining ones in each of the 50 families were manually segmented and used as ground-truth for our evaluation. Table 1 provides an evaluation of the segmentation and classification for these baselines with respect to precision, recall, and the F_1 measure. Our method outperforms these baselines in almost all cases (except for a few cases with lower recall). Figure 6 illustrates a subset of this evaluation, providing F_1 results of baselines and of our method.

In Figure 8(a), we investigate the scalability of our method with respect to the number of initial seeds for classes with size larger than 50. Note that precision, recall, and F_1 of the segmentation increase as

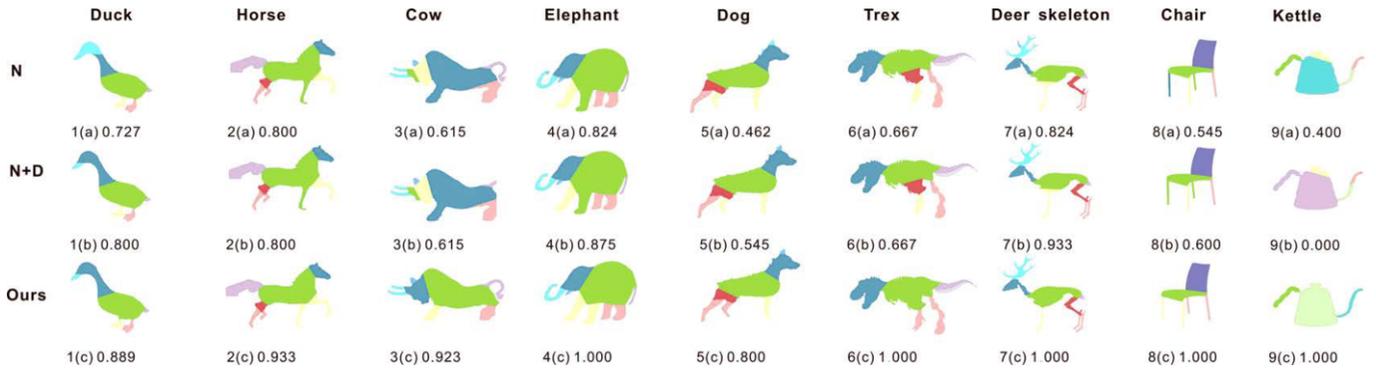


Figure 6: Representative results by our method and baseline solutions. The F_1 measure is shown below each result.

the number of seeds gets larger. After 20 seeds, the results appear to converge and the improvement becomes marginal. Thus, 20 seeds are a reasonable threshold in our experiments. This shows that a small number of seeds can suffice to represent a shape-space sufficiently well and adding more seeds can be redundant.

Figures 8(b) and 8(c) graphically plot a comparison between the baselines and ShapeLearner’s full inference mechanism according to the values in Table 1. We see that even for a small number of seeds, our method outperforms other baselines and has very good precision, recall, and F_1 .

Our classification (Table 1, bottom part) also outperforms the baselines on average. For a few classes, we did not improve over the baselines, since their contours were quite similar and lacked distinctive parts. For example, the small horn of the deer is similar to the ear of the horse. A cat’s tail may be recognized as a back leg in unique situations when the tail hangs down and the cat’s hind legs are occluded. Similarly, skeleton classes can be quite challenging. They are similar in appearance both with other kinds of skeletons and with the respective full living animal. Ribs in the skeleton are similar to legs in size and orientation. Nevertheless, the segmentation of skeletons is often successful in part precisely due to their similarity with living mammals, enabling ShapeLearner to transfer the corresponding knowledge.

Our method can infer a semantically correct segmentation even for classes that are not currently indexed in ShapeLearner. The experimental results in Table 3 show that even without any human-labeled seeds from the target class, ShapeLearner is able to exploit seeds of classes from related categories to transfer segmentation and annotation information. When the seeds from different classes are rather similar with the test shape, the outcome can be even better than the direct segmentation and annotation (see the tiger and bedroom lamp examples in Table 3). Morphological differences between tortoises and other reptiles are quite profound. Thus in this case, the transfer segmentation is less successful.

Figure 7 provides examples of three entirely new classes (a lion, ostrich, and alpaca) that were properly segmented and annotated by ShapeLearner without any prior knowledge about these classes.

We also compared the running time of our method with all constraints (“Ours”) to the method without constraints (“N”). The results show that including all constraints does not increase the complexity, increasing the runtime by just 0.5 seconds on average.

Evaluation and Comparison. Although our paper has a different target, we compare our method with segmentation algorithms for hand-drawn sketches ([17], direct retrieval (DR), and [29]). One major difference is that that work is aimed at analysing the brush strokes, which may contain significant information on the shape’s interior,

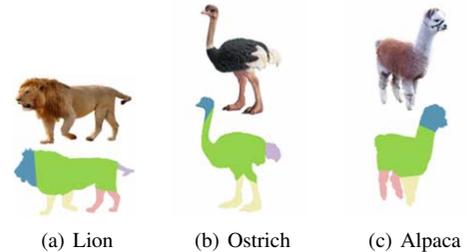


Figure 7: Semantic segmentation of three new shapes (without prior indexing of these classes by ShapeLearner).

while ours considers only the contour. From their dataset, we select all object classes with meaningful contours (omitting three classes consisting of many thin lines rather than clear contours) and compare the average segmentation and annotation precision (see Table 4). While the algorithm of Huang et al. can resolve many ambiguities due to occlusions based on the interior brush strokes, our method nevertheless gives superior results on a majority of classes, demonstrating the effective power of ShapeLearner’s knowledge. For the airplane and vase classes, our method was inferior due to the large variety (airplanes) and non-distinctiveness of parts (vases). Unfortunately, we could not perform a more in-depth comparison (e.g., w.r.t. occlusions and a larger variety of classes) since their code is not publicly available.

6 Use Cases

Finally, we present a set of use cases utilizing ShapeLearner to solve a set of challenging shape related problems.

6.1 ShapeExplorer

We have developed a system called ShapeExplorer, an interactive software tool based on a detailed analysis of images in terms of object shapes and parts. For instance, given an image of a donkey, the system may draw on ShapeLearner and its previously acquired knowledge about zebras and dogs to automatically locate and label the head, legs, tail, and so on. Based on such semantic models, ShapeExplorer can then generate morphing animations, synthesize new shape contours, and support object part-based queries (see Figure 9), as well as clipart-based image retrieval. Details were published in Ge et al. [12]. Please also refer to the URL <https://youtu.be/JTQcQkBhvyk> for an online video of this system.

6.2 Keyword Queries

Our system enables novel forms of image queries referring to specific parts of objects, e.g. for “pans with long handles”.

Table 1: Experimental results for segmentation and annotation (top) and classification (bottom).

System	Mammals	Home Appliances	Misc. Artifacts	Foods	Reptiles	Fowl	Skeletons	All Avg.	
N	68.3%	86.0%	88.5%	100.0%	74.5%	65.8%	57.3%	77.2%	Prec.
N+D	69.2%	90.4%	92.3%	100.0%	74.5%	66.6%	59.8%	79.0%	
Ours	79.4%	92.5%	92.6%	100.0%	84.1%	75.6%	71.8%	85.1%	
N	85.5%	93.3%	93.7%	100.0%	85.4%	90.4%	76.4%	89.2%	Recall
N+D	85.1%	92.4%	94.0%	100.0%	85.4%	90.3%	77.1%	89.2%	
Ours	86.9%	93.6%	94.4%	100.0%	83.0%	91.4%	80.8%	90.0%	
N	74.8%	88.1%	90.2%	100.0%	78.9%	74.3%	64.5%	81.6%	F1
N+D	75.3%	90.9%	93.0%	100.0%	78.9%	74.9%	66.4%	82.8%	
Ours	82.2%	92.5%	93.3%	100.0%	82.9%	81.1%	74.6%	86.7%	
N	65.3%	91.4%	87.8%	93.3%	92.5%	87.8%	61.9%	82.9%	Class
N+D	72.0%	93.2%	92.6%	93.3%	95.0%	88.9%	58.8%	84.8%	
Ours	71.9%	93.7%	92.6%	93.3%	95.0%	89.3%	59.3%	85.0%	

Table 2: Excerpts for segmentation and annotation (top) and classification (bottom).

System	Mammals					Home Appliances			Misc. Artifacts		Foods	Reptiles		Fowl		Skeletons		
	Elephant	Cow	Deer	Horse	Cat	Vase	Hairdryer	Broom	Rifle	Axe	Mushroom	Tortoise	Crocodile	Duck	Bird	Mammals	Dinosaur	
N	74.6%	62.4%	80.6%	64.5%	63.3%	67.8%	96.7%	96.7%	59.1%	93.3%	100.0%	65.6%	68.8%	63.2%	67.4%	62.2%	52.5%	Prec.
N+D	75.8%	63.2%	81.7%	64.6%	65.5%	73.3%	96.7%	96.7%	78.2%	93.3%	100.0%	65.6%	68.8%	63.8%	69.2%	64.3%	55.3%	
Ours	86.0%	71.4%	87.0%	77.9%	79.6%	73.3%	96.7%	96.7%	79.9%	93.3%	100.0%	78.2%	81.4%	74.4%	79.2%	75.1%	68.4%	
N	90.5%	81.3%	87.7%	83.9%	80.3%	80.0%	96.7%	96.7%	85.3%	93.3%	100.0%	80.9%	84.8%	89.5%	90.6%	78.4%	74.5%	Recall
N+D	88.9%	79.6%	87.7%	83.1%	80.6%	78.3%	96.7%	96.7%	86.8%	93.3%	100.0%	80.9%	84.8%	87.5%	92.2%	78.4%	75.9%	
Ours	91.1%	84.2%	90.0%	86.2%	84.8%	78.3%	96.7%	96.7%	88.5%	93.3%	100.0%	81.1%	89.9%	86.7%	93.1%	82.8%	78.8%	
N	81.2%	69.7%	83.0%	72.0%	70.1%	72.1%	96.7%	96.7%	67.8%	93.3%	100.0%	71.4%	75.4%	72.6%	75.6%	68.3%	60.6%	F1
N+D	81.3%	69.5%	83.7%	71.8%	71.5%	75.0%	96.7%	96.7%	81.6%	93.3%	100.0%	71.4%	75.4%	72.4%	77.5%	69.6%	63.1%	
Ours	88.0%	76.5%	87.8%	81.2%	81.4%	75.0%	96.7%	96.7%	83.3%	93.3%	100.0%	79.2%	84.9%	78.6%	84.1%	77.9%	71.4%	
N	94.4%	53.1%	90.4%	37.8%	86.0%	90.0%	96.7%	76.7%	69.0%	70.0%	93.3%	96.6%	90.0%	83.5%	83.3%	34.0%	89.9%	Class
N+D	94.4%	60.5%	80.9%	76.7%	76.0%	86.7%	100.0%	76.7%	93.1%	70.0%	93.3%	96.6%	100.0%	89.9%	76.7%	37.7%	79.8%	
Ours	94.4%	59.3%	80.9%	75.6%	76.0%	86.7%	100.0%	76.7%	93.1%	70.0%	93.3%	96.6%	100.0%	91.1%	76.7%	38.9%	79.8%	

Table 3: Experimental results for with seeds (top) and with only transfer (bottom).

Method	Feline			Reptiles				Lamp			Canine			
	Cat	Leopard	Tiger	Tortoise	Crocodile	Lizard	Gecko	Desk Lamp	Floor Lamp	Bedroom Lamp	Dog	Wolf	Fox	
Precision	79.6%	73.7%	75.1%	78.2%	81.4%	88.2%	88.7%	92.6%	94.6%	85.0%	79.8%	71.5%	77.3%	Direct
Recall	84.8%	91.1%	78.1%	81.1%	89.9%	78.5%	82.4%	90.7%	96.4%	85.6%	92%	84.6%	84.9%	
F1	81.4%	80.3%	76.2%	79.2%	84.9%	82.6%	84.9%	91.4%	95.2%	83.2%	84.5%	76.1%	80.2%	
Precision	66.7%	70.1%	86.7%	46.2%	71.3%	78.4%	81.2%	88.9%	92.9%	91.7%	78.0%	69.8%	74.3%	Trans.
Recall	60.0%	72.6%	70.2%	52.9%	69.5%	71.9%	77.5%	87.0%	100.0%	78.9%	69.2%	77.1%	66.1%	
F1	62.0%	69.6%	76.7%	48.6%	69.8%	74.8%	78.3%	87.7%	95.2%	82.1%	71.6%	71.2%	68.6%	

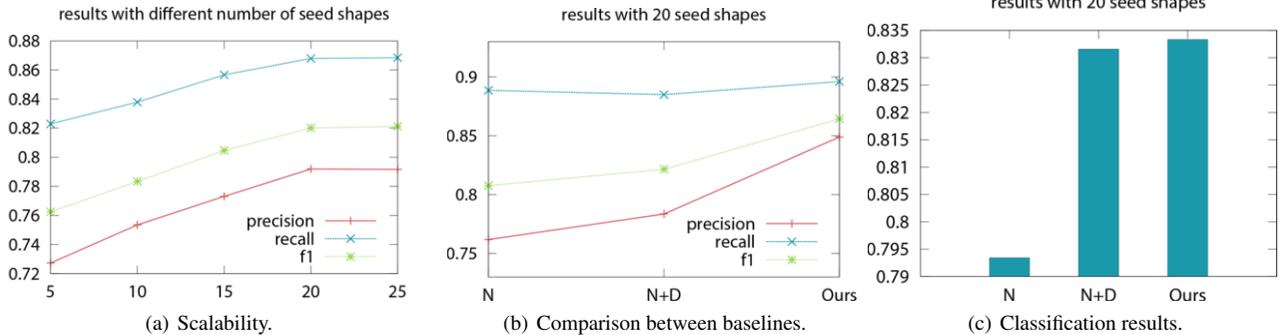


Figure 8: Experimental results graphs. In (a) we show the scalability of the average precision, recall and F1, and in (b) the comparison with other baselines. In (c) we show classification precision comparison with other baselines.

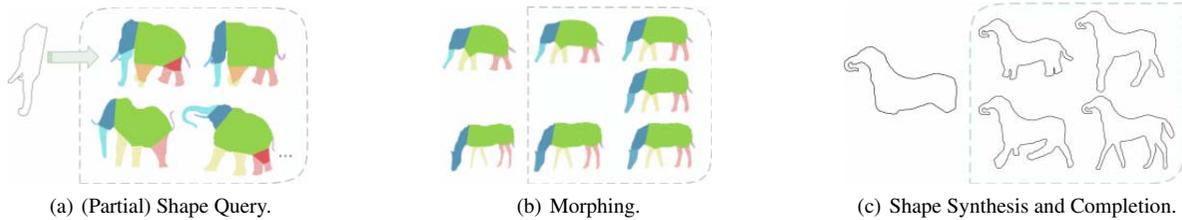


Figure 9: ShapeLearner use cases, demonstrating partial shape querying of an elephant head and trunk (a), part-based morphing between a horse and an elephant (b) and synthesis of a new creature (c).

Table 4: Comparison with Huang et al. (2014), DR, and Shen et al. (2012) in precision.

Class	DR	Shen	Huang	Ours
airplane 	40.2%	56.1%	66.2%	65.8%
candelabra 	39.8%	56.1%	56.7%	68.5%
rifle 	49.6%	48.5%	62.2%	67.2%
fourleg 	52.3%	50.0%	67.2%	80.9%
vase 	51.7%	54.1%	63.1%	51.0%
human 	49.2%	47.7%	64.0%	94.1%
lamp 	67.8%	76.9%	89.3%	94.9%

Nouns are matched with object and part names in the database, while adjectives are matched with attributes as described below. Stop words and other unmatched words are ignored.

Attributes that can be matched include colors, angles, size, and length. All the objects are normalized according to their bounding boxes. Given the segmented parts of an object, the key line of a part is defined as the line connecting the middle point of the cut and the midpoint of its contour. The angle of a part is defined as the angular offset from a vertical line, i.e., the angle between the key line and a vertical line. The length of a component is defined as the length of the skeleton of its shape. To obtain the skeleton, we relied on an existing method [30]. Table 5 provides 7 example queries. The corresponding query results are shown in Figure 10.

Table 5: Example keyword queries.

Query	
Q1	horse with head down
Q2	horse with long tail
Q3	long tail of horse
Q4	cup with small handle
Q5	small cup handle
Q6	cup with black handle
Q7	pot with handle on the top

We observe that if the segmentation is correct, we obtain meaningful results, e.g. for Q1 and Q6. The quality of the cut of a part affects the results. In Q2 and Q3, the tail of the fifth horse is shortened due to inaccuracies in the cut of the tail. In Q4 and Q5, the system has located a handle at the top of the first cup, rather than on the right side.

7 Conclusion

We have introduced ShapeLearner, a novel system for organizing 2D shapes and their parts in a hierarchical structure that learns to



Figure 10: Keyword query results.

process new images and even new categories of objects. Our system starts with annotated seed data but then augments its knowledge by automatically processing new images and shapes. We derive a set of statistical constraints that we apply to correctly classify and segment an unknown input shape. ShapeLearner is able to transfer hypotheses based on visual similarity and relies on integer linear programming for joint inference. Our experiments show that, after seeding, ShapeLearner is able to collect valuable knowledge about shapes from uncategorized images. We additionally present several applications as use-cases of ShapeLearner, showcasing enhanced shape processing and manipulation.

In future work, we would like to extend ShapeLearner to focus not only on 2D shapes represented by their contours, but also to analyse the interior textures, for which we are exploring the use of deep convolutional neural networks. While a reduction to 2D shape contours reduces some of the noise, it results in a minimalist geometric representation. By going beyond it, ShapeLearner could thus also be extended to handle object shapes with severe shape occlusions.

We also plan to extend ShapeLearner to cover a wider range of semantic relationships and integrate it more tightly with the growing ecosystem of large-scale resources centered around the WordNet taxonomy, including ImageNet [9], YAGO [15], and UWN [8].

Finally, we are in the process of extending the seed data to cover many new categories, including medical data on bones and organs. We are also investigating crowdsourcing techniques to harvest a very broad range of categories. Initial experiments indicate that novices can fairly quickly learn how to mark parts of an object's shape. Thus, crowdsourcing techniques could enable us to quickly grow ShapeLearner's knowledge to cover thousands of categories of objects.

Acknowledgments

This project was sponsored by National 973 Program (No. 2015CB352500), National Natural Science Foundation of China (No. 61503217), Shandong Provincial Natural Science Foundation of China (No. ZR2014FP002), and The Fundamental Research Funds of Shandong University (No. 2014TB005, 2014JC001). Gerard de Melo's research is supported by China 973 Program Grants 2011CBA00300, 2011CBA00301, and NSFC Grants 61033001, 61361136003, 61550110504.

REFERENCES

- [1] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen, 'icoseg: Interactive co-segmentation with intelligent scribble guidance', in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pp. 3169–3176, (2010).
- [2] Serge Belongie, Jitendra Malik, and Jan Puzicha, 'Shape matching and object recognition using shape contexts', *IEEE Trans. Pat. Ana. & Mach. Int.*, **24**, 509–522, (2001).
- [3] Irving Biederman, 'Recognition-by-components: A theory of human image understanding', *Psychological Review*, **94**, 115–147, (1987).
- [4] Thomas O. Binford, 'Visual perception by computer', in *Proc. IEEE Conf. on Systems and Control*, (1971).
- [5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu, 'ShapeNet: An Information-Rich 3D Model Repository', Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, (2015).
- [6] Na Chen, Qian-Yi Zhou, and Viktor Prasanna, 'Understanding web images by object relation network', in *Proc. WWW*, pp. 291–300, (2012).
- [7] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta, 'NEIL: extracting visual knowledge from web data', in *Proc. ICCB*, pp. 1409–1416, (2013).
- [8] Gerard de Melo and Gerhard Weikum, 'Towards a universal wordnet by learning from combined evidence', in *Proc. CIKM 2009*, (2009).
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li, 'Imagenet: A large-scale hierarchical image database', in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255, (2009).
- [10] Santosh Divvala, Ali Farhadi, and Carlos Guestrin, 'Learning everything about anything: Webly-supervised visual concept learning', in *CVPR*, (2014).
- [11] Christiane Fellbaum, *WordNet: An Electronic Lexical Database*, Bradford Books, 1998.
- [12] Tong Ge, Yafang Wang, Gerard de Melo, Zengguang Hao, Andrei Sharf, and Baoquan Chen, 'Shapeexplorer: Querying and exploring shapes using visual knowledge', in *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016*, pp. 648–651, (2016).
- [13] Matthieu Guillaumin and Vittorio Ferrari, 'Large-scale knowledge transfer for object localization in imagenet.', in *Proc. CVPR*, pp. 3202–3209, (2012).
- [14] Bharath Hariharan, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik, 'Hypercolumns for object segmentation and fine-grained localization', in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 447–456, (2015).
- [15] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum, 'YAGO2: Exploring and querying world knowledge in time, space, context, and many languages', in *Proc. WWW 2011*, (2011).
- [16] Qi-Xing Huang, Vladlen Koltun, and Leonidas J. Guibas, 'Joint shape segmentation with linear programming', *ACM Trans. Graph.*, **30**(6), 125, (2011).
- [17] Zhe Huang, Hongbo Fu, and Rynson W. H. Lau, 'Data-driven segmentation and labeling of freehand sketches', *ACM Trans. on Graphics*, (2014).
- [18] Hongwen Kang, Martial Hebert, and Takeo Kanade, 'Discovering object instances from scenes of daily living', in *Proc. ICCB*, (2011).
- [19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei, 'Visual genome: Connecting language and vision using crowdsourced dense image annotations', (2016).
- [20] Haibin Ling and David W. Jacobs, 'Shape classification using the inner-distance', *IEEE Trans. Pat. Ana. & Mach. Int.*, (2007).
- [21] Lei Luo, Chunhua Shen, Xinwang Liu, and Chunyuan Zhang, 'A computational model of the short-cut rule for 2d shape decomposition', *CoRR*, **abs/1409.2104**, (2014).
- [22] Tomasz Malisiewicz and Alexei A. Efros, 'Beyond categories: The visual memex model for reasoning about object relationships', in *NIPS*, (2009).
- [23] David Marr, 'Early processing of visual information', *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, **275**(942), 483–519, (1976).
- [24] Genevieve Patterson, Chen Xu, Hang Su, and James Hays, 'The sun attribute database: Beyond categories for deeper scene understanding', *Int. J. Comp. Vis.*, 59–81, (2014).
- [25] Zhou Ren, Junsong Yuan, Chunyuan Li, and Wenyu Liu, 'Minimum near-convex decomposition for robust shape representation.', in *Proc. ICCB*, pp. 303–310, (2011).
- [26] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "'grabcut": Interactive foreground extraction using iterated graph cuts', in *Proc. SIGGRAPH*, pp. 309–314, (2004).
- [27] Olga Russakovsky, Jia Deng, et al., 'ImageNet Large Scale Visual Recognition Challenge', *Int. J. Comp. Vis.*, (2015).
- [28] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman, 'LabelMe: A database and web-based tool for image annotation', *Int. J. Comp. Vis.*, (1-3), 157–173, (2008).
- [29] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu, 'Structure recovery by part assembly', *ACM Trans. Graph.*, **31**(6), 180, (2012).
- [30] Wei Shen, Yan Wang, Xiang Bai, Hongyuan Wang, and Longin Jan Latecki, 'Shape clustering: Common structure discovery', *Pattern Recognition*, **46**(2), 539–550, (2013).
- [31] K. Siddiqi, A. Shokoufandeh, S. J. Dickenson, and S. W. Zucker, 'Shock graphs and shape matching', in *Sixth International Conference on Computer Vision*, pp. 222–229, (1998).
- [32] Yang Song, Ming Zhao, Jay Yagnik, and Xiaoyun Wu, 'Taxonomic classification for web-based videos.', in *Proc. CVPR*, pp. 871–878, (2010).
- [33] Niket Tandon, Gerard de Melo, Fabian Suchanek, and Gerhard Weikum, 'WebChild: Harvesting and organizing commonsense knowledge from the web', in *Proc. WSDM 2014*, (2014).
- [34] Bart Thomee, Benjamin Elizalde, David A. Shamma, Karl Ni, Gerald Friedland, Douglas Poland, Damian Borth, and Li-Jia Li, 'YFCC100M: The new data in multimedia research', *Commun. ACM*, **59**(2), 64–73, (January 2016).
- [35] S. Vicente, C. Rother, and V. Kolmogorov, 'Object cosegmentation', in *Proc. CVPR*, (2011).
- [36] Peng Wang, Xiaohui Shen, Zhe L. Lin, Scott Cohen, Brian L. Price, and Alan L. Yuille, 'Joint object and part segmentation using deep learned potentials', in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 1573–1581, (2015).
- [37] Walter Wohlkinger, Aitor Aldoma, Radu Bogdan Rusu, and Markus Vincze, '3dnet: Large-scale object class recognition from CAD models', in *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*, pp. 5384–5391, (2012).
- [38] Shiliang Zhang, Qi Tian, Gang Hua, Qingming Huang, and Wen Gao, 'Generating Descriptive Visual Words and Visual Phrases for Large-Scale Image Applications', *IEEE Trans. on Image Processing*, (2011).
- [39] Shiliang Zhang, Qi Tian, Gang Hua, Qingming Huang, and Wen Gao, 'ObjectPatchNet: Towards scalable and semantic image annotation and retrieval', *Comput. Vis. Image Underst.*, 16–29, (2014).
- [40] Xinming Zhang, Zheng-Jun Zha, and Changsheng Xu, 'Learning "verb-object" concepts for semantic image annotation.', in *ACM Multimedia*, pp. 1077–1080, (2011).

Observation-Based Multi-Agent Planning with Communication

Luca Gasparini and Timothy J. Norman and Martin J. Kollingbaum¹

Abstract. Models of decentralized online planning vary in the information that individual agents use to make local action decisions. Some models consider only local observations, eschewing coordination through communication. Others use communication to ensure that all agents are aware of the action decisions of others, but assume costless and delay-free communication. In this paper, we propose a model of online planning (OB-MAP) that uses estimates of the value of communicating to manage coordination through communication as costs vary. We compare this approach to existing models in widely employed benchmark problems, demonstrating that OB-MAP performs significantly better in many scenarios regardless of varying (including infinite) cost of communication.

1 Introduction

Decentralized planning problems are often modelled as Decentralized Partially Observable Markov Decision Processes (Dec-POMDPs), where multiple agents, each with a local view of the environment, must coordinate their actions in a decentralized fashion in order to optimize some reward [1]. Goldman and Zilberstein [6] have demonstrated, however, that even approximately solving a Dec-POMDP is intractable. One of the reasons for this complexity is that the number of possible joint-histories grows doubly exponentially with the horizon. In order to address this problem, a number of online planning algorithms [4, 5, 13, 15] have been proposed, which interleave planning and enactment. These algorithms heuristically estimate the long-term value of an action and use some of the information available at runtime in order to make planning more tractable. The majority of existing algorithms (such as [5, 13, 15]) plan in such a way that each agent always has full knowledge of what actions are being performed by its team-mates. This is referred to as *strict coordination*, and is often argued to be a necessary condition for effective planning in decentralized settings [15]. In order to guarantee strict coordination, however, agents must be limited in the extent to which they exploit local observations. The argument is as follows. If agents start with a common belief (a probability distribution) about the state of the environment, and use the same planning algorithm, they will agree on a common joint action to be performed. Since each agent potentially receives a different local observation at each time step, if they take into account these observations, their beliefs may diverge. Each agent will, therefore, plan for a different joint-action, and will have incorrect beliefs about the actions of its team-mates. As a result, strict coordination is not guaranteed.

In contrast to strict coordination models, Chechetka and Sycara [4] propose BaGa-S, which extends BaGa (Bayesian Games approxima-

tion algorithm) [5] in order to take advantage of local observations. BaGa-S has been shown to provide significant advantages over strict coordination models in some scenarios. These scenarios are, however, those in which local observations provide the best evidence for good local action decisions to maximise the reward. In contrast, we show that this approach performs significantly worse in domains that require a tighter coordination, supporting the strict coordination argument, albeit in an important class of problem domains.

Another important issue to consider when planning at runtime is whether and when agents should communicate their *local observations* as opposed to action decisions. By sharing observation histories, a coalition of agents can synchronize on a common belief, and take advantage of this information while maintaining strict coordination. Then, in algorithms that trade off strict coordination for a more opportunistic exploitation of local observations, communication can be used to re-synchronize agents' beliefs once coordination is lost.

In this paper, we argue that agents are faced with an important trade-off between maintaining (almost) strict coordination, and exploiting local observations to maximize their expected reward. We analyse this trade-off by comparing the performance of different algorithms in widely employed benchmark scenarios. We propose an algorithm, OB-MAP, that attempts to capture the best of both worlds. While OB-MAP does not guarantee strict coordination, we show experimentally that it performs at least as good as strict coordination algorithms, and is able to take advantage of local observations in scenarios that favour a more opportunistic planning approach. We also propose a heuristic that takes into account the value of communication in order to decide whether or not agents should communicate.

Before formalising OB-MAP, analysing its complexity, and evaluating its performance, in the following section we provide necessary technical background. We present a précis of the Dec-POMDP approach to multi-agent planning, and then give details of the two on-line planning models that we use to represent the state-of-the-art in on-line planning algorithms.

2 Background

A Dec-POMDP [1] is a tuple, $\langle I, S, b^0, \{A_i\}, P, \{\Omega_i\}, O, R \rangle$ where:

- I is a set of agents, and S is the set of states;
- b^0 is an initial belief state, *i.e.* a probability distribution over possible initial states;
- A_i is a finite set of actions available to agent i and $\vec{a} = \langle a_1, \dots, a_n \rangle$ is a joint-action consisting of one action for each agent;
- $P(s_j | s_i, \vec{a})$ represents the probability that taking joint-action \vec{a} in state s_i will result in a transition to state s_j ;

¹ Department of Computing Science, University of Aberdeen, Aberdeen, UK, l.gasparini@abdn.ac.uk, tnorman@acm.org, m.j.kollingbaum@abdn.ac.uk

- Ω_i is a finite set of local observations o_i available to agent i and $\vec{\Omega}$ is the set of joint observations \vec{o} consisting of one local observation for each agent;
- $O(\vec{o} \mid s_j, \vec{a})$ specifies the probability of observing \vec{o} when performing a joint-action \vec{a} that leads to a state s_j ;
- $R : S \times \vec{A} \rightarrow \mathbb{R}$ is a reward function, and $R(s_i, \vec{a})$ specifies the reward obtained by performing \vec{a} in s_i .

We define a local history h_i for agent i up to time t as a sequence of interleaved local actions and observations. $h_i = (a_i^0, o_i^1, a_i^1, \dots, o_i^t)$ and a joint-history as a tuple \vec{h} consisting of one local history for each agent $\vec{h} = \langle h_0, \dots, h_n \rangle$. A belief state at time t , $b^t : S \rightarrow \mathbb{R}$, is a function that represents the probability that the system is in each state. Given a belief b^t at time t , the belief state at time $t+1$ after the agents have executed joint-action \vec{a}_i and received joint-observation \vec{o}_j can be computed as follows:

$$b^{t+1}(s) = \frac{\sum_{s'} b^t(s') \cdot P(s \mid s', \vec{a}_i) O(\vec{o}_j \mid s, \vec{a}_i)}{\sum_{s', s''} b^t(s') \cdot P(s'' \mid s', \vec{a}_i) O(\vec{o}_j \mid s'', \vec{a}_i)} \quad (1)$$

We denote the updated belief state as $p(\ast \mid b^t, \vec{a}_i, \vec{o}_j)$. The denominator corresponds to the probability of observing \vec{o}_j after performing \vec{a}_i from belief b^t . Where we are interested in the expected joint observation, we also use the explicit notation $p(\vec{o}_j \mid b^t, \vec{a}_i)$. A local policy for agent i is a mapping from local histories to actions. A local policy $u_i^t \in U_i^t$ for an horizon length t can be represented as a tree where each node represents an action, and each edge of the tree an observation. We denote with $a_{u_i^t}$ the local action prescribed by a local policy u_i^t and with $u_i^t(o_i)$ the sub-policy (for horizon length $t-1$) that should be followed after receiving an observation o_i . A joint-policy $\vec{u}^t \in \vec{U}^t$ is defined as consisting of one local policy for each agent. We denote with $\vec{a}_{\vec{u}^t}$ the joint action prescribed by policy \vec{u}^t and with $\vec{u}^t(\vec{o})$ the joint sub-policies that the agents follow after receiving joint observation \vec{o} . The value of executing a joint-policy \vec{u} , from a state s with t steps to go can be computed recursively as follows:

$$V(\vec{u}^t, s_i) = R(s_i, \vec{a}_{\vec{u}^t}) + \sum_{s_j, \vec{o}} P(s_j \mid s_i, \vec{a}_{\vec{u}^t}) O(\vec{o} \mid s_j, \vec{a}_{\vec{u}^t}) V(\vec{u}(\vec{o}), s_j) \quad (2)$$

Given a belief state, solving a Dec-POMDP means finding a joint-policy \vec{q} that maximizes $\sum_s b(s) V(\vec{q}, s)$.

Goldman and Zilberstein [6] demonstrated that even approximately solving a Dec-POMDP is intractable (NEXP-COMplete). A great deal of research on offline planning for Dec-POMDPs, therefore, focuses on tractable approximate algorithms that do not provide a guarantee on solution quality. A different body of work has explored heuristic-based online planning for Dec-POMDPs. These algorithms interleave planning and execution, and use heuristics to make decisions on what actions to perform next. The long-term expected value of executing joint-action \vec{a}_j from a belief state b_i is $Q(b_i, \vec{a}_j)$. This Q function can be computed, for example, by performing an l -step lookahead and assuming that the state of the system becomes fully observable after the l -th step.

Given our aim is to propose a model of on-line planning that effectively balances the trade-off between maintaining coordination through communication and exploiting local observations, we choose as our comparators MAOP-COMM [15] and BaGa-S [4].

Wu *et al.* [15] propose the MAOP-COMM algorithm for online planning in Dec-POMDPs with communication. At each step, each

agent maintains a pool of possible joint-histories H (one local history per agent), each associated with a probability and a joint belief; that is, the belief that would be obtained by an hypothetical agent that has complete knowledge of all the agents' histories. Given a joint-history \vec{h}^t , we denote $b_{\vec{h}^t}$ to be the belief associated with \vec{h}^t , and $p(\vec{h})$ to be its probability.

Each agent approximates a one-step lookahead policy $\vec{\pi}$ that maps, for each agent, a local history to an action. We denote π_i to be the local component of $\vec{\pi}$ and $\pi_i(h_i)$ to be the action associated with history h_i . Given a joint-history $\vec{h} = \langle h_0, \dots, h_n \rangle$ the joint-action executed will be $\vec{\pi}(\vec{h}) = \langle \pi_0(h_0), \dots, \pi_n(h_n) \rangle$. The objective is to find a policy $\vec{\pi}$ that optimizes the following value function:

$$V(\vec{\pi}) = \sum_{\vec{h} \in H} p(\vec{h}) Q(b_{\vec{h}}, \vec{\pi}(\vec{h})) \quad (3)$$

In order to efficiently find a policy, it is initialized randomly, and then each agent improves its local policy by assuming the policies of other agents are fixed. Improvement terminates when an equilibrium among the local policies is found; *i.e.* when no agent can improve its own policy. After each action, all the histories in the pool are updated by considering the corresponding action and every possible observation, and all the joint beliefs are updated. Coordination is guaranteed because each agent will maintain the same set of possible histories, and they use the same seed for a pseudo-random generator to initialize the policies. In MAOP-COMM, agents decide to communicate if they receive an observation that is inconsistent with their current history pool. Formally, given the current history pool H , a local observation o_i , and a small number ϵ , an agent decides to communicate if and only if:

$$\max_{\vec{h} \in H, \vec{o}_{-i}} \left(\sum_{s' \in S} O(\langle o_i, \vec{o}_{-i} \rangle \mid s', \vec{a}) \sum_{s \in S} P(s \mid \vec{a}, s') b_{\vec{h}}(s) \right) < \epsilon \quad (4)$$

The rationale for this is that agents should communicate when they receive an unexpected observation.

While MAOP-COMM guarantees coordination, we argue that this comes at a cost. In order to ensure that agents reach the same equilibrium, each agent i must consider all their own possible previous histories, whereas only one history has been observed. This information, in fact, is not generally available to the other agents. Moreover, by taking into account an observation o_i , an agent is often able to infer additional knowledge about the probability of other agents' histories. Even though the current action of an agent depends on its local history, the fact that MAOP finds equilibria among all possible histories, and uses only information that is available to all agents, it results in policies that cannot take full advantage of local observations.

BaGa-S (Subjective Bayesian Game approximation algorithm) [4] attempts to make use of local observations in a more opportunistic way. In BaGa-S agents consider only histories that are consistent with their local observations. Agent i estimates the best action for the other agents in each history $a_{-i}^*(\vec{h})$ (Equation 5) and then finds its best response a_i^* (Equation 6).

$$a_{-i}^*(\vec{h}) = \arg \max_{a_{-i} \in A_{-i}} \left(\max_{a_i} Q(b_{\vec{h}}, a_i) \right) \quad (5)$$

$$a_i^* = \arg \max_{a_i \in A_i} \sum_{\vec{h}} p(\vec{h}) Q(b_{\vec{h}}, (a_i, a_{-i}^*(\vec{h}))) \quad (6)$$

After executing an action and receiving an observation, each agent updates its belief pool by considering, from each possible joint belief, the estimated action of other agents and all the possible joint-observations that are compatible with its local observation. In order

to limit the exponential growth of the belief history, BaGa-S uses weighted k -means clustering to find, after each update, a fixed number of beliefs that represent the distribution over possible histories. k -means clustering divides joint beliefs into k clusters and maintains only the centre of each cluster as a representation of it. The clusters are found such that the sum of the distance of each belief from the corresponding centre is minimized. The centre of each cluster is a belief where the probability of each state is the weighted average among the probabilities of that state given the beliefs in the cluster, with the weights being the probability of each history. Given two possible beliefs b_1 , and b_2 , held by an agent i , a distance measure for joint beliefs can be defined as:

$$d(b_1, b_2) = \sqrt{\sum_s (b_1(s) - b_2(s))^2 \cdot p(h_2)} \quad (7)$$

Intuitively, this estimates the expected loss of information obtained by merging b_2 with b_1 . We multiply the distance only by $p(h_2)$ because this merging represents a loss of information only if the true belief is b_2 .

Note that, each agent takes into account its local observation in updating the belief pool, and so the pools maintained by different agents may diverge at runtime. This, in turn, will lead to different policies being computed for each agent and a further divergence in the belief pools. As pointed out by Wu *et al.* [15], this might lead to arbitrarily bad outcomes. On the other hand, taking into consideration local observations in computing a plan might prove beneficial in scenarios where only loose coordination is necessary.

3 OB-MAP

We now present Observation Based Multi-Agent Planning (OB-MAP), an online planning algorithm that provides a good balance between opportunistic exploitation of information and the maintenance of a certain degree of coordination. We argue that BaGa-S fails to do this because, when an agent i is planning, it doesn't consider the fact that agents other than i also have only a partial view of the environment. In Equation 5, for example, they assume that other agents are able to observe the current joint belief. Moreover, the clustering of histories only takes into account the distance among joint beliefs. It has been demonstrated by Oliehoek *et al.* [12] that in order for the clustering of joint-histories to be lossless, one should merge only histories that are equivalent both in terms of joint beliefs, and in terms of the probability distribution over joint histories held by other agents. Merging only equivalent histories only allows for limited reduction in the size of the joint-histories pool, especially in large scenarios where the number of possible histories grows very quickly with the execution horizon. Our aim is to define a distance metric that approximates the lossless criterion and to use it in standard clustering algorithms to perform a more aggressive clustering while still minimizing the loss of information.

In order to do that, while updating the belief pool we also keep track of the local belief of each agent in each history. Formally we define a belief-node n^k as a tuple:

$$n^k = \langle \vec{h}^k, b_0^k, b_1^k, \dots, b_n^k, p^k \rangle \quad (8)$$

where \vec{h}^k is a joint-history, b_0^k is the joint belief associated with the joint history, b_i^k with $1 \leq i \leq n$ is the local belief for agent i and p^k is the probability associated with the node. Given a local belief, b_i^t , for agent i at time t , the joint-action \vec{a}^t and the local observation o_i^t ,

the local belief for i at time $t + 1$ can be computed as follows:

$$b_i^{t+1}(s) = p(s|b_i^t, \vec{a}^t, o_i^t) = \frac{\sum_{s', o_{-i}} b^t(s') p(s|s', \vec{a}^t) O(\langle o_i^t, o_{-i} \rangle | s, \vec{a}^t)}{\sum_{s', s'', o_{-i}} b^t(s') p(s''|s', \vec{a}^t) O(\langle o_i^t, o_{-i} \rangle | s'', \vec{a}^t)} \quad (9)$$

where o_{-i} is a tuple consisting of local observations for all agents other than i . The update procedure is similar to the one for a joint belief, but considers all possible joint-observations that have o_i as a local component. We refer to the updated local belief as $p(*|b_i^t, \vec{a}^t, o_i^t)$. Note that, while the update considers only the local component of an observation, the complete joint-action is needed.

3.1 Planning

In common with BaGa-S, when planning, each agent estimates the local action that will be taken by the other agents in each joint history and finds the best response. In doing so, however, it takes into account the fact that if a set of joint histories is associated with the same local history for agent j , the agent will not be able to distinguish among them, and will choose the same action for all of them. A local policy π is defined in the same way as in MAOP-COMM; *i.e.* a mapping from local histories to local actions. Let $h(n^k, i)$ denote the local history for agent i in the node n^k . Given the current set of belief nodes N_i held by an agent i , and a history h_j for agent j different from i , agent i estimates the local action performed by j by finding the joint action that maximizes the following:

$$\pi_j(h_j) = \arg \max_{a_j \in A_j} \left(\max_{\substack{n^k \in N_i \text{ s.t.} \\ h(n^k, j) = h_j}} \sum Q(b_0^k, \langle a_j, \vec{a}_{-j} \rangle) \cdot p^k \right) \quad (10)$$

For each agent, j , we consider together all the nodes that are associated with the same local history for j . These nodes, therefore, represent joint beliefs that are indistinguishable from j 's perspective. We assume that j will select, for all these nodes, the action that maximises the expected reward over all the associated joint beliefs.

After an agent has estimated all the actions of other agents, it finds the local action that maximizes its expected reward over all possible nodes. Suppose we take π_{-i} to denote the joint policy that maps each node n^k to a joint action that is left unspecified for agent i and that associates the action $\pi_j(h(n^k, j))$ to each agent j other than i . Now, we can estimate the best action for agent i , thus:

$$a_i^* = \arg \max_{a_i \in A_i} \sum_{n^k \in N_i} q(b_0^k, \langle a_i, \pi_{-i}(n^k) \rangle) \cdot p^k \quad (11)$$

The expected value of local action a_i^* corresponds to $\sum_{n^k \in N_i} q(b_0^k, \langle a_i^*, \pi_{-i}(n^k) \rangle) \cdot p^k$. When describing the reasoning of agent i we will use $\pi_i(n^k)$ to denote the policy that assigns a_i^* to every node n^k and π to denote the joint policy that assigns to each node n^k the joint action $\langle a_i^*, \pi_{-i}(n^k) \rangle$.

There is an important point of comparison to note here regarding the OB-MAP and BaGa-S models. If there are only two agents, and assuming the same clustering technique is used by them both, the plan computed by our algorithm will be identical to that computed by BaGa-S. Consider, for example, the point of view of agent 1. Since agent 1 only maintains beliefs that are compatible with its local history, each belief node in the pool must be associated with

Algorithm 1 Belief propagation

Input: N_i^t, π_{-i}, o_i
Output: N_i^{t+1}

- 1: $N_i^{t+1} = \emptyset$
- 2: **for all** $n^k \in N_i^t$ **do**
- 3: $\vec{a} = \pi(n^k)$
- 4: **for all** $o_{-i} \in \Omega_{-i}$ **do**
- 5: $\vec{o} = \langle o_i, o_{-i} \rangle$
- 6: $b_0(*) = p(*|b_0^k, \vec{a}, \vec{o})$
- 7: $p = p(\vec{o}|b_0^k, \vec{a})$
- 8: **for all** $0 \leq j \leq n$ **do**
- 9: $b_j(*) = p(*|b_j^k, \vec{a}, \vec{o}[j])$
- 10: $h_j = (h_j^k, \pi_j(n^k), \vec{o}[j])$
- 11: **end for**
- 12: $\vec{h} = \langle h_1, \dots, h_n \rangle$
- 13: $N_i^{t+1} = N_i^{t+1} \cup \langle \vec{h}, b_0, b_1, \dots, b_n, p \rangle$
- 14: **end for**
- 15: **end for**

a different history for agent 2, otherwise the two nodes would be equivalent. When estimating the action of agent 2, each belief-node will be considered separately and Equations 10 and 11 (OB-MAP) are equivalent to Equations 5 and 6 (BaGa-S). When there are more than 2 agents, two beliefs in the pool might have the same history for agent 2, but a different one for, for example, agent 3.

After an action is taken and an observation received, the nodes in the belief pool need to be propagated. We consider each node in the pool, with the estimated joint action and, for each joint-observation compatible with the local observation received, we update the joint-history and all the joint and local beliefs. We refer to the set of all possible joint observations as Ω_{-i} , with the i -th component left unspecified, and we refer to the local component of a joint observation \vec{o} associated with agent j as $\vec{o}[j]$. Algorithm 1 specifies how this pool of belief nodes is updated.

3.2 Clustering

After propagating the beliefs, we perform clustering in order to maintain a bounded number of beliefs. The distance metric for belief nodes is defined as:

$$d(n^k, n^l) = \sqrt{\sum_s \left(\max_{0 \leq i \leq n} (b_i^k(s) - b_i^l(s))^2 \right)} \cdot p^l \quad (12)$$

This captures the idea that, for each state, we take the maximum distance among all pairs of corresponding (joint or local) beliefs. Moreover, instead of using weighted k -means clustering, we use a modified k -medoid clustering. This algorithm partitions N_i into k clusters and finds, for each cluster $C_j = \{n^k, \dots\}$, the node \bar{n}^{C_j} that minimizes the sum of the distances of each other element of the cluster from \bar{n}^{C_j} . Formally:

$$\bar{n}^{C_j} = \arg \min_{n^k \in C_j} \sum_{n^l \in N^k} d(n^l, n^k) \quad (13)$$

We refer to node \bar{n}^{C_j} as the medoid of the cluster. Since the medoid is an actual data-point, k -medoids is more robust to outliers and noise, which are important to consider in planning problems. For each cluster, we retain the one node at the medoid of the cluster, defined as follows:

$$n_*^{C_j} = \langle \vec{h}^{C_j}, \bar{b}_0, \dots, \bar{b}_n, p_*^{C_j} \rangle \quad (14)$$

where:

- $\bar{b}_0, \dots, \bar{b}_n$ are the joint and local beliefs of the medoid node \bar{n}^{C_j} .
- $p_*^{C_j} = \sum_{n^k \in C_j} p^k$ is the sum of all the probabilities of the nodes in the cluster.
- $\vec{h}^{C_j} = \langle h_1^{C_j}, \dots, h_n^{C_j} \rangle$ consists of, for each agent, the local history that appears with highest probability in the cluster. Formally, if H_i denotes the set of possible local histories for agent i :

$$h_i^{C_j} = \arg \max_{h_i \in H_i} \sum_{\substack{n^k \in C_j \text{ s.t.} \\ h(n^k, i) = h_i}} p^k \quad (15)$$

3.3 Communication Heuristics

We consider agents that can communicate in order to share their local histories and synchronize on a common joint belief. This enables agents to obtain more precise information about the current state of the environment and to restore coordination when this is lost due to misaligned belief pools. We assume that communication comes at a cost (a negative reward), and we propose a heuristic technique to adaptively make decisions on whether or not to communicate, based on the current level of uncertainty and the expected value obtained from communication. Given a communication cost R_c , and the current belief node pool N_i , agent i can estimate the expected value obtained by communicating as follows:

$$V_c = -R_c + \sum_{n^k \in N_i} p^k \cdot \max_{\vec{a} \in \vec{A}} Q(b_0^k, \vec{a}) \quad (16)$$

Informally, since after communicating each agent will have full knowledge about the current joint belief, we assume that the agents can choose a different joint-action for each joint belief. Each agent finds the best action for each belief-node and averages over their expected values. The agent then subtracts the cost of communication and, if the resulting V_c is greater than the estimated value without communication, it chooses to communicate. Note that, since the action chosen for other agents when updating the belief pool are only an estimate, and because of the clustering, the actual joint belief might not be present in the belief-node pool. Moreover, from the point of view of agent i , the heuristic does not take into account the value obtained by other agents when they receive i 's local history, but only the value obtained by i when it receives other agent's observations. We will demonstrate experimentally that this heuristic works well in practice and provides an efficient way to estimate the added value of communication.

Algorithm 2 specifies the main function of the OB-MAP planner. The belief-node pool is initialized with a single node corresponding to the empty history and that is assigned the initial beliefs b^0 of all the agents (Line 1). The **computePolicy** function (Line 3) estimates the actions of other agents and finds the best response according to Equations 10 and 11. **ComputeCommValue** finds the expected value after communication by applying Equation 16. If this value is higher than the estimated value without communication the agent will communicate with its team-mates in order to **sync** their true histories and find a common joint belief. If communication occurs, the agents must recompute their policies taking into account the true joint belief (Lines 4-9). After executing their part of the policy (the local action found as best response) and receiving an observation, the agents will propagate the current belief-nodes and use k -medoid clustering to find k belief nodes that best represent all the possible histories.

Algorithm 2 Main OB-MAP execution function

Input: b^0, R_c, k

- 1: $N = \{(\cdot, b^0, b^0, \dots, b^0, 1)\}$
- 2: **for** $t = 0$ to T **do**
- 3: $\pi = \text{computePolicy}(N)$
- 4: $V_C = \text{computeCommValue}(N, R_c)$
- 5: **if** $V_C \geq V(\pi)$ **then**
- 6: $\langle b^t, h^t \rangle = \text{sync}()$
- 7: $N = \{\langle h^t, b^t, b^t, \dots, b^t, 1 \rangle\}$
- 8: $\pi = \text{computePolicy}(N)$
- 9: **end if**
- 10: execute best local response.
- 11: $o = \text{received observation}$
- 12: $N = \text{propagateBeliefs}(N, \pi, o)$
- 13: $N = \text{cluster}(N, k)$
- 14: **end for**

3.4 OB-MAP Complexity

Before presenting an empirical analysis of our model, we analyse its runtime complexity for both belief propagation and belief node clustering. At each step, the size of the set of beliefs N is bounded by k , the number of clusters specified for the k -medoids algorithm. When expanding the beliefs we take each of these k beliefs, consider the action given by the policy $\pi(n^k)$ and, for each observation that is compatible with the observed o_i , we update the joint belief and all the local beliefs. This gives a time complexity of

$$O(|\Omega_{-i}| \cdot k \cdot |I| \cdot |S|^2)$$

where S^2 is the belief update, and $|\Omega_{-i}| = \prod_{j \in I \setminus \{i\}} |\Omega_j|$.

For k -medoids, we use the Partitioning Around Medoid (PAM) [8] algorithm, which performs $O(I \cdot (k \cdot (N - k)^2))$ comparisons, where I is the number of iterations performed by the algorithm, N is the initial number of beliefs, and k the number of clusters. Since, as discussed above, the number of data points is $|\Omega_{-i}| \cdot k$, and each comparison (Equation 12) takes $|I| \cdot |S|$, the complexity of the clustering phase is:

$$O(I \cdot |\Omega_{-i}|^2 \cdot k^3 \cdot |S| \cdot |I|)$$

The complexity of the process of computing a policy will depend on the heuristics employed, but, in general, we evaluate $|\vec{A}| \cdot k$ actions at each step. For the large majority of problems, runtime is dominated by the clustering phase (which we confirmed experimentally), and PAM is not the most efficient algorithm that could be employed. CLARANS, for example, is an efficient clustering model designed for mining large data sets. Although our problem is to solve a large number of small clustering problems, it was analysed to be “a few times faster than PAM” [11] for small data sets, and hence could be employed to further optimise OB-MAP.

4 Evaluation

In order to effectively evaluate the OB-MAP model, we consider a number of widely-employed benchmark problems. We pitch OB-MAP against two state-of-the-art algorithms: one that uses communication to synchronise agents’ beliefs to maintain coordination (MAOP-COMM), and one that exploits only local observations (BaGa-S). Furthermore, in order to provide a fair comparison, in each benchmark we assess relative performance where communication is cost-free (OB-MAP versus MAOP-COMM under the same assumptions) and where no communication is permitted (OB-MAP versus

BaGa-S under the same assumptions, and also MAOP-COMM with no communication).

In all the benchmark problems, we used a multi-step lookahead MDP-based Q heuristic, Q_{MDP} . The heuristic is defined as follows, where $V_{MDP}(s)$ is the expected value of the optimal policy for the underlying MDP, starting from state s :

$$Q_{MDP}(b, \vec{a}_j, l) = \begin{cases} \sum_{s, s'} b(s)(R(s, \vec{a}) + P(s'|s, \vec{a}_j)V_{MDP}(s')) & \text{if } l \leq 1 \\ \begin{cases} \sum_{s'} [b(s')R(s', \vec{a})] + \sum_{\vec{o}_k} [p(\vec{o}_k|b^t, \vec{a}_j)] \\ \max_{\vec{a}_l} \{Q(b_{\vec{a}_l, \vec{o}_k}, \vec{a}_l, l-1)\} \end{cases} & \text{if } l > 1 \end{cases} \quad (17)$$

Essentially the heuristic assumes full communication for l steps and full observability after that. A better approximation could be obtained by using a POMDP policy, which assumes full communication, but partial observability, over the whole problem horizon.

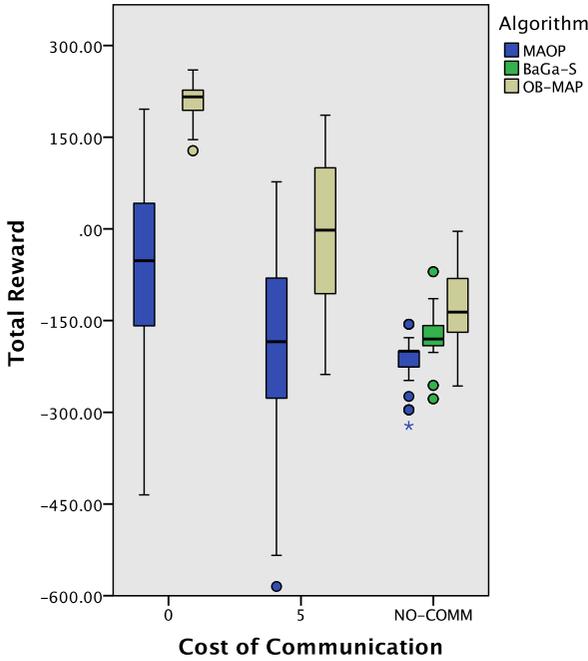
The benchmark problems that we used are:

- The decentralized tiger problem [10]. In this scenario, two agents are in a corridor facing two doors: “left” and “right”. Behind one door lies a hungry tiger and behind the other lies a treasure. Each agent can either open one of the doors or listen for the presence of the tiger. After each step, each agent receives a noisy observation about the position of the tiger. By listening, agents increase their probability of receiving the correct observation. The agents minimize their loss if they jointly open the door with the tiger and maximize their reward if they jointly open the treasure door. After either door is opened the problem is reset.
- The variant of 3x3 grid problem presented by Amato *et al.* [2]. In this problem two agents can move along 4 directions in a 3x3 grid world. Each agent only receives noisy observations about neighbouring walls. The objective of the agents is to meet in either the top-left or bottom-right cell.
- The stochastic Mars rover problem [3]. In this scenario, two agents must perform different experiments at certain research sites. Some of these sites may require multiple agents performing an experiment together in order to get the most scientific value, while other sites may require a specific tool to be used by a single agent. Positive rewards are given for successfully performing experiments at each site and the task is completed when all experiments have been conducted.

In order to verify our claim that the approximate planning model described by Equations 10 and 11 improves upon BaGa-S for scenarios with more than 2 agents, we compared our approach against standard BaGa-S and BaGa-S with our medoids-based clustering approach in the following scenarios:

- A 3 agent version of the decentralized tiger problem.
- A 3 agent version of the broadcast channel problem [7]. In this problem, 3 agents attempt to send messages over a shared channel. Each agent has a buffer of at most one message. The channel can deliver only one message at a time. Moreover the channel randomly switches between a functional and a non-functional state. If only one agent attempts to send a message, the channel is functioning, and the buffer of that agent has one message, the agents receive a reward of +1. If no message is delivered because the channel is not functioning, because of a collision, or because the agent trying to send a message has an empty buffer, the agents obtain a reward of $-s$, where s is the number of agents that attempted to send a message. At each step each agent receiving a noisy observation signalling whether there has been a collision, a successfully delivered message, or the state of the channel.

Figure 1. The 2-Agent Decentralized Tiger Scenario



For all the scenarios we used an horizon of 100 steps.

Table 1. The 2-Agent Decentralized Tiger Scenario

R_C	MAOP-COMM			OB-MAP			BaGa-S
	0	5	NO	0	5	NO	NO
R	-62.9 ± 138.2	-185.3 ± 136.6	-207.0 ± 30.7	207.3 ± 30.8	-6.7 ± 117.6	-128.8 ± 54.9	-174.1 ± 28.7
C	24.5 ± 4.1	24.5 ± 4.1	0 ± 0	36.3 ± 2.4	7.6 ± 3.3	0 ± 0	0 ± 0
T[ms]	0.65	0.65	0.80	0.53	0.57	0.61	1.07

Table 1 presents results for the 2-agent decentralized tiger scenario. The table summarises the average and standard deviation over 100 trials of the reward (**R**), the number of communication steps (**C**) and the average execution time per agent per step (**T[ms]**) for the three algorithms with varying cost of communication (R_C). For both BaGa-S and OB-MAP we maintained 20 clusters. Since BaGa-S does not consider communication, we report only the results in the absence of communication (NO). In Figure 1 we present the distribution of reward for each algorithm and for varying cost of communication. Notice that the frequency of communication does not change with the communication cost in MAOP-COMM. This is due to the fact that MAOP-COMM only allows us to specify whether agents can or cannot communicate, and communication cost is not taken into account. In order to simulate different communication costs we simply subtracted from the obtained reward the total cost of communication. Comparing the results of BaGa-S and MAOP (*i.e.* MAOP-COMM with no communication) we can see that this scenario favours a more opportunistic approach over guaranteed coordination. The OB-MAP planner has a better average than either of the other algorithms, and it makes better use of communication when available in compari-

son with MAOP-COMM. Since the results are not normally distributed, we tested them for significance using the Kruskal Wallis test with post-hoc analysis consisting of Bonferroni corrected Mann-Whitney tests. We obtained an asymptotic p-value of 0.000 both for the comparison with MAOP-COMM and with BaGa-S². The execution times for OB-MAP are comparable, but slightly lower than those for MAOP-COMM and almost half of those for BaGa-S.

Table 2. The 3x3 Grid Scenario

R_C	MAOP-COMM			OB-MAP			BaGa-S
	0	0.1	NO	0	0.1	NO	NO
R	25.4 ± 1.4	22.2 ± 1.8	15.6 ± 3.2	25.3 ± 1.2	21.7 ± 2.7	21.5 ± 3.9	6.88 ± 5.7
C	32.2 ± 4.6	32.2 ± 4.6	0 ± 0	68.3 ± 3.0	0.6 ± 0.7	0 ± 0	0 ± 0
T[ms]	0.77	0.77	72	7.09	90.0	71.19	86.45

Table 2 and Figure 2 summarize the results for the 3x3 Grid scenario. For this scenario we used the Q_{MDP} heuristic with lookahead equal to 1 and we set the number of clusters to 20 for both BaGa-S and OB-MAP. In this scenario MAOP-COMM performs better than BaGa-S. We believe that this is because this scenario requires tighter coordination among team-mates; see discussion of the strict coordination argument above. Our approach performs significantly better (asymptotic p-value of 0.000) than either MAOP-COMM or BaGa-S in the absence of communication. When communication is available, however, the results for OB-MAP are not significantly different from MAOP-COMM (asymptotic p-value of 0.508).

² The p-value of 0.000 denotes 0 to the precision available from the statistical analysis tool used.

Figure 2. The 3x3 Grid Scenario

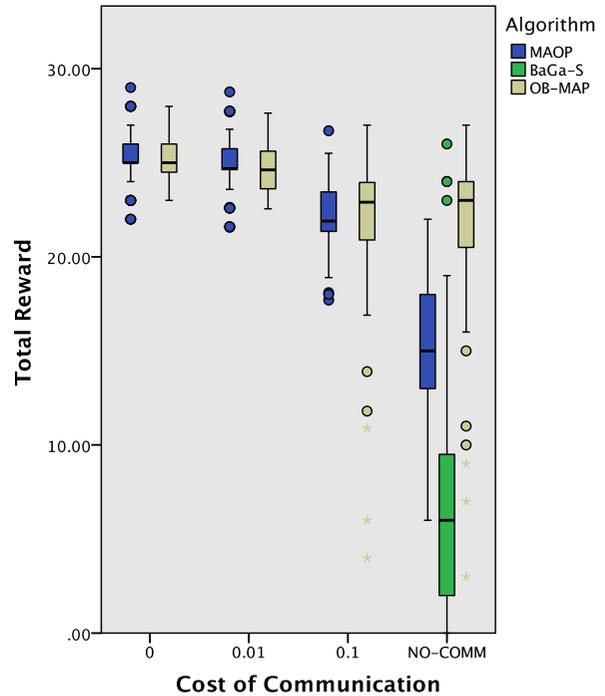


Table 3. The Stochastic Mars Rover Scenario

R_c	MAOP-COMM			OB-MAP			BaGa-S
	0	2	NO	0	2	NO	NO
R	150.9 ± 8.8	119.7 ± 10.2	43.5 ± 15.8	284.6 ± 12.01	222.6 ± 29.8	141.9 ± 61.9	110.0 62.1
C	15.5 ± 3.3	15.5 ± 3.3	0 ± 0	30.9 ± 3.2	15.9 ± 1.7	0 ± 0	0 ± 0
T[ms]	0.7	0.7	434.3	3.5	12.8	31.5	27.9

Table 3 and Figure 3 report results for the Stochastic Mars Rover Scenario. This represents another situation in which BaGa-S significantly outperforms MAOP-COMM in the absence of communication. OB-MAP, however, performs better than either MAOP (asymptotic p-value of 0.000) or BaGa-S (asymptotic p-value of 0.005) in the absence of communication. When communication is available, OB-MAP significantly outperforms MAOP-COMM (asymptotic p-value of 0.000) and adapts well to more costly communication. Note that, since all the scenarios discussed so far include only two agents, the differences between OB-MAP and BaGa-S are only due to the different clustering methods used.

In Table 4 we present the results for the comparison between BaGa-S and OB-MAP algorithms on the 3-agent version of the tiger scenario. We could not test MAOP-COMM in this scenario because the implementation of MAOP-COMM used (kindly provided by the authors) does not support scenarios with more than two agents. In order to separate the effects of the clustering algorithm used, and the different action selection mechanisms, we also analysed the behaviour of a version of BaGa-S that uses medoid clustering (**BaGa-S-medoids** in Table 5). In our experiments, BaGa-S with k -medoids agents always listened, without ever attempting to open any door. This is the reason why their reward is always -303 (where -1 is

Table 4. The 3-Agent Decentralized Tiger Scenario

R_c	OB-MAP			BaGa-S	BaGa-S-medoids
	0	5	NO	NO	NO
R	577.6 ± 58.0	200.3 ± 39.9	-267.5 ± 133.9	-299.6 ± 11.1	-303.0 ± 0
C	61.0 ± 2.5	34.0 ± 1.1	0 ± 0	0 ± 0	0 ± 0
T[ms]	7.16	10.29	62.3	157.6	97.5

the cost of one agent listening). BaGa-S agents deviate only slightly from this behaviour. OB-MAP behaves significantly better (asymptotic p-value of 0.000) than either BaGa-S or BaGa-S-medoids, while the results of BaGa-S with medoids are not different from those of BaGa-S (p-value of 1.000). We argue that this difference is due to the fact that agents using the BaGa-S algorithm fail to correctly estimate the other agents' actions, because they assume that the true joint belief is known to them.

In Table 5 and Figure 5 we report results for the BaGa-S and OB-MAP algorithms on the 3-agent Broadcast Scenario. In this scenario BaGa-S-medoids performs considerably better than standard BaGa-S (p value 0.000) or BaGa-S-medoids (p value 0.040). The significance of this difference between OB-MAP and BaGa-S-medoids confirms that our planning algorithm better captures scenarios where there are more than two agents.

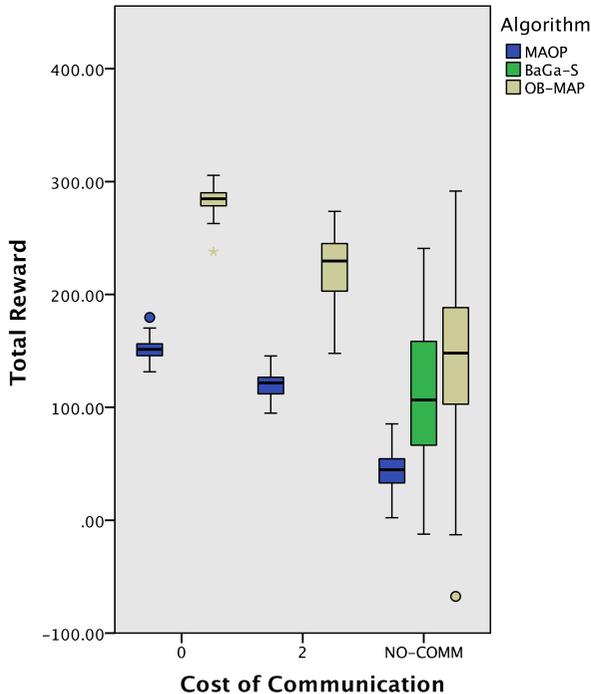
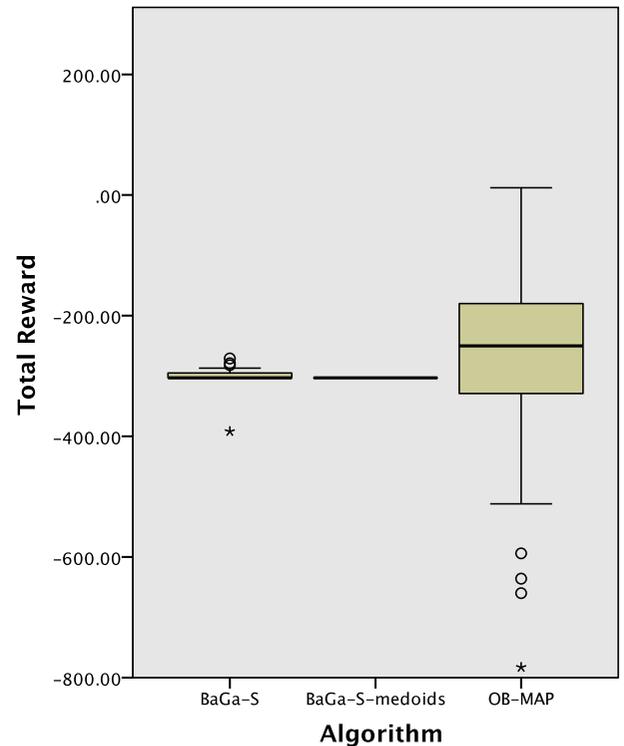
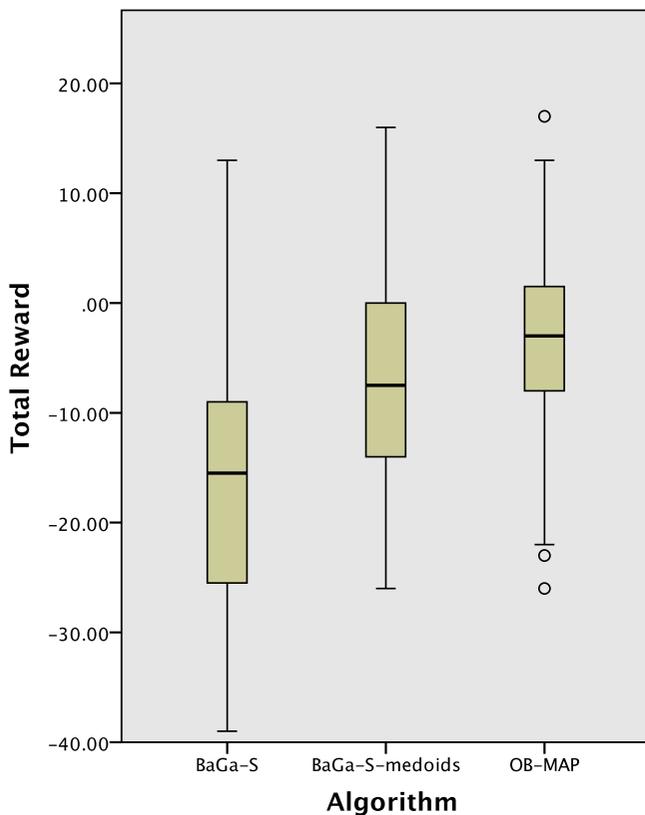
Figure 3. The Stochastic Mars Rover Scenario**Figure 4.** The 3-Agent Decentralized Tiger Scenario

Table 5. The 3-Agent Broadcast Scenario

	OB-MAP	BaGa-S	BaGa-S-medoids
R_c	NO	NO	NO
R	-3.72 ± 7.88	-16.55 ± 10.74	-6.75 ± 9.12
C	0 ± 0	0 ± 0	0 ± 0
$T[\text{ms}]$	520.7	300.0	514.0

Figure 5. The 3-Agent Broadcast Scenario

5 Discussion

Online decentralized planning under uncertainty has been the subject of a number of prior studies. Roth *et al.* [14], for example, proposed Dec-COMM, an approach to online planning that provides guaranteed coordination at the cost of completely ignoring observations unless these have been shared with all the team-mates. In common with OB-MAP, agents keep track of the possible joint beliefs of the team and use a Q_{MDP} (or Q_{POMDP}) heuristic to find an action that maximizes the expected reward over all possible beliefs. In order to guarantee that all agents will agree on the same action, they do not take into account their local observation when propagating the possible beliefs. Agents decide to communicate their observations when including these would result in choosing a different action.

BaGa-S builds upon BaGa [5]. In the BaGa algorithm, each agent

keeps track of all possible types (local histories) of all agents and finds a policy mapping, for each agent type to actions. A policy represents an equilibrium; *i.e.* a situation where no agent can improve the expected value of the policy by unilaterally changing their local policy. Montemerlo proposed a number of heuristics for deciding whether or not to communicate, some of which take into account the cost of communication. They consider a model of communication where agents can decide to broadcast their own history to other agents, without requiring other agents to also synchronize their history. In order to estimate the value of communication, each agent prunes all the possible histories that are incompatible with its local histories and find a policy for that belief pool.

In MAOP-COMM [15], agents perform one-step planning in a decentralized manner and estimate the long-term value from a belief using a Q_{MDP} heuristic. In order to find a locally optimal equilibrium among agents' policies, they use alternative maximization, where each agent iteratively improves its policies assuming other agents' policies are fixed. Agents communicate when they receive a local observation that is inconsistent with their beliefs. While these algorithms represent improvements in the way observations are taken into account, the fact that the policy must represent an equilibrium between all histories, even those that are not compatible with an agent's local history, results in policies that are sometimes too conservative. Moreover, by taking into account only joint histories that are compatible with an agent's observations, we can decrease the number of candidate histories and obtain better clusters.

As mentioned by Wu *et al.* [15], BaGa is not able to deal with scenarios as large as those used to evaluate our approach. This is due to the fact that the clustering technique used in BaGa does not ensure that only a bounded number of beliefs is retained at each step. Moreover, the results reported by Wu *et al.* [15] shows that MAOP-COMM out-performs Dec-COMM in most situations. For these reasons we used MAOP-COMM to compare OB-MAP against; it represents the best available state-of-the-art algorithm.

Kaufman and Roberts [9] analyse the trade-off between using local information and guaranteed coordination in multi-agent planning. Their analysis, however, considered limited scenarios where the transition probabilities are uniform and therefore the value of observations is limited.

6 Conclusion

In this paper we have proposed OB-MAP, a novel algorithm for online planning in Dec-POMDPs. The algorithm is inspired by BaGa-S in terms of the use of local observations, but also enables value-aware communication between agents to maintain coordination in domains in which local observations are insufficient. This provides a balance between approaches that guarantee strict coordination but fail to take into account local information during planning, and approaches that use local information but fail to maintain acceptable levels of coordination in many scenarios. We propose a heuristic to decide when, given the cost of communication, agents should communicate in order to synchronize their histories and agree on a common joint belief. We evaluated our approach on a number of benchmark scenarios and we have shown that it performs significantly better than two algorithms that best represent the state-of-the-art.

Acknowledgments

This research has been sponsored by SELEX ES. We thank Feng Wu for providing the source code of the MAOP-COMM planner.

REFERENCES

- [1] C. Amato, 'Cooperative decision making.', in *Decision Making Under Uncertainty: Theory and Application*, ed., Mykel J. Kochenderfer, chapter 7, MIT Press, (2014).
- [2] C. Amato, J. S. Dibangoye, and S. Zilberstein, 'Incremental policy generation for finite-horizon Dec-POMDPs', in *Proceedings of the 19th International Conference on Automated Planning and Scheduling*, pp. 2–9, (2009).
- [3] C. Amato and S. Zilberstein, 'Achieving goals in decentralized POMDPs', in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pp. 593–600, (2009).
- [4] A. Chechotka and K. Sycara, 'Subjective approximate solutions for decentralized POMDPs', in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, p. 224, (2007).
- [5] R. Emery-Montemerlo, *Game-Theoretic Control for Robot Teams*, Ph.D. dissertation, Rutgers, The State University of New Jersey, 2005.
- [6] C. V. Goldman and S. Zilberstein, 'Decentralized control of cooperative systems: Categorization and complexity analysis', *Journal of Artificial Intelligence Research*, **22**, 143–174, (2004).
- [7] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, 'Dynamic programming for partially observable stochastic games', in *Proceedings of the 19th National Conference on Artificial Intelligence*, pp. 709–715, (2004).
- [8] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*, volume 344, John Wiley & Sons, 2009.
- [9] M. Kaufman and S. Roberts, 'Coordination vs. information in multi-agent decision processes', in *Proceedings of the 5th Workshop on Multi-agent Sequential Decision Making in Uncertain Domains*, pp. 1–6, (2010).
- [10] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella, 'Taming decentralized POMDPs: Towards efficient policy computation for multi-agent settings', in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pp. 705–711, (2003).
- [11] R. T. Ng and J. Han, 'CLARANS: A method for clustering objects for spatial data mining', *IEEE Transactions on Knowledge and Data Engineering*, **14**(5), 1003–1016, (2002).
- [12] F. A. Oliehoek, S. Whiteson, and M. T. J. Spaan, 'Lossless clustering of histories in decentralized POMDPs', in *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, pp. 577–584, (2009).
- [13] M. Roth, R. Simmons, and M. Veloso, 'Reasoning about joint beliefs for execution-time communication decisions', in *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 786–793, (2005).
- [14] M. Roth, R. Simmons, and M. Veloso, 'What to communicate? execution-time decision in multi-agent POMDPs', in *Distributed Autonomous Robotic Systems 7*, 177–186, Springer, (2006).
- [15] F. Wu, S. Zilberstein, and X. Chen, 'Online planning for multi-agent systems with bounded communication', *Artificial Intelligence*, **175**(2), 487–511, (2011).

A Framework for Actionable Clustering Using Constraint Programming

Thi-Bich-Hanh Dao¹ and Christel Vrain¹ and Khanh-Chuong Duong¹ and Ian Davidson²

Abstract. Consider if you wish to cluster your ego network in Facebook so as to find several useful groups each of which you can invite to a different dinner party. You may require that each cluster must contain equal number of males and females, that the width of a cluster in terms of age is at most 10 and that each person in a cluster should have at least r other people with the same hobby. These are examples of cardinality, geometric and density requirements/constraints respectfully that can make the clustering useful for a given purpose. However existing formulations of constrained clustering were not designed to handle these constraints since they typically deal with low-level, instance-level constraints. We formulate a constraint programming (CP) languages formulation of clustering with these cluster-level styles of constraints which we call *actionable clustering*. Experimental results show the potential uses of this work to make clustering more actionable. We also show that these constraints can be used to improve the accuracy of semi-supervised clustering.

1 Introduction

Most clustering is unsupervised with a recent movement to adding constraints, an area generally known as constrained clustering [2]. Previous work is most suitable for the *semi-supervised* setting where a few instances are labeled and the instance-level `must-link` and `cannot-link` constraints can be generated from them [2]. The data is then clustered under small numbers of these constraints with the number of clusters equaling the number of different labels. Performance is typically measured in terms of prediction: how well the clustering found matches the ground truth clustering induced by the labels. However, in many domains experts can provide complex constraints that are not generated from a ground truth, rather they capture what makes the clustering useful (or not useful) in the domain. To emphasize this focus we term this *actionable* clustering.

Consider our motivating example of clustering an ego network so that each cluster can be invited to a separate dinner party. A clustering algorithm may find a useful grouping that results in a successful party but is unlikely to unless we somehow encode what is required. Further uses can be modeled in the semi-supervised clustering setting if we know something about the underlying label set. Suppose we have labels that correspond to gender. If we know that in our data set there are twice as many males as females we can constrain the cluster sizes accordingly. Similarly if we know the males are typi-

cally closer in age than the females we can create the requirement to enforce that the diameter of one cluster is smaller than the other.

Existing instance level constraints cannot be used to specify this type of guidance. Consider the constraint that the number of males and females in each cluster should be approximately equal. Since instance level constraints are specified *a priori* to the algorithm beginning execution they typically cannot constrain any property that dynamically changes during the algorithm's execution such as the cluster composition. Constraining cluster level properties has probably not been well studied as it is challenging to do so in procedural languages and mathematical programming but can be elegantly performed in constraint programming (CP) languages which we use in this work.

The use of CP means that any clustering algorithm implementation will not scale to data sets larger than 10,000 points as these methods find the global optimum. Though this will prevent our work's usage on big data problems, recently work has shown [15, 14] many interesting real world problems are small data problems involving small but difficult to understand data sets. One characteristic of these problems is the need to find the very best clustering (the global optima) which effectively precludes scalability due to the intractability of the underlying optimization problem. Our work can be considered as allowing actionable clustering for these small data problems but our work can still be applied to thousands of data points as we have done in our experiments.

Our contributions are as follows:

- We look at new types of constraints beyond instance level constraints that must be calculated dynamically allowing for a new style of cluster level constraints.
- We introduce cardinality, density and geometric styles of cluster level constraints and show they can be easily specified in CP formulations.
- We experimentally explore the uses of these constraints to enforce guidance to ensure clusters are actionable by showing they can find alternative clusterings.
- We show experimentally that these constraints can help in the semi-supervised setting in a number of ways. They can be used in addition to instance level constraints or can be used in lieu of instance level constraints to improve clustering accuracy.

Our paper is laid out as follows. In the next section we outline the actionable clustering problem and then related work. Next follows the type of constraints we can build actionable clusters on including geometric, density and cardinality constraints. Such constraints are challenging to encode in procedural or mathematical programming formulations so instead we use a CP framework that allows to encode a variety of objective functions. The benefits of modeling both

¹ Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, F-45067, Orléans, France, email: thi-bich-hanh.dao@univ-orleans.fr, khanh-chuong.duong@univ-orleans.fr, christel.vrain@univ-orleans.fr

² Department of Computer Science, University of California, Davis, email: davidson@cs.ucdavis.edu

the constraints and objective functions in CP include ease of solving but also allow specifying a framework which others can build upon by adding in more objectives and constraints. Next we describe our experiments including a variety of data set types after which we conclude.

2 The Actionable Clustering Problem

Consider the classic clustering problem. We are given a data set \mathcal{X} where each instance $x \in \mathcal{X}$ is described by a vector of features f . Typical objective functions include minimizing the vector quantization error (k-means), minimizing graph cuts (spectral methods) if the data is represented as a graph and a similarity measure is used, and optimizing cluster properties such as minimizing the maximum cluster diameter. In our work we present the novel extension that each instance is further described by a set of properties from which the definitions of what is actionable/interesting is given. To separate our features and properties we use the notation: x_i^f and x_i^p to represent the features and properties of the i^{th} instance respectfully.

In our formulation the feature vectors \mathcal{X}^f are used to calculate the clustering objective function value and the constraints are enforced on the property vectors \mathcal{X}^p . However, there is nothing stopping the same attribute of an instance being in both vectors and used as both a feature and property.

Formally the actionable clustering problem is formulated as:

$$\begin{aligned} & \operatorname{argmin}_{\Pi} g(\mathcal{X}^f, \Pi) \quad (1) \\ \text{s.t. } & \phi_c(\mathcal{X}^p, \Pi), \phi_d(\mathcal{X}^p, \Pi), \phi_g(\mathcal{X}^p, \Pi) \\ & \text{where } \Pi \text{ is a partition/clustering,} \\ & g \text{ is an objective function,} \end{aligned}$$

$\phi_{c,d,g}$ models the cardinality, density and geometric constraints

The type of constraints we will explore in this paper can be divided into four categories: i) cardinality, ii) density, iii) geometric and iv) complex logical combination of these constraints which we now discuss in turn. They are not the only relevant ones but the most pragmatic in clustering: cardinality constraints are useful for categorical attributes, density constraints for relational information and geometric constraints for real value attributes. It is important to note that these constraints can be applied **simultaneously for multiple different properties** on multiple clusters.

i) *Cardinality* constraints place a requirement on a count of the elements in a cluster having a property. They may be as simple as each cluster should contain at least one female to more complex variations such as the number of males must be no greater than two times the number of females.

ii) *Density* constraints relate to a cardinality constraint in that it provides requirements on a count of a property except not for an entire cluster but rather a subset of instances in the cluster. For example, we may require each person have at least 10 people in his/her cluster sharing the same hobby.

iii) *Geometric* constraints place an upper or lower bound on some geometric property of a cluster or cluster combination. Examples include that the maximum diameter of a cluster with respect to the age property is 10 years. This would prevent clusters containing individuals with a wide range of ages.

iv) *Complex logic* constraints express logic combinations of constraints, which can be instance-level or cluster-level constraints. For instance, we may require that any cluster having more than 2 professors should have more than 10 PhD students.

Positive vs Negative Requirements. Up to this point we have discussed finding clusters which are actionable since they meet a particular set of requirements. However, it is also likely that a clustering is actionable because it *does not* contain a set of properties. This idea of using negative feedback was first explored in the alternative clustering literature [21, 3, 25, 19]. There the problem was given a good (according to the objective function value) clustering Π which is not actionable (perhaps because it is trivial or inappropriate) find an alternative clustering Π' such that Π' has a good objective function value but Π and Π' are different using some sort of measure such as the Rand index. However, that work is limited in that *how* Π' is different to Π is not controlled. Instead with actionable clustering if the existing clustering has undesirable properties such all females in one cluster, we can explicitly require that females be equally distributed (i.e. $\forall i, j \text{ CountFemale}(\pi_i) \approx \text{CountFemale}(\pi_j)$ where π_i is the i^{th} cluster).

3 Related Work

The style of constraints we explore in this paper to our knowledge has not been explored before. We briefly survey the related areas of: i) instance level constrained clustering, ii) relational clustering and iii) clustering using CP.

Instance level constrained clustering. As mentioned earlier most of this work is applicable to the semi-supervised setting where the `must-link` and `cannot-link` constraints must be given apriori. There have been a large variety of clustering algorithms to encode these constraints such as k-means, hierarchical, expectation maximization (EM) and spectral methods [2]. Our work differs since we explore more complex constraints beyond simple pairwise instance level constraints. Moreover, our framework finds a global optimum while classic clustering algorithms look for local optima.

Relational clustering. The area of relational clustering [20] aims to cluster data represented by both feature and simple pairwise relations such as `brother`, `son-of` and `married`. The technical challenge is to combine both feature and relational information in such a way that both can be used in the objective function of the clustering algorithm. Our work differs since we are looking at more complex guidance beyond relational information but also as we use extra information beyond features as constraints to be enforced not modeled as part of the objective function.

Clustering using CP. To our knowledge little work uses Constraint Programming for clustering. We proposed a CP model for distance-based clustering integrating several optimization criteria and different kinds of constraints [4, 6]. This work integrates only classic constraints and does not contain the complex constraints on properties we consider in this paper. A fundamentally different pattern mining approach based on CP [10, 17] has been generalized to k -pattern set mining [18] and applied to model conceptual clustering, where each cluster is characterized by a set of properties common to all the elements of the cluster.

4 A Quick Primer on CP

There are many benefits to formulate our work in a CP language. Not only does it allow elegant formulation as a constrained optimization problem but it guarantees to find a global optimum, a critical requirement for valued data that are small and collected only once (eg. fMRI data). Furthermore, relaxing the problem so it becomes just a satisfaction problem allows to explore all the possible solutions and even to determine if there exists a feasible one.

A *Constraint Satisfaction Problem* is modeled by (X, Dom, C) , where X is a set of variables, each variable $x \in X$ is associated with a domain $Dom(x)$ and C is a set of constraints, each constraint expresses a condition on a subset of X . A solution is a complete assignment of value $v \in Dom(x)$ to each variable $x \in X$ that satisfies all the constraints of C . When an objective function is added, the problem becomes a *Constraint Optimization Problem* and the aim is to find a solution that optimizes the objective. The originality of CP solvers is to alternate two steps: propagation allowing to filter the variable domains and branching. Different strategies can be used to create and to order branches at each branching point. They can be standard search strategies defined by CP solvers or can be specifically developed. Moreover, many kinds of constraints are available: elementary constraints expressing arithmetic or logic relations, or global constraints, as for instance the cardinality constraint, denoted by $\#$ and allowing to count a set of objects and to place a constraint on the obtained number. Although equivalent to conjunctions of elementary constraints, global constraints usually benefit from more efficient filtering algorithms. Reified constraints are available, which allow to link a boolean variable to the truth value of a constraint.

5 A CP Formulation of Actionable Clustering

We begin by discussing how to find globally optimal clusterings in CP and then move onto discuss how to encode the new types of constraints we introduce in this work. As is typical in the field, we show how to encode these constraints in a generic language-free manner. To aid in reproducibility of results all code will be made available in the CP solver library Gecode³.

5.1 Clustering in CP

For modeling clustering we rely on the CP clustering model proposed in our earlier work [6]. Clusters are identified by their index, varying from 1 to K (K denotes the number of clusters). Instances are also identified by their index, ranging from 1 to N . The assignment of instances to clusters is modeled by the integer variables G_i for $i \in [1, N]$, with $Dom(G_i) = [1, K]$ (the set of integers from 1 to K): $G_i = k$ means that the i^{th} instance is put into cluster number k . A complete assignment of the variables G_i therefore defines a clustering. However, several complete assignments can lead to the same composition of the clusters, where the only difference is the index of the clusters. In order to break this kind of symmetry, the CP constraint *precede* ($[G_1, \dots, G_n], [1, \dots, K]$) is used. This constraint enforces that the instance number 1 is in the cluster number 1, and an instance number i can be in a cluster k if the cluster $k - 1$ is not empty and contains an instance j with $j < i$.

The model in [6] integrates several optimization criteria: maximizing the minimal separation between clusters, minimizing the maximal diameter of the clusters, minimizing the within-cluster sum of dissimilarities, or minimizing the within-cluster sum of squares [5]. All these criteria are NP-hard: minimizing the diameter is polynomial for $K = 2$ but NP-hard for $K \geq 3$ [16], maximizing the separation is polynomial [11] but becomes NP-hard with user constraints [8], minimizing the sum of dissimilarities is NP-hard since the weighted max-cut problem, which is NP-complete [12], is a particular instance of this problem with $K = 2$ and the NP-hardness of minimizing the sum of squares is proven in [1]. These criteria can be used in our framework where the distances are computed from \mathcal{X}^f .

³ <http://www.gecode.org>

We extend this previous work to integrate constraints put on the property \mathcal{X}^p . From the semantic point of view these constraints can be divided into four categories: cardinality constraints, density constraints, geometric constraints and complex logic constraints.

5.2 Cardinality Constraints

Cardinality constraints allow to express requirements on the number of instances that satisfy some conditions in each cluster. The condition can be for instance being more than 20 years old and the cardinality constraint can state that each cluster must have more than 30 persons being more than 20 years old. The minimal capacity constraint (also called minimum significant constraint in [13]) is then a special case of a cardinality constraint.

Given a condition, the set C of the instances that satisfy it can be computed and the number of instances of C that are in a cluster k can be captured using the CP cardinality constraint and a variable Y_k :

$$\#\{i \in C \mid G_i = k\} = Y_k \quad (2)$$

The constraint $\sum_{k=1}^K Y_k = |C|$ enforces the link between the variables Y_k . Cardinality constraints are then expressed by arithmetic constraints on Y_k . Let us illustrate this by some examples.

- *Each cluster must have at least 50 females of age more than 20.* The set C is the set of instances i such that *female*(i) = *true* and *age*(i) > 20. For $k \in [1, K]$, the constraint of Equation (2) is put as well as the constraint $Y_k \geq 50$. The number of new introduced variables is K and the number of constraints is $2K + 1$.
- *In each cluster, the number of teachers must be no less than half the number of students.* Let C_t and C_s be the sets of instances that are teachers and students, respectively. For $k \in [1, K]$, constraints similar to Equation (2) are put with the variables T_k and S_k to capture the number of teachers or students in the cluster k . These variables are linked by the constraint $2T_k \geq S_k$. The number of new variables is $2K$ and the number of constraints is $3K + 1$.

5.3 Density Constraints

Density constraints provide bounds on the occurrence of some properties on a subset of instances in each cluster. For instance, each person being more than 20 years old should have in his/her cluster more than 5 persons sharing the same hobby. Density constraints allow a more general form than the basic ϵ -ball count constraint [8]. To express this constraint, for each instance $i \in [1, N]$ which is eligible (eg. more than 20 years old), the set of neighborhood instances $NI(i)$ (eg. persons having the same hobby) is determined. The number of instances of $NI(i)$ in the same cluster as i can be captured using the variable Z_i and:

$$\#\{j \in NI(i) \mid G_j = G_i\} = Z_i \quad (3)$$

Arithmetic conditions are then stated on Z_i to express density constraints. Let us take the following examples.

- *In the same cluster, each person should have at least 5 persons having the same hobby.* For each instance i , we compute the set $NI(i) = \{j \in [1, n] \mid \text{hobby}(i) = \text{hobby}(j)\}$. The fact that there must be at least 5 other persons of $NI(i)$ in the same cluster as i means the value of G_i must be taken at least 6 times by the elements in $NI(i)$. Therefore the constraint of Equation (3) is put as well as the constraint $Z_i \geq 6$.

- *Each person of age between 20 and 55 should have at least 10% persons with a difference in age less than 5 in the same cluster.* For each instance i such that $20 \leq \text{age}(i) \leq 55$, $NI(i)$ contains all the instances j such that $|\text{age}(i) - \text{age}(j)| \leq 5$. The constraint of Equation (3) is put as well as the constraint $Z_i \geq |NI(i)|/10$.

In these cases, the number of new introduced variables is N and the number of CP constraints is $2N$. Let us notice that the computation of the neighborhoods is done only once before putting CP constraints.

5.4 Geometric Constraints

Geometric constraints allow to set bounds on some geometric properties inside each cluster, or between the clusters.

- *Any two clusters must be separated by at least 10 on the age property.* Therefore any pair of instances i, j having $|\text{age}(i) - \text{age}(j)| < 10$ must be in the same cluster. For these pairs of instances the constraint $G_i = G_j$ is put.
- *Each cluster must have at most 40 difference on the age property.* That means any pair of instances i, j having $|\text{age}(i) - \text{age}(j)| > 40$ must be in different clusters. For them, $G_i \neq G_j$ is put.

In these cases, the number of CP constraints needed is at most quadratic compared to the number of instances. However, we have defined in our earlier work [6] the global constraints *split* and *diameter* that use a distance d . The constraint *split*(G, S, d) enforces that the minimal split (separation) between cluster is captured by a variable S , and *diameter*(G, D, d) enforces that the maximal diameter of the clusters is captured by a variable D . These global constraints are more efficient than expressing split or diameter constraints by must-link or cannot-link constraints. They can be used in our setting with d defined by $d(i, j) = |\text{age}(i) - \text{age}(j)|$, the first case is then expressed by two constraints *split*(G, S, d) and $S \geq 10$.

A geometric constraint can also place a bound on the sum of all the values on some properties inside each cluster, or a condition on the ranges of some properties of the clusters. For instance, age ranges of the clusters should or should not overlap, or the total sum of age in each cluster must not exceed some value.

- *The average age in each cluster must not exceed 50.* To express this constraint, for each instance i and each cluster k , we introduce a boolean variable $B_{ik} \in \{0, 1\}$ (0: *false* and 1: *true*). A reified constraint⁴ $B_{ik} \leftrightarrow (G_i = k)$ is put, ie. B_{ik} represents whether or not instance i is in the cluster k . For each $k \in [1, K]$, the sum of age S_k and the cardinality C_k are linked by:

$$\sum_{i \in [1, N]} \text{age}(i) B_{ik} = S_k \\ \#\{i \in [1, N] \mid G_i = k\} = C_k$$

The bound is therefore expressed by: $S_k \leq 50C_k$. In this case, $N \times K$ boolean variables (B_{ik}), K float point value variables (S_k) and K integer value variable (C_k) are introduced. This case is expressed by $3K$ CP constraints.

- *Constraints on the property ranges of the clusters.* We consider constraints that state conditions on the ranges of the clusters on a property p . To capture the range of a cluster, for each cluster k , we introduce the variables Min_k and Max_k that represent the

minimal and the maximal values on the property p of the elements in the cluster k . Let m_p be the maximal value of the property p for all the instances. The minimal and maximal values are linked by the following constraints:

$$Min_k = \min_{i \in [1, n]} (p(i) B_{ik} + m_p (1 - B_{ik})) \\ Max_k = \max_{i \in [1, n]} (p(i) B_{ik})$$

The constraint that the ranges on p of the clusters should not overlap can be expressed by putting, for any two clusters k, k' ,

$$(Min_k > Max_{k'}) \vee (Min_{k'} > Max_k)$$

A constraint stating that the range of a cluster k must be included in the range of another cluster k' can be expressed by:

$$Min_k \geq Min_{k'} \text{ and } Max_k \leq Max_{k'}$$

This requires $N \times K$ boolean variables and $2K$ float point value variables. The number of constraints is linear on K .

5.5 Complex Logic Constraints

Complex logic constraints can be used to enhance the expressivity power of formulating knowledge. This can be done in CP using reified and Boolean constraints as shown by the following examples.

- *Two instances 3, 9 are in the same cluster if the instances 11, 15 are in different clusters.* Two Boolean variables B_1, B_2 are introduced with the constraints: $B_1 \leftrightarrow (G_{11} \neq G_{15})$, $B_2 \leftrightarrow (G_3 = G_9)$ and $B_1 \leq B_2$.
- *Any cluster having more than 5 professors must have at least 10 PhD students.* For each $k \in [1, K]$, let P_k and S_k be the variables that capture the number of professors and students in the cluster k , using cardinality constraints such as in Equation (2). Two Boolean variables BP_k and BS_k are introduced and linked by $BP_k \leftrightarrow (P_k \geq 5)$, $BS_k \leftrightarrow (S_k \geq 10)$, and $BP_k \leq BS_k$.

6 Experiments

We begin by outlining the underlying questions our experiments attempt to address and then move onto the experimental methodology and results. It is important to realize that many clustering papers will have experiments showing how well their algorithm performed by performance on some objective function value. These types of experiments are not applicable in our setting since the CP framework finds the global optima. Similarly experiments to verify our method finds a clustering that satisfies the constraints are not required since once the requirements are formulated by constraints, the CP framework is guaranteed to converge if a feasible clustering exists. Our experiments attempt to address the following questions:

1. In the semi-supervised clustering setting, can the quality of the solution be improved with constraints beyond instance-level constraints?
2. Can our constraints be used to find actionable clusterings that are alternatives to an existing clustering?
3. How do our constraints increase the run-time of the underlying clustering algorithm?

For the first question, we show that a cardinality constraint added to instance-level constraints yields better solutions than just using

⁴ A reified constraint on a constraint c , stated by $B \leftrightarrow c$, links the truth value of a constraint c to a boolean variable B : B is 1 if the constraint c is satisfied, 0 if c cannot be satisfied, or $B \in \{0, 1\}$ if the satisfaction of c has not yet been determined.

instance-level constraints (see Table 2). This is so as we can effectively control the resultant cluster sizes to better match the label populations, something instance level constraints cannot readily do. We also show for the setting where different labels are not well separated that a geometric constraint can increase performance where as previously no instance level constraints have been able to. Finally, we also show that cardinality constraints enforcing cluster size by themselves can improve semi-supervised clustering accuracy (see Figures 2 and 3). For the second question we show on a visually understandable digit data set that our constraints are capable of finding quite different clusterings (Figures 4, 5 and 6) with even simple constraints. Importantly this explores the novel direction of *guided* alternative clustering. For the last question we show that examples of cardinality, diameter geometric and logical constraints do increase run times compared to using no constraints (Tables 4, 6 and 7). This increase is only fractional for most of cases but can be more important for instance with a density constraint.

The framework is implemented using the CP solver library Gecode 4.3.3. The objective function in the experiments is to minimize the maximal diameter of the clusters. As it was done in [4, 6], the instances are reordered by the Furthest Point First algorithm [16], so that instances that are far from the others have small index. Concerning the search strategy, at each branching point, the solver chooses an unassigned variable G_i with the smallest remaining domain. All values $k \in \text{Dom}(G_i)$ are examined and the distance between instance i and cluster k is computed: it is equal to the smallest distance $d(i, j)$ such that G_j is instantiated to k , or 0 if cluster k is empty. The value k of the closest cluster to i is chosen and two branches are created with $G_i = k$ and $G_i \neq k$. All experiments are performed on a 3.4GHz Intel Core i5 processor with 8Gb of RAM under Ubuntu 14.04. The relevant code to reproduce the results is available on www.cp4clustering.com.

6.1 Improving Semi-Supervised Clustering Results

In the semi-supervised learning setting typically labels on a subset of instances are used to generate instance-level constraints such as must-link or cannot-link constraints. Here we first explore if the labels can be better exploited by inferring more complex constraints on the clusters. We illustrate this point by considering diverse UCI datasets, which vary on the number of instances and on the number of classes and that will be used in various conditions. They are described in Table 1. In these experiments all the objects of the datasets are considered and the number K of clusters is set to the true number of classes for each dataset. Performance is typically measured in terms of prediction: how well the found clustering matches the ground truth clustering. To measure the accuracy of a clustering P compared to the ground truth clustering P^* , we use the Rand Index [23] which is defined by $RI = (a + b)/(a + b + c + d)$, where a and b are the numbers of pairs of instances for which P and P^* are in agreement (a , or b , is the numbers of pairs of instances that are in the same class, or respectively in different classes, in both P and P^*), c and d are the numbers of pairs of instances for which P and P^* disagree (same class in P but different classes in P^* and vice versa). This index varies from 0 to 1 and the better the partitions are in agreement, the closer RI to 1.

Dataset	# objects	# classes
Iris	150	3
Wine	178	3
Glass	214	7
Ionosphere	351	2
Breast cancer	569	2
Synthetic control	600	6
Vehicle	846	4
Yeast	1484	10
Multiple feature morphology	2000	10
Image segmentation	2100	7

Table 1. Properties of datasets

6.1.1 Adding A Single Cardinality Constraint to Instance Level Constraints

Instance-level constraints can decrease the quality of the found clustering compared to the ground truth clustering, as was reported in our earlier work [9]. We consider the datasets Iris, Wine, Breast Cancer and Ionosphere in order to match the experiments in [9]. All the attributes are considered as features (\mathcal{X}^f) in order to compute pairwise Euclidean distance and they are also considered as properties (\mathcal{X}^p). We generate a number of randomly created (from labels) instance-level constraints as is standard [2]: two instances are randomly taken, whether their labels are the same or not a must-link or a cannot-link constraint is stated, and this is repeated until required the number of instance-level constraints is reached. On those same taken instances we generate a minimum cardinality constraint for all clusters as follows: let the whole dataset be of N instances, the labeled sample be of e instances, and the smallest cluster size observed on the labeled sample be m , then for the whole dataset, the smallest cluster size is set to $0.9 \frac{m}{e} N$. This simulates a user guess at how big the smallest cluster should be based on the less frequent occurring label.

We compare two cases: first, clustering with a set of randomly generated instance-level constraints and second, clustering with the very same instance-level constraints and a minimum cardinality constraint as described above. We perform 1000 experiments and report the average Rand Index of all the experiments in Table 2. We can observe that for Iris, Wine and Breast Cancer, adding a cluster cardinality constraint helps to get better clusterings on average. For these datasets, we analyze the distribution over all experiments of the accuracy decrease or increase over not using any constraints. Figure 1 shows that adding instance-level constraints decreases the quality in a large number of cases which agrees with [9]. On the other hand, the improvement is more stable when using a minimum cardinality constraint along with instance-level constraints. This is most like because enforcing a cardinality constraint prevents skewed cluster sizes which can yield poor performance.

#	Iris		Wine		Breast Cancer	
0	0.8737	0.8737	0.6859	0.6859	0.5509	0.5509
10	0.8735	0.8743	0.6844	0.6899	0.5547	0.7110
20	0.8768	0.8782	0.6857	0.6959	0.5577	0.6757
30	0.8811	0.8825	0.6885	0.7031	0.5602	0.6419

Table 2. The average Rand Index vs number of instance-level constraints over 1000 runs for 2 cases: just instance-level constraints (left) and instance-level constraints with a minimal cluster cardinality constraint (right).

A Challenging Data Set. We observe from Table 3 what was earlier reported [9] for Ionosphere, instance level constraints do not improve results significantly. In that earlier paper of ours it was shown

that the geometry of the data is quite different to the labels which means that many labels of different types are inter-mixed close together, that is there is not naturally a cluster for each class. Not surprisingly, adding a minimum cardinality constraint does not help either, as the clusterings found for this dataset with instance level constraints have quite balanced cluster sizes. Adding a minimum cardinality constraint in this case does not change therefore the result. We therefore experiment with a third case, where we force separate clusters as follows. From the subset of the labeled instances, we compute the approximate minimal split S between clusters, and infer that the clusters should be separated by at least $\frac{2}{3}S$. This enforces a cluster separation that would normally not be found with instance level constraints. This results in Table 3, which shows that this constraint yields better clustering.

#	Ionosphere		
0	0.4988	0.4988	0.4988
10	0.4992	0.4992	0.5988
20	0.4997	0.4997	0.5584
30	0.5002	0.5002	0.5364

Table 3. The average Rand Index over 1000 runs for 3 cases: (1) instance-level constraints, (2) instance-level constraints with a minimal cluster cardinality constraint, (3) only geometric minimal split constraint.

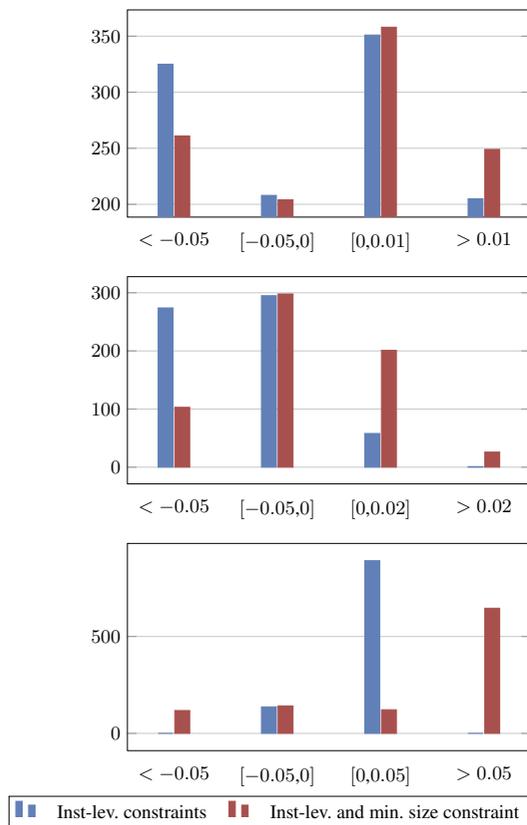


Figure 1. For the experiments reported in Table 2, the frequency distribution (y-axis) over the 1000 experiments by the amount of increase or decrease in accuracy over not using any constraints (x-axis). Ordering of figures are for Iris, Wine and Breast Cancer each one with 30 instance-level constraints. As we can see instance-level + cardinality constraints (red-bar) produces more increases and less decreases in accuracy.

6.1.2 Adding Multiple Cardinality Constraints

Here we consider the novel use of our work for semi-supervised clustering. Rather than using the labeled data to generate must-link and cannot-link constraints, we instead use the labels to provide upper and lower bound estimates on the cluster sizes. The datasets Glass, Breast cancer, Vehicle and Yeast, which have larger numbers of objects and of clusters are considered. We experiment four cases: (1) no constraint, (2) a constraint on minimal cluster size α is added, (3) a constraint on maximal cluster size β is added, and (4) a constraint on minimal cluster size α and a constraint on maximal cluster size β are added. The thresholds α and β are set depending on the known labels of the datasets: they are respectively 9 and 90 for Glass, 150 and 400 for Breast cancer, 150 and 250 for Vehicle and 10 and 500 for Yeast. The Rand Index is measured for each case and the results are presented in Figure 2. We can clearly observe that with a size constraint the Rand Index is improved compared to the unconstrained case. This behavior is the same for a small value of K (Breast cancer with $K = 2$) or with a large value of K (Yeast with $K = 10$). The improvement is very large for Breast cancer and convincingly shows that if the cluster sizes can be effectively estimated based on labels they can yield improved results over using no constraints.

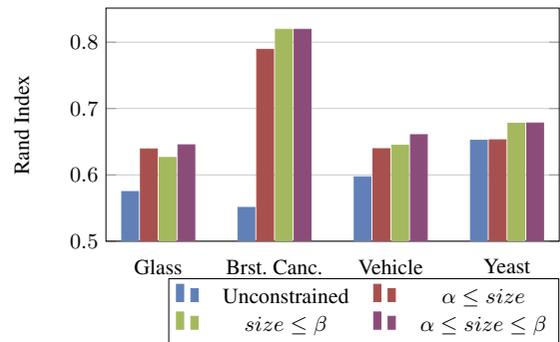


Figure 2. Rand Index in different cases: without constraint and with minimal and/or maximal size constraints

A related question is of course how do instance level and cardinality constraints interact. To explore this question we use a wide variety of data set properties. Figure 3 presents the obtained Rand Index for the datasets Breast cancer, Synthetic control, Multiple feature and Image segmentation with four other cases. In the first case, there is no user constraint. In the second case, 20 instance-level constraints are randomly generated and added. In the third case, the constraints requiring that the cluster size must be between α and β are added, where α and β are respectively 150 and 400 for Breast cancer, 50 and 150 for Synthetic control, 50 and 350 for Multiple feature and 200 and 400 for Image segmentation. In the fourth case both cluster size constraints and 20 instance-level constraints are added. For the second and the fourth cases, we make 100 runs and report the average Rand Index of all the runs. We can observe different behaviors with cluster size constraints here. While for Synthetic control, the constraints slightly decrease the Rand Index, for Multiple Feature, they bring slight improvement. The most significant improvement is observed for Image Segmentation. This dataset is composed by 2100 objects of 7 classes with 300 objects per class. In the unconstrained case, the clustering found has the Rand Index 0.226314 and is very unbalanced, with two clusters of 1 object and with a large cluster of 1972 objects. The situation is not improved with 20 random must-link or cannot-link constraints, the clusterings found are always un-

balanced. Adding cluster size constraints such that the clusters must have their size between 200 and 400, the clustering found has the Rand Index grow to nearly 0.80. On Breast cancer we can see that with only size constraints, the Rand Index is over 0.80, while the same constraint together with 20 instance-level constraints gives the Rand Index only about 0.66.

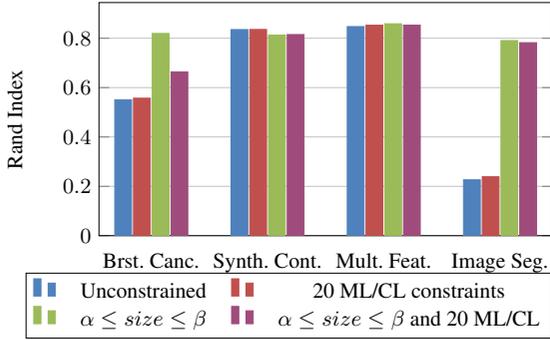


Figure 3. Rand Index in different cases: unconstrained, with instance-level constraints and/or minimal and maximal size constraints

6.2 Guided Alternative Clustering

In the area of alternative clustering one tries to find an equally good alternative to a given clustering [21, 3, 25, 19]. However, it is somewhat unrefined in that no guidance can be given to specify how the clustering is to differ from the given clustering. To address this issue we consider the UCI Pen Digit dataset where each instance corresponds to a single digit and has 16 attributes, which represent the 8 x, y positions of the pen as the digit is being written. All the 16 attributes are considered as features (\mathcal{X}^f) in order to compute pairwise Euclidean distances and are also considered as properties (\mathcal{X}^p). We use 1000 random instances of the dataset. We aim to find alternative ways that people write digits and we consider the simplest case where the number of clusters $k = 2$. This effectively forms a dichotomy of the two ways people in the data set write their digits.

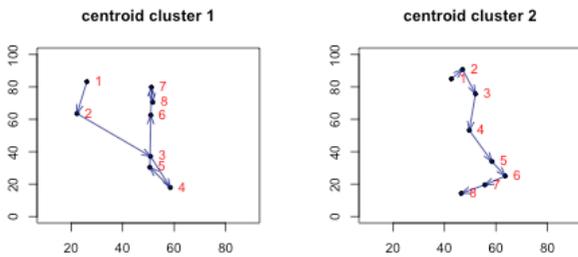


Figure 4. The centroids of the clustering found without any constraints. Time=1.16s, maximal cluster diameter=263.58. Arrows indicate pen movement.

For each cluster, the centroid is computed, which is considered as the representative of the cluster and can be easily visualized to show the underlying digit prototype the cluster represents. In the case of minimizing the maximal cluster diameter and without any constraints, the centroids of the found clusters, which represent two types of writing, are represented in Figure 4. The clustering found is the **global optimum** of the clustering algorithm objective function. However the centroids are not really meaningful unless of saying that in one way people write from up to down then up again, and in

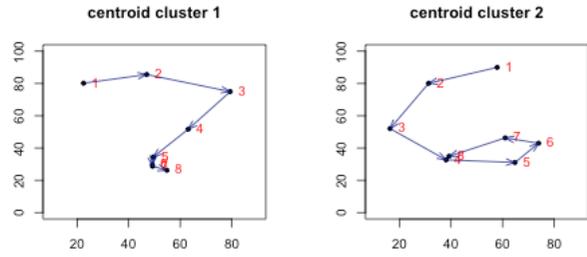


Figure 5. The centroids of the clustering found with a diameter constraint on the horizontal value of the third time step. Time=0.02s, maximal cluster diameter=291.50. Arrows indicate pen movement.

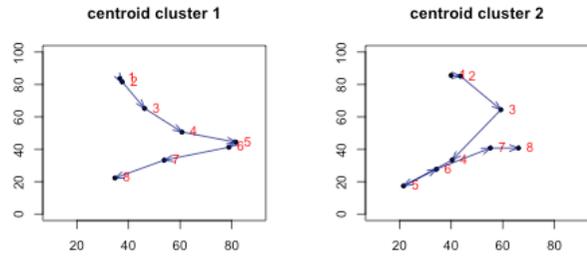


Figure 6. The centroids of the clustering found with a diameter constraint on the horizontal value of the fifth time step. Time=0.02s, maximal cluster diameter=269.68. Arrows indicate pen movement.

the other way only from up to down. Instead we wish to find an actionable alternative by adding a diameter constraint on the horizontal position of the pen at the 3rd time step. This effectively means that all digits in the same cluster must have a similar horizontal pen location at the third time step. We obtain a different clustering whose centroids are shown in Figure 5 but whose quality, in term of the objective function, is comparable to the first clustering found in Figure 4. These centroids have the 3rd positions respectively on the right and on the left. The centroids can give an interpretation such that in one way people write from left to right and in the other way like a spiral from right to left and in both ways from up to down. By adding a diameter constraint on the horizontal position of the pen at the 5th time step, we obtain another very different clustering whose centroids are shown in Figure 6. Again the centroids have the 5th position either on the right or on the left, and the quality of this clustering is comparable to the initial clustering in Figure 4. This is an example of guided alternative clustering which unlike ours and others earlier work [7, 3] did not find an arbitrary alternative clustering, rather we find one with specific properties. Adding constraints can deteriorate the quality in term of the objective function, however the flexibility of our framework allows to control the gap between the constrained case and the unconstrained case by means of constraints. For instance let the maximal diameter of the clusters in the unconstrained case be D_{opt} , one can require an actionable alternative clustering with both a diameter constraint on the horizontal position of the pen at the 3rd time step and another constraint stating that the maximal diameter of the clusters does not exceed $1.2D_{opt}$.

6.3 Computational Effect of Constraints

To address the computational effect of constraints, we report times taken by different cases in Subsection 6.1. The total runtimes (stating constraints and search) are presented in Table 4, where +1800 means the solver did not complete the search after 30 minutes. Cluster size constraints can give large variations in runtime. One explanation is

that the efficiency of a CP framework depends on the power of constraint propagation. For cardinality constraint filtering the domains of all variables to arc-consistency [22] is NP-hard. However, when setting an upper bound and a lower bound on the count variables, efficient filtering algorithms have been developed [24], which help pruning the search space.

Dataset	N	K	case	time (s)
Glass	214	7	unconstrained	0.02
			$9 \leq size \leq 90$	0.62
Breast cancer	569	2	unconstrained	0.30
			$150 \leq size \leq 400$	0.70
Vehicle	846	4	unconstrained	0.62
			$150 \leq size \leq 250$	5.57
Yeast	1484	10	unconstrained	3.15
			$10 \leq size \leq 500$	11.56
Synthetic Control	600	6	unconstrained	0.49
			$50 \leq size$	+1800
			$size \leq 150$	2.58
Multiple Feature	2000	10	unconstrained	10.30
			$50 \leq size$	76.80
			$size \leq 350$	+1800
Image Segmentation	2100	7	$50 \leq size \leq 350$	71.04
			unconstrained	3.29
			$200 \leq size$	86.85
			$size \leq 400$	+1800
			$200 \leq size \leq 400$	92.00

Table 4. Total runtime in seconds for (un)constrained cases with cardinality requirements.

For other kinds of constraints we consider the census dataset available at UCI⁵. This dataset has 48,842 instances, each one is described by 14 attributes with 6 continuous and 8 symbolic. We choose 5 continuous attributes (age, capital-gain, capital-loss, hours-per-week and fnlwtg) as features (\mathcal{X}^f) to compute distances on. All of the 14 attributes are used as properties. We generate 5 samples each one of 1000 instances and for each sample we conduct experiments with 5 use-cases described in Table 5. In each use-case, the number of clusters is set to 2 and to 3. Tables 6 and 7 give average runtime across five samples for the use cases with $K = 2$ and $K = 3$ respectively.

- | | |
|---|---|
| 1 | <i>No constraints.</i> |
| 2 | <i>Cardinality constraint.</i> A cardinality constraint is added, which requires that in each cluster, the ratio of $\#female_c/\#male_c$ for the cluster c is between a half and twice the ratio of females and males in the sample. |
| 3 | <i>Density constraint.</i> A constraint is added, stating each person of age between 20 and 50 must have at least 10% of people with the same work occupation in the same cluster. |
| 4 | <i>Diameter Geometric constraint.</i> A constraint is added, which states that the difference in age in each cluster must not exceed $2(\max(age) - \min(age))/3$. |
| 5 | <i>Complex logic constraint.</i> A constraint is added, which states that a cluster having more than 20 persons younger than 20 should have more than 30 persons older than 45, and each cluster has at least 100 persons. |

Table 5. The five use cases for testing the computation time effect of constraints.

Sample	UC1	UC2	UC3	UC4	UC5
1	0.42	0.41	2.51	0.42	0.49
2	0.32	0.69	1.71	0.32	0.35
3	0.44	0.54	2.94	0.41	0.34
4	0.44	0.31	0.71	0.32	0.34
5	0.36	0.48	2.16	0.32	0.32

Table 6. The runtime (seconds) for use cases (see Table 5) across five samples, for $K = 2$.

Sample	UC1	UC2	UC3	UC4	UC5	UC3+4
1	0.32	1.93	+1800	0.40	0.36	4.14
2	0.44	2.79	6.82	0.44	0.45	2.72
3	0.50	2.23	+1800	0.48	0.40	+1800
4	0.44	0.33	+1800	0.86	0.34	1.19
5	0.33	2.30	+1800	0.36	0.32	2.91

Table 7. The runtime (seconds) for use cases (see Table 5) across five samples, for $K = 3$. Note how using UC3 and UC4 together mitigates the increases of just using UC3.

We can see from Table 6 that the run-time to find the best solution in the unconstrained setting is under 0.5 second (use case 1) for all the 5 samples. Use cases 2, 4 and 5 take comparable run-time. Use case 3, which is expressed by a large number of CP cardinality constraints, is the most difficult among all the use cases. This trend is confirmed with $K = 3$ (Table 7), for the solver does not complete the search after the timeout of 30 min for 4 of the 5 samples.

One explanation for this variety of run times is that some constraints when added help the solver to prune the search tree at the top levels which has a large effect. Some other constraints, for instance, the cardinality constraint, however are useful in pruning the search tree only in more deeper levels towards the leaf nodes reducing down the benefits of pruning. On the other hand, constraints such as the diameter geometric constraint are useful in general. We have combined the diameter geometric constraint of use case 4 with the constraint of use case 3. The run-times of the combination reported in the last column of Table 7 have almost dropped for most of the samples.

7 Conclusion

Clustering is ubiquitously used in AI as it can add structure to collections of images, documents and even songs. A recent progression has been the addition of instance level constraints to clustering. These constraints though useful are severely limited in the information they can encode. In particular they cannot constrain any dynamic property of the clustering and hence cannot be used to enforce complex constraints such as those on cardinality (have equal number of males and females) or density (each person in the cluster should have other q persons with the same hobby). In this work we introduce three new styles of complex constraints, geometric, cardinality and density as well as logical combinations of them. We show how CP is a natural vehicle to encode such constraints and has the added benefit of finding the global optima. Although this requirement precludes scaling our work to huge data sets in many settings finding the global optima is desirable and our method works with thousands but not tens of thousands of instances. We showed that our new constraints can improve semi-supervised clustering accuracy when added to instance level constraints, find guided alternative clusterings and finally does not significantly increase run time except for density constraints.

⁵ <https://archive.ics.uci.edu/ml/datasets/Adult>

REFERENCES

- [1] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat, ‘NP-hardness of Euclidean Sum-of-squares Clustering’, *Machine Learning*, **75**(2), 245–248, (2009).
- [2] Sugato Basu, Ian Davidson, and Kiri Wagstaff, *Constrained clustering: Advances in algorithms, theory, and applications*, CRC Press, 2008.
- [3] Xuan Hong Dang and James Bailey, ‘A framework to uncover multiple alternative clusterings’, *Machine Learning*, **98**(1-2), 7–30, (2015).
- [4] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain, ‘A Declarative Framework for Constrained Clustering’, in *ECMLPKDD*, pp. 419–434, (2013).
- [5] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain, ‘Constrained minimum sum of squares clustering by constraint programming’, in *Principles and Practice of Constraint Programming, CP 2015, Proceedings*, pp. 557–573, (2015).
- [6] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain, ‘Constrained clustering by constraint programming’, *Artificial Intelligence*, (To appear).
- [7] Ian Davidson and Zijie Qi, ‘Finding alternative clusterings using constraints’, in *Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on*, pp. 773–778. IEEE, (2008).
- [8] Ian Davidson and S.S. Ravi, ‘The complexity of non-hierarchical clustering with instance and cluster level constraints’, *Data Min Knowl Disc*, **14**, 25–61, (2007).
- [9] Ian Davidson, Kiri L. Wagstaff, and Sugato Basu, ‘Measuring Constraint-Set Utility for Partitional Clustering Algorithms’, in *PKDD*, pp. 115–126, (2006).
- [10] Luc De Raedt, Tias Guns, and Siegfried Nijssen, ‘Constraint programming for itemset mining’, in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 204–212, (2008).
- [11] Michel Delattre and Pierre Hansen, ‘Bicriterion Cluster Analysis’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4), 277–291, (1980).
- [12] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, 1979.
- [13] Rong Ge, Martin Ester, Wen Jin, and Ian Davidson, ‘Constraint-driven clustering’, in *KDD*, pp. 320–329, (2007).
- [14] Sean Gilpin and Ian Davidson, ‘A flexible ilp formulation for hierarchical clustering’, *Artificial Intelligence*, (2015).
- [15] Sean Gilpin, Siegfried Nijssen, and Ian N Davidson, ‘Formalizing hierarchical clustering as integer linear programming.’, in *AAAI*, (2013).
- [16] T. Gonzalez, ‘Clustering to minimize the maximum intercluster distance’, *Theoretical Computer Science*, **38**, 293–306, (1985).
- [17] Tias Guns, Siegfried Nijssen, and Luc De Raedt, ‘Itemset mining: A constraint programming perspective’, *Artificial Intelligence*, **175**, 1951–1983, (2011).
- [18] Tias Guns, Siegfried Nijssen, and Luc De Raedt, ‘k-Pattern set mining under constraints’, *IEEE Transactions on Knowledge and Data Engineering*, **25**(2), 402–418, (2013).
- [19] Kleantjis-Nikolaos Kontonassios and Tijl De Bie, ‘Subjectively interesting alternative clusterings’, *Machine Learning*, **98**(1-2), 31–56, (2015).
- [20] Bo Long, Zhongfei Mark Zhang, and Philip S Yu, ‘A probabilistic framework for relational clustering’, in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 470–479. ACM, (2007).
- [21] Zijie Qi and Ian Davidson, ‘A principled and flexible framework for finding alternative clusterings’, in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 717–726. ACM, (2009).
- [22] Claude-Guy Quimper, Alejandro López-Ortiz, Peter van Beek, and Alexander Golynski, ‘Improved algorithms for the global cardinality constraint’, in *Principles and Practice of Constraint Programming - CP 2004, 10th International Conference, Proceedings*, pp. 542–556, (2004).
- [23] William M. Rand, ‘Objective Criteria for the Evaluation of Clustering Methods’, *Journal of the American Statistical Association*, **66**(336), 846–850, (1971).
- [24] Jean-Charles Régin, ‘Generalized arc consistency for global cardinality constraint’, in *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, August 4-8, 1996, Volume 1.*, pp. 209–215, (1996).
- [25] Duy Tin Truong and Roberto Battiti, ‘A flexible cluster-oriented alternative clustering algorithm for choosing from the pareto front of solutions’, *Machine Learning*, **98**(1-2), 57–91, (2015).

Repetitive Branch-and-Bound Using Constraint Programming for Constrained Minimum Sum-of-Squares Clustering

Tias Guns¹ and Thi-Bich-Hanh Dao² and Christel Vrain² and Khanh-Chuong Duong²

Abstract. Minimum sum-of-squares clustering (MSSC) is a widely studied task and numerous approximate as well as a number of exact algorithms have been developed for it. Recently the interest of integrating prior knowledge in data mining has been shown, and much attention has gone into incorporating user constraints into clustering algorithms in a generic way.

Exact methods for MSSC using integer linear programming or constraint programming have been shown to be able to incorporate a wide range of constraints. However, a better performing method for unconstrained exact clustering is the Repetitive Branch-and-Bound Algorithm (RBBA) algorithm. In this paper we show that both approaches can be combined. The key idea is to replace the internal branch-and-bound of RBBA by a constraint programming solver, and use it to compute tight lower and upper bounds. To achieve this, we integrate the computed bounds into the solver using a novel constraint. Our method combines the best of both worlds, and is generic as well as performing better than other exact constrained methods. Furthermore, we show that our method can be used for multi-objective MSSC clustering, including constrained multi-objective clustering.

1 INTRODUCTION

Cluster analysis or clustering is an important task in data mining, which has various applications in different domains such as biology, chemistry, medicine or business. Given a set of objects, cluster analysis aims at partitioning the objects into homogeneous subsets, called clusters. The homogeneity is usually formulated by an optimization criterion. One of the most used criterion is minimizing the Within-Cluster Sum of Squares (WCSS), which is defined by the sum of the squared Euclidean distances from each object to the centroid of the cluster to which it belongs. In order to make the clustering task more accurately fit the problem at hand, prior user knowledge has been integrated into the clustering process by means of user-defined constraints.

Minimum sum-of-squares clustering (MSSC) has been proven to be NP-Hard [1] and has been studied in numerous works. The well-known k-means algorithm as well as other dedicated heuristic algorithms find a local optimal for this criteria [21]. They have been also extended to integrate different user constraints but they can fail to

find a solution that satisfies all the constraints even when such a solution exists. On the other hand, general and declarative frameworks using generic optimization tools offer the flexibility of handling a wide variety of user constraints, and finding an exact solution of the problem whenever one exists. As a consequence, this precludes the use of these approaches on large datasets, but finding an exact solution may be of high importance on small but valuable datasets. Different frameworks have been proposed, based either on Integer Linear Programming with column generation [4] or on Constraint Programming [9].

On the other hand, Brusco [6] proposed a simple yet effective method for unconstrained MSSC: the Repetitive Branch-and-Bound Algorithm (RBBA). It computes increasingly tight bounds on the MSSC score by repetitively searching for the optimal solution, starting from a small subset of points up to the full set of all points. In this work we show how the idea of clustering with RBBA can be combined with the ideas of clustering with constraint programming [9].

Our contributions are as follows:

- We extend RBBA using Constraint Programming (CP) to support user-defined constraints. The key idea is to use CP in each branch-and-bound step and we show that this eases the modeling of a range of user constraints;
- The use of CP enables the computation of (constrained) lower bounds and upper bounds for the non-linear MSSC, and we develop a novel CP constraint that incorporates these bounds;
- We show that the resulting method is generic yet better performing than other exact constrained clustering methods.
- We experimentally illustrate the interest of our framework by its use in a multi-objective constrained clustering setting that minimizes WCSS and maximizes the split between clusters. To the best of our knowledge, this framework is the first one to support this bi-criterion clustering and different kinds of user-constraints.

Outline. Section 2 gives the preliminaries and Section 3 reviews related work. Section 4 presents RBBA and the extension we propose to integrate user constraints. Section 5 presents a framework using CP to achieve the extension of RBBA. Section 6 is devoted to the experiments and comparisons of our method with other existing approaches. Section 7 discusses perspectives and concludes.

2 PRELIMINARIES

Let us consider a dataset of N objects \mathcal{O} in an Euclidean space. Let d be the Euclidean distance ($d(o, o') = \|o - o'\|$). Minimum Sum-of-Squares Clustering (MSSC) aims at finding a partition Δ of the

¹ KU Leuven, Department of Computer Science, Celestijnenlaan 200A, Leuven, Belgium, email: tias.guns@cs.kuleuven.be

² Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, F-45067, Orléans, France, email: thi-bich-hanh.dao@univ-orleans.fr, khanh-chuong.duong@univ-orleans.fr, christel.vrain@univ-orleans.fr

objects into K clusters C_1, \dots, C_K such that: (1) $\forall k \in \{1, \dots, K\}$, $C_k \neq \emptyset$, (2) $\bigcup_k C_k = \mathcal{O}$, (3) $\forall k \neq k'$, $C_k \cap C_{k'} = \emptyset$ and (4) the Within-Cluster Sum of Squares (WCSS) is minimized. The WCSS criterion is defined by:

$$WCSS(\Delta) = \sum_{k \in \{1, \dots, K\}} \sum_{o \in C_k} d(o, m_k)^2 \quad (1)$$

where for each $k \in [1, K]$, m_k is the centroid (mean) of the cluster C_k . Equivalently [14, 16]:

$$WCSS(\Delta) = \sum_{k \in \{1, \dots, K\}} \frac{1}{2|C_k|} \sum_{o, o' \in C_k} d(o, o')^2 \quad (2)$$

There exists other optimization criteria, such as minimizing the Within-Cluster Sum of Dissimilarities criterion ($WCSD = \sum_{k \in \{1, \dots, K\}} \sum_{o, o' \in C_k} d(o, o')$), minimizing the maximal diameter D of the clusters ($maxDiam = \max_{k \in \{1, \dots, K\}} \max_{o, o' \in C_k} d(o, o')$) or maximizing the minimal split S between clusters ($minSplit = \min_{k, k' \in \{1, \dots, K\}, k \neq k'} \min_{o \in C_k, o' \in C_{k'}} d(o, o')$).

In applications, the user can have prior knowledge or requirements on the objects. For instance, the labels of a subset of objects can be known or an upper bound on the number of objects in each cluster can be required. Prior knowledge is integrated into the clustering process by user-defined constraints that have to be satisfied. User constraints can be *instance-level*, specifying requirements on pairs of objects, or *cluster-level*, giving requirements on the clusters. Instance-level constraints, introduced first in [25], are used most often. They are either must-link (ML) or cannot-link (CL) constraints on pairs of objects, which states that the objects must be or cannot be in the same cluster. Different kinds of cluster-level constraints also exist, the most popular ones being:

- A diameter constraint sets an upper bound γ on the cluster diameter: $\forall k \in \{1, \dots, K\}, \forall o, o' \in C_k, d(o, o') \leq \gamma$. This constraint can be expressed by cannot-link constraints: each pair of objects o, o' having $d(o, o') > \gamma$ must be in different clusters.
- A split constraint sets a lower bound δ on the separation between clusters: $\forall k \neq k' \in \{1, \dots, K\}, \forall o \in C_k, \forall o' \in C_{k'}, d(o, o') \geq \delta$. This constraint can be expressed by must-link constraints: each pair of objects o, o' having $d(o, o') < \delta$ must be in the same cluster.
- A density constraint requires that each object has in its neighborhood of radius ϵ at least m objects belonging to the same cluster as itself: $\forall k \in \{1, \dots, K\}, \forall o \in C_k, \exists o_1, \dots, o_m \in C_k \setminus \{o\}, d(o, o_i) \leq \epsilon$, or at least $m\%$ objects: $\forall k \in \{1, \dots, K\}, \forall o \in C_k, \frac{|\{o_i \in C_k | d(o, o_i) \leq \epsilon\}|}{|\{o_i \in \mathcal{O} | d(o, o_i) \leq \epsilon\}|} \geq \frac{m}{100}$.
- A minimal (maximal) capacity constraint requires each cluster to have at least (at most, resp.) a given α (β , resp.) number of objects: $\forall k \in \{1, \dots, K\}, |C_k| \geq \alpha$ (or $|C_k| \leq \beta$, resp.).

Constraint Programming (CP) is a constraint-based satisfaction and optimization framework. A constraint optimisation problem is expressed as a quadruple (V, D, C, f) where V is a set of *variables* and each variable $v \in V$ must take a value from its *domain* $D(v)$. The set C is a set of constraints over (a subset of) the variables V . The function f is an objective function defined over V and a solution that maximizes/minimizes f is an optimal solution.

Typical constraint solvers use depth-first branch-and-bound search. Each node in the search tree represents a partial solution consisting of a domain D' where $\forall v \in V : D'(v) \subseteq D(v)$. In

each node of the search tree, the constraint solver tries to *propagate* each constraint. Propagation is achieved when a constraint reduces the domains of its variables by removing those values that violate the constraint. For example, a constraint $X > 2$ can remove from $D(X) = \{1, 2, 3, 4, 5\}$ the values 1 and 2. Constraint solvers contain many different constraints, from logical to arithmetic and domain-specific constraints, such as for scheduling, each with its own propagation algorithm. If a propagator detects that the current partial solution cannot be extended to a full solution, namely when the domain of a variable becomes empty, the search backtracks. A solution is reached when the domain of each variable is reduced to a single value: $\forall v \in V : |D(v)| = 1$ and none of the constraints is violated. When a solution is reached, a new bound on the objective function is added stating that the next solution must score better than the currently best solution. Due to this branch-and-bound search, constraint solvers are exact: the search stops when it has proven that no better solution exists.

3 RELATED WORK

Constrained Minimum Sum-of-Squares Clustering has been studied in both heuristic and exact approaches. Among the heuristic approaches, even in the case without user constraints, the k-means algorithm as well as numerous other heuristic algorithms find a local optimal [21]. Considering must-link and cannot-link constraints, the k-means algorithm has been extended to COP-kmeans [26] or LCVQE [20]. However, when the number of constraints increases, such algorithms either fail to find a solution satisfying all the constraints even if one exists, or they find solutions that do not satisfy all the constraints.

Exact approaches for MSSC without user constraints use branch-and-bound search [18, 5, 6], dynamic programming [17, 23], Integer Linear Programming (ILP) and column generation [13, 3], a cutting plane algorithm [29] or a branch-and-cut semi-definite programming [2]. There exists few exact methods for MSSC that can handle user constraints [4, 9]. They are based on a generic optimization tool, so that different kinds of user constraints can be expressed. Extending [3], a framework based on ILP and column generation has been proposed in [4]. Using Constraint Programming (CP), a generic framework has been developed in [9], with a global constraint to compute and prune the search space for the WCSS criterion of MSSC.

Constrained clustering settings using an objective function different from WCSS have also been developed. A framework using ILP is proposed in [19]; it requires a set of clusters to be given in advance and considers different criteria to choose the best clustering from candidate clusters. A SAT based framework has been developed for constrained clustering for the diameter and the split criteria [11]. A well-performing CP based framework is developed in [7, 8] that includes diameter, split and sum of squared distances criteria, as well as user constraints.

Our work extends the Repetitive Branch-and-Bound Algorithm (RBBA) [6]. This algorithm finds a global optimal for MSSC without user constraints. We show that the methodology can be combined with a CP framework to obtain an efficient method that can easily incorporate user constraints.

4 EXTENDING RBBA TO USER-CONSTRAINTS

We first explain the bound used in RBBA and the standard RBBA algorithm. We then show the validity of the bounds under user con-

straints and how to extend the algorithm to support constraints in a generic way.

Let \mathcal{O} be a set of N points. Let Δ be a partition of \mathcal{O} into at most K clusters. For any subset S of \mathcal{O} , let Δ_S denote the projection of Δ onto the objects in S and $WCSS(\Delta_S)$ the WCSS value of Δ_S . Let $WCSS^*(S) = \min_{\Delta}(WCSS(\Delta_S))$. Let us note that in Δ_S some clusters of Δ may become empty.

4.1 Lower Bound Inequalities Without User-Constraints

The bounds used in RBBA rely on the following result [18]. Let S be a subset of \mathcal{O} , and let S_1 and S_2 be such that $S = S_1 \cup S_2$ and $S_1 \cap S_2 = \emptyset$ (non-overlapping). We have:

$$WCSS(\Delta_S) \geq WCSS(\Delta_{S_1}) + WCSS(\Delta_{S_2}) \quad (3)$$

Since $WCSS^*(S_2) = \min_{\Delta}(WCSS(\Delta_{S_2}))$, so $WCSS^*(S_2)$ is the smallest WCSS value for all partitions of S_2 into at most K clusters. Hence we have:

$$WCSS(\Delta_{S_2}) \geq WCSS^*(S_2) \quad (4)$$

and hence [6]:

$$WCSS(\Delta_S) \geq WCSS(\Delta_{S_1}) + WCSS^*(S_2) \quad (5)$$

Eq. (5) can be used during the search for an optimal partition of S as follows. Let us suppose that we have previously built a partition of S , thus giving an upper bound for $WCSS^*(S)$, that we have currently built a partial solution Δ_{S_1} and that we know an optimal solution of $WCSS^*(S_2)$. If $WCSS(\Delta_{S_1}) + WCSS^*(S_2)$ is greater than the actual upper bound, then the partial solution Δ_{S_1} can never lead to a better solution than the current upper bound.

4.2 Repetitive Branch-and-Bound Algorithm

The Repetitive Branch-and-Bound Algorithm (RBBA) [6] is presented in Algorithm 1.

Algorithm 1: RBBA input: objects \mathcal{O} , number clusters K

```

1 OrderPoints( $\mathcal{O}$ )
2  $\mathcal{O}_K \leftarrow \{o_{N-K+1}, \dots, o_N\}$ 
3  $\Delta_K^* \leftarrow \text{Init}(\mathcal{O}_K)$ 
4  $W_n \leftarrow 0, \forall n \in \{1, \dots, K\}$ 
5 for  $n = K + 1$  to  $N$  do
6    $\mathcal{O}_n \leftarrow \mathcal{O}_{n-1} \cup \{o_{N-n+1}\}$ 
7    $\Delta_n \leftarrow \text{Greedy\_Extension}(\mathcal{O}_n, \Delta_{n-1}^*)$ 
8    $U_n \leftarrow WCSS(\Delta_n)$ 
9    $\Delta_n^* \leftarrow \text{BaB\_Search}(\mathcal{O}_n, U_n, W)$ 
10   $W_n \leftarrow WCSS(\Delta_n^*)$ 

```

Points in \mathcal{O} are first ordered following an heuristic by $\text{OrderPoints}(\mathcal{O})$. Different heuristics can be used for ordering points, they will be presented in Subsection 4.5. We assume that according to the ordering, points are named by their index $i \in [1, N]$. \mathcal{O}_n is composed of the *last* n points according to this order.

In this algorithm, Δ_n indicates any partition of \mathcal{O}_n into at most K clusters and Δ_n^* denotes the optimal partition of \mathcal{O}_n into at most K clusters. This algorithm starts with the set \mathcal{O}_K of the last K points and $\text{Init}(\mathcal{O}_K)$ creates Δ_K^* by putting each point alone in a cluster.

The optimal value $WCSS(\Delta_n^*)$ is stored in W_n for each n , and the first K values W_1, \dots, W_K are 0 (each point in its own cluster).

The algorithm next iterates by adding to the set \mathcal{O}_n one point each time, from the point $N - K$ down to the first point; \mathcal{O}_n represents this set of last n points o_{N-n+1}, \dots, o_N ; $\text{Greedy_Extension}(\mathcal{O}_n, \Delta_{n-1}^*)$ greedily finds a partition Δ_n for \mathcal{O}_n , by adding the new point to the previous best partition Δ_{n-1}^* so that the value WCSS is minimally increased. The value $WCSS(\Delta_n)$ constitutes an upper bound U_n for $WCSS(\Delta_n^*)$. $\text{BaB_Search}(\mathcal{O}_n, U_n, W)$ is a branch-and-bound algorithm which searches for a global optimal partition Δ_n^* on the set of points \mathcal{O}_n , using U_n as an upper bound and exploiting Eq. (5) with the W_i values ($i < n$) as lower bounds. Let $o_m = o_{N-n+1}$ be the new point added at this step. The branch-and-bound search considers the points in \mathcal{O}_n in the order o_m, o_{m+1}, \dots, o_N and tries to assign them to clusters.

Let us consider an arbitrary step when a point number p ($m \leq p < N$) is assigned to a cluster. Let S_1 be the set of points $\{o_m, \dots, o_p\}$ and S_2 be $\{o_{p+1}, \dots, o_N\}$. All the points in S_1 have already been assigned and hence $WCSS(\Delta_{S_1})$ is known. All the points in S_2 are currently unassigned, however, $WCSS^*(S_2)$ has been computed in a previous step of RBBA and stored in $W_{|S_2|}$; U_n is the current upper bound. Eq. (5) is used and if $WCSS(\Delta_{S_1}) + WCSS^*(S_2) \geq U_n$, we cannot extend Δ_{S_1} to a solution having WCSS better than U_n . Therefore BaB_Search will not continue to extend Δ_{S_1} and the branch is pruned. When $p = N$, the partition Δ is complete and U_n is set to $WCSS(\Delta)$. When the entire search space is explored, the last complete partition found is the optimal solution.

This algorithm takes advantage of the optimal solutions previously computed to provide lower bounds in the branch-and-bound search. Also important are the upper bounds found by the greedy extension, they are often tight (meaning that the greedy extension is the optimal partitioning). Because of these tight bounds, even though the algorithm runs the branch-and-bound search N times, it is nevertheless one of the best exact algorithms for minimum sum-of-squares clustering. A similar search method was proposed for valued (soft) CSPs with an additive objective function, called Russian Doll Search [24].

4.3 Lower Bound Inequalities With User-Constraints

We now study the conditions under which Eq. (5) is still valid in the presence of a set of user constraints \mathcal{C} on \mathcal{O} . Given a set of points $S \subseteq \mathcal{O}$ and a set of constraints \mathcal{C} on S , $\mathcal{S}(S, \mathcal{C})$ denotes the set of all partitions Δ_S of S satisfying \mathcal{C} . We denote by $WCSS^*(S, \mathcal{C})$ the optimal WCSS of $S \subseteq \mathcal{O}$ under constraint set \mathcal{C} , that is, $WCSS^*(S, \mathcal{C}) = \min(\{WCSS(\Delta_S) | \Delta_S \in \mathcal{S}(S, \mathcal{C})\})$. We denote by $WCSS(\Delta_S, \mathcal{C})$ the WCSS value of a partition Δ_S under the condition that it satisfies the constraint set \mathcal{C} .

One can see from this that Eq. (4) still holds when considering a set of constraints \mathcal{C} : $WCSS(\Delta_S, \mathcal{C}) \geq WCSS^*(S, \mathcal{C})$. Indeed, any $\Delta_S \in \mathcal{S}(S, \mathcal{C})$ will have a score equal or worse than the optimal one satisfying \mathcal{C} .

The main question is then under what conditions Eq. (3), and hence (5), holds in the presence of constraints. Eq. (3) is always true, but the difficulty is that when considering a projection Δ_{S_i} of Δ_S with $S_i \subset S$, some constraints may become ill-defined or even be violated for Δ_{S_i} , even if they are satisfied by Δ_S . For instance, let us consider 5 points $\{a, b, c, d, e\}$, two cannot-link constraints $CL(a, b)$ and $CL(b, c)$ and a minimal size constraint specifying that each class must have at least 2 points. Let $\Delta_S = \{\{a, c\}, \{b, d, e\}\}$, $S_1 = \{a, b\}$ and $S_2 = \{c, d, e\}$. Then $\Delta_{S_1} = \{\{a\}, \{b\}\}$ and

$\Delta_{S_2} = \{\{c\}, \{d, e\}\}$. The constraint $CL(a, b)$ is satisfied on S_1 whereas $CL(b, c)$ is undefined on both S_1 and on S_2 . Moreover the minimal size constraint is satisfied on Δ_S but it is no longer satisfied on S_1 , nor on S_2 . The question is hence, given a set of constraints \mathcal{C} on S which Δ_S satisfies, what set of constraints \mathcal{C}_{S_i} can be put on S_1 and S_2 such that Eq. (5) is still valid?

In general, given a set of \mathcal{C} of constraints put on objects of S , we can restrict the set \mathcal{C}_{S_i} with $S_i \subseteq S$ to those constraints for which all objects in the constraint are in the set S_i . For example, one can add to S_i all instance-level constraints whose two objects are both in S_i . In the previous example, $CL(a, b)$ can be considered on S_1 whereas $CL(b, c)$ cannot. If a partition Δ_S satisfies a set of constraints \mathcal{C} , then its projection onto S_i (Δ_{S_i}) will satisfy the subset of constraints \mathcal{C}_{S_i} . Therefore

$$WCSS(\Delta, \mathcal{C}) \geq WCSS(\Delta_{S_1}, \mathcal{C}_{S_1}) + WCSS^*(S_2, \mathcal{C}_{S_2}) \quad (6)$$

Many *cluster-level* constraints involve all variables and hence with this approach cannot be considered until the very end. However, for two constraint sets \mathcal{C}_1 and \mathcal{C}_2 such that $\mathcal{C}_1 \subseteq \mathcal{C}_2$, then $\mathcal{S}(\mathcal{C}_2) \subseteq \mathcal{S}(\mathcal{C}_1)$ and therefore $WCSS^*(S, \mathcal{C}_1) \leq WCSS^*(S, \mathcal{C}_2)$. Hence, including more constraints can lead to tighter lower bounds.

In order to incorporate some cluster-level constraints, we distinguish those that are anti-monotonic from those that are not. A constraint c is said to be anti-monotonic if when satisfied by a partition Δ_S , it is satisfied by all the projections Δ_{S_i} , with $S_i \subseteq S$. In other words, let v_c be the function that tests whether c is satisfied on a partition. Then an anti-monotonic constraint satisfies the following property: if Δ is a partition on S and $S_i \subseteq S$ then $v_c(\Delta_{S_i}) \geq v_c(\Delta)$. As an example, a maximal size constraint is anti-monotonic whereas a minimal size constraint is not.

Let \mathcal{C}_a be the anti-monotonic constraints in \mathcal{C} . Then, since Δ_{S_2} satisfies the constraints on \mathcal{C}_{S_2} and the anti-monotonic constraints of \mathcal{C} , and similarly for S_1 , we have:

$$WCSS(\Delta, \mathcal{C}) \geq WCSS(\Delta_{S_1}) + WCSS(\Delta_{S_2}) \quad (7)$$

$$\geq WCSS(\Delta_{S_1}, \mathcal{C}_{S_1} \cup \mathcal{C}_a) + WCSS^*(S_2, \mathcal{C}_{S_2} \cup \mathcal{C}_a) \quad (8)$$

A constraint solver can additionally reason over *partial* solutions, namely over the domain of a set of variables. A constraint solver is guaranteed not to reject a partial solution that can be extended to a full solution, while it can reject partial solutions that provably can not satisfy a constraint (such as an anti-monotonic constraint and more). This will ease searching for a partial solution Δ_{S_1} in branch-and-bound search, without needing to identify $\mathcal{C}_{S_1} \cup \mathcal{C}_a$ each time S_1 changes.

4.4 RBBA with User Constraints

Let \mathcal{C} be the set of all constraints on \mathcal{O} . We assume that the set \mathcal{C} is satisfiable on \mathcal{O} , ie. there exists a partition Δ of \mathcal{O} that satisfies \mathcal{C} . The extension of RBBA to incorporate user constraints is presented in Algorithm 2.

After ordering points, Algorithm 2 constructs an initial partition Δ_K of *at most* K clusters taking constraints $\mathcal{C}_K = \mathcal{C}_{\mathcal{O}_K}$ into account. It does so by putting each point that can be in its own cluster in a separate cluster (if there is a must-link, the two points must be put in the same cluster). Among all such partitions, the one with smallest $WCSS(\Delta_K)$ is chosen. Since \mathcal{C} is satisfiable on \mathcal{O} , the partition Δ_K^* must exist.

At each step n , for the set \mathcal{O}_n of the last n points, Algorithm 2 searches in the solution space $\mathcal{S}(\mathcal{O}_n, \mathcal{C}_n)$. There are different options for the constraint set \mathcal{C}_n . As discussed in the previous section, \mathcal{C}_n can be $\mathcal{C}_{\mathcal{O}_n}$ or $\mathcal{C}_{\mathcal{O}_n} \cup \mathcal{C}_a$. We note that the more constraints that are considered at one step, the tighter the lower bound for the next step would be. At the last step, when $\mathcal{O}_N = \mathcal{O}$, the full set of user constraints \mathcal{C} , anti-monotonic or not, will be considered.

Feasible_Extension tries to extend the best partition of the previous step Δ_{n-1}^* to a partition Δ_n of \mathcal{O}_n that satisfies \mathcal{C}_n . If such an extension Δ_n exists, then $WCSS(\Delta_n)$ is an upper bound for $WCSS(\Delta_n^*)$. Otherwise, the upper bound is set to ∞ . *Constrained_BaB*($\mathcal{O}_n, \mathcal{C}_n, U_n, W$) performs a branch-and-bound search to find an optimal partition among all the partitions that satisfy the set of constraints \mathcal{C}_n . It uses U_n as the initial upper bound and W for the lower bounds, in the same way as *BAB_Search* in Algorithm 1.

Algorithm 2: Extended RBBA

input: objects \mathcal{O} , *number clusters* K , *constraint set* \mathcal{C}

```

1 OrderPoints( $\mathcal{O}$ )
2  $\mathcal{O}_K \leftarrow \{o_{N-K+1}, \dots, o_N\}$ 
3  $\Delta_K^* \leftarrow \text{Init}(\mathcal{O}_K, \mathcal{C}_K)$ 
4  $W_K \leftarrow WCSS(\Delta_K^*)$ 
5 for  $n = K + 1$  to  $N$  do
6    $\mathcal{O}_n \leftarrow \mathcal{O}_{n-1} \cup \{o_{N-n+1}\}$ 
7    $\Delta_n \leftarrow \text{Feasible.Extension}(\mathcal{O}_n, \mathcal{C}_n, \Delta_{n-1}^*)$ 
8   if  $\Delta_n$  exists then
9      $U_n \leftarrow WCSS(\Delta_n)$ 
10  else
11     $U_n \leftarrow \infty$ 
12   $\Delta_n^* \leftarrow \text{Constrained.BaB}(\mathcal{O}_n, \mathcal{C}_n, U_n, W)$ 
13   $W_n \leftarrow WCSS(\Delta_n^*)$ 

```

4.5 Ordering of Points

Algorithms 1 and 2 start by ordering points and they do branch-and-bound for an increasing set of points following this order. Different orders can be used. In RBBA [6], the nearest-neighbor separation heuristic is used: at each step of the ordering, the two points that have the smallest distance among all pairs of points are withdrawn from the set of points and are placed at opposite ends in the ordering. This heuristic is aimed at putting *easy-to-cluster* points near the end of the RBBA process, to avoid introducing disruptive points near the end of the process and hence having to do much search there.

The ordering that we will use is based on the furthest-point-first (FPF) algorithm [15]. This algorithm starts by choosing the furthest point from all points and stores it as the first point in the ordering. It then assigns this point as the *head* of all other points. At each iteration, the point i that is the furthest to its head is marked as the next point in the order, and all the unmarked points that are closer to i than to their head change their head to i . This ordering tends to put points that are far from each other early in the ordering, also aiming to consider *disruptive* points earlier in the process.

5 A FRAMEWORK USING CONSTRAINT PROGRAMMING

We present a framework to achieve Algorithm 2. In this framework, CP is used both to do complete branch-and-bound search for each

clustering step (*Constrained_Bab*) and to construct a feasible clustering if one exists (*Feasible_Extension*). We also present improvements for enhancing the computation of lower and upper bounds.

5.1 A Basic CP Model for Constrained BaB

Constrained_BaB($\mathcal{O}_n, \mathcal{C}_n, U_n, W$) in Algorithm 2 aims at finding a clustering Δ_n^* on \mathcal{O}_n that satisfies \mathcal{C}_n and that minimizes the sum-of-squares WCSS.

The CP model for this task is inspired by the model for constrained clustering in [8], the main difference being the objective. In order to define the assignment of points to clusters, integer value variables G_1, \dots, G_n with $Dom(G_i) = \{1, \dots, K\}$ are introduced. $G_i = k$ means that point i is assigned to the cluster number k . This formulation ensures that a point can never belong to two clusters. A complete assignment of the variables G_i therefore defines a partitioning. However, different assignment can represent the same partitioning but with a permutation on the cluster indices used. In order to break this kind of symmetry and to enforce that each partition corresponds to one complete assignment, the CP constraint *precede*($G, [1, \dots, K]$) is used [8]. This constraint enforces that point number 1 is in cluster number 1, and point number i can only have cluster number k if there is a point $j < i$ with the same cluster number, or if $k - 1$ is the highest used cluster number so far. For the objective, we introduce a floating point variable V to represent the sum-of-squares of the clustering defined by the variables G . The domain of V is initially $[0, U_n)$. The bounds of V are updated by a novel global constraint $V = \text{sumSquares}(G, d, W)$, where d is the (precomputed) distance between each pair of points, and W contains the previous WCSS* values (as per Algorithm 2).

Additional constraints can be expressed over the G variables, including the user constraints defined in Section 2. Instance-level constraints are expressed by $G_i = G_j$ for a must-link constraint and $G_i \neq G_j$ for a cannot-link constraint on i, j . A maximal cluster size constraint, following its formal definition, is expressed by K CP cardinality constraints: $\#\{i \in [1, N] \mid G_i = k\} \leq \beta$ for each $k \in [1, K]$. Each of these constraints enforces that the number of variables G_i that are assigned to k must not exceed β . Other constraints can be modelled following their formal definition as well, see [8] for more examples.

According to the principle of RBBA, the variable order used during search instantiates (branches over) the variables G_1, \dots, G_n in increasing order of their index.

5.2 A Novel Sum-of-Squares Constraint

The filtering algorithm for constraint $V = \text{sumSquares}(G, d, W)$ is detailed in Algorithm 3. Because of the variable order, at any time the propagator is called, there is an index p ($1 \leq p < n$) such that G_1, \dots, G_p are instantiated and G_{p+1}, \dots, G_n are not.

Algorithm 3 enforces bound consistency for V by first computing a lower bound for V . The values $\text{sum}[k]$ and $\text{size}[k]$ represent respectively the sum of squared distances between any two points in the cluster k and the number of points in that cluster. The value V_1 represents the sum of squares of the partial clustering formed by the first p assigned points, using Equation (2). Since W_{n-p} represents the minimal WCSS value for the last $n - p$ points (the unassigned points G_{p+1}, \dots, G_n), according to Equation (6) and (8), $V_1 + W_{n-p}$ is a lower bound for V (line 15). Since $V.lb \leq V < V.ub$, a failure will occur if $V_1 + W_{n-p} \geq V.ub$ (line 12) leading the search to backtrack. Otherwise the lower bound $V.lb$ is revised.

Algorithm 3 exploits also W to do a look ahead to filter the domain of G_{p+1} . Each value $s[k]$ represents the contribution of point $p + 1$ in case it is assigned to cluster k . For each $k \in Dom(G_{p+1})$, that is, all clusters k not forbidden for this point because of another constraint, if point $p + 1$ is assigned to the cluster k , V'_1 is the revised value of V_1 . So V'_1 represents the sum of squares of the partial clustering formed by the first $p + 1$ points. Since W_{n-p-1} represents the minimal WCSS value for the last $n - p - 1$ points, according to Equation (6), if $V'_1 + W_{n-p-1} \geq V.ub$ then a failure would occur. This means point $p + 1$ cannot be assigned to cluster k . The value k is then removed from $Dom(G_{p+1})$.

Algorithm 3: Filtering of: “ $V = \text{sumSquares}(G, d, W)$ ”

```

input:  $V, G, d, W$  with  $G_1, \dots, G_p$  assigned,  $G_{p+1}$  unassigned
// computation of lower bound for  $V$ 
1 for  $k = 1$  to  $K$  do
2    $\text{sum}[k] \leftarrow 0$ ;  $\text{size}[k] \leftarrow 0$ ;  $s[k] \leftarrow 0$ 
3 for  $i = 1$  to  $p$  do
4    $k \leftarrow G_i.\text{val}()$ 
5    $\text{size}[k] \leftarrow \text{size}[k] + 1$ 
6   for  $j = i + 1$  to  $p$  do
7     if  $G_j.\text{val}() == k$  then
8        $\text{sum}[k] \leftarrow \text{sum}[k] + d(i, j)^2$ 
9  $V_1 \leftarrow 0$ 
10 for  $k = 1$  to  $K$  do
11    $V_1 \leftarrow V_1 + \text{sum}[k]/\text{size}[k]$ 
12 if  $V_1 + W_{n-p} \geq V.ub$  then
13   return Failure
14 else
15    $V.lb \leftarrow \max(V.lb, V_1 + W_{n-p})$ 
// look ahead to filter  $Dom(G_{p+1})$ 
16 for  $i = 1$  to  $p$  do
17    $s[G_i.\text{val}()] \leftarrow s[G_i.\text{val}()] + d(i, p + 1)^2$ 
18 foreach  $k$  in  $Dom(G_{p+1})$  do
19    $V'_1 \leftarrow V_1 - \text{sum}[k]/\text{size}[k] + (\text{sum}[k] + s[k]) / (\text{size}[k] + 1)$ 
20   if  $V'_1 + W_{n-p-1} \geq V.ub$  then
21     remove  $k$  from  $Dom(G_{p+1})$ 

```

The complexity of this algorithm is $O(p^2)$, due to the computation of sum and size . It can be reduced to $O(p)$ when the arrays sum and size are stored and computed incrementally over different propagation runs.

5.3 Other Improvements

5.3.1 Must-link Constraints

Must-link constraints agglomerate related points to the same cluster. Therefore to make better use of this kind of constraint, first of all the transitive closure of all the must-link constraints is computed. This defines a set of super-points or ML-blocks [10]. Instead of clustering the set of initial points, we search for a clustering on the set of ML-blocks. Given a set of N initial points, assume that there are M ML-blocks to be considered ($M \leq N$). The distance between two ML-blocks b_i, b_j is defined as $d(b_i, b_j) = \sqrt{\sum_{o \in b_i, o' \in b_j} d(o, o')^2}$. Each block b_i has also its weight $w(i) = \sum_{o, o' \in b_i} d(o, o')^2 / 2$ and

its size $s(i)$ which is the number of initial points in it. A block b_i that contains only one point has $w(i) = 0$ and $s(i) = 1$. Instance-level constraints that remain to be satisfied are only cannot-link constraints. A cannot-link constraint is defined on two blocks b_i, b_j if there exists a cannot-link constraint on two points o, o' such that $o \in b_i$ and $o' \in b_j$.

Using blocks means that in the model of Subsection 5.1, each variable G_i corresponds to a block b_i . All user constraints can be redefined on blocks. For instance, a minimal cardinality constraint states that each cluster should have at least α initial points. To express this constraint, we define an array T , where each variable G_i is repeated $s(i)$ times. The size of T is therefore N and the minimal cardinality constraint has to be expressed by $|\{j \in \{1, \dots, N\} \mid T_j = k\}| \geq \alpha$ for $k \in [1, K]$. Algorithm 3 can also be adapted to take into account size and weight of blocks.

5.3.2 Finding a Feasible Extension

Without user constraints, $\text{Greedy_Extension}(\mathcal{O}_n, \Delta_{n-1}^*)$ is found by adding the new point to the previous best clustering Δ_{n-1}^* . This typically yields a good upper bound, often even being the optimal value. For $\text{Feasible_Extension}(\mathcal{O}_n, \mathcal{C}_n, \Delta_{n-1}^*)$ in Algorithm 2, one has to additionally take the user constraints into account, since the clustering Δ_n must satisfy all \mathcal{C}_n constraints.

We aim at finding a good feasible clustering that satisfies all the user constraints quickly. To achieve this, the same model as described in Subsection 5.1 is used with one restriction, namely that the last $n - 1$ variables G_2, \dots, G_n are assigned to the value they had in clustering Δ_{n-1}^* ; this mimics a greedy strategy as only one variable can be decided, corresponding to adding the point to an existing cluster. If no such extension of the clustering exists, the clustering Δ_n is undefined and its WCSS value is ∞ .

5.3.3 Local vs. Full Constraint Sets

Let \mathcal{C} be the set of all user constraints on the whole set of points $\{o_1, \dots, o_N\}$. There may be instance-level constraints (must-link or cannot-link constraints) or cluster-level constraints (cardinality, density constraints etc.). At each step n , Constrained_BaB finds a clustering that minimizes the WCSS value and that satisfies the set of constraints \mathcal{C}_n . We propose two different ways to define the set \mathcal{C}_n in the constraint solver, following the discussion in Section 4.3.

Local model Let \mathcal{O}_n be the set of points to cluster at step n . The simplest way is to define \mathcal{C}_n by $\mathcal{C}_{\mathcal{O}_n}$, the set of user constraints on a (sub)set of the elements of \mathcal{O}_n . One can see that for $n = N$, $\mathcal{O}_N = \mathcal{O}$ and hence we will consider the set $\mathcal{C}_{\mathcal{O}} = \mathcal{C}$ of all constraints.

Full model To obtain tighter bounds, we can take anti-monotonic constraints into account too. However, we can also use CP capabilities to reason over partial solutions, to let it consider *all* constraints at every step. In this case, at each iteration $n \leq N$, Constrained_BaB operates on the full set of N variables and all the user constraints in \mathcal{C} are considered in the model. However, since we are interested in finding a best clustering on the last n points of G only, the constraint *sum-Squares* is defined only on the last n variables G_{N-n+1}, \dots, G_N . The branching is also on these n variables only.

The interest of such a *full* model is that it can allow to prune earlier cases that cannot be extended to a full solution. Let us take an example with 3 points a, b, c ($N = 3$), $K = 2$ and two cannot-link constraints $CL(a, b)$ and $CL(a, c)$. In step $n = 2$, the two last

points are considered, $\mathcal{O}_2 = \{b, c\}$. The local model that is defined on G_b, G_c has no constraint ($\mathcal{C}_2 = \emptyset$) and will return a clustering Δ_2 where each point is in one cluster. The clustering Δ_2 cannot be used anymore at the next step, where the constraints cannot-link are taken into account. Meanwhile, the full model at each step has the 3 variables G_a, G_b, G_c and two constraints $G_a \neq G_b$ and $G_a \neq G_c$. At step $n = 2$, even though only two variables G_b, G_c are instantiated, the existence of G_a in the model prevents b and c to be in two different clusters, since otherwise $\text{Dom}(G_a) = \emptyset$. The full model can therefore yield better, higher but more realistic, lower bounds for the WCSS attainable in later iterations.

6 EXPERIMENTS

We compare CPRBBA to other state-of-the-art exact clustering approaches: original RBBA³ [6], CPCLustering 2.1⁴ [9] using CP with one phase branch-and-bound search and CCCG-0.5.1⁵ [4] using Integer Linear Programming and column generation. Both unconstrained and constrained settings are considered. We also show the interest of our generic approach by its use in a multi-objective constrained clustering setting, which minimizes the WCSS and maximizes the separation between clusters.

CPRBBA is developed using the Gecode⁶ framework, version 4.3.3. Due to the computational demand of exact clustering we use small but classic datasets from the UCI repository⁷ with the true number of class labels, except for the Hatco dataset [6] which has an unknown number of classes, see Table 1. All experiments are performed on Intel Xeon E3-1225 CPUs running Ubuntu 14.04; a time limit of 30 minutes is used and a memory limit of 4 gigabytes (which is never reached). Codes and examples are available on <http://www.cp4clustering.com>.

6.1 Unconstrained Clustering

As noted before, the performance of (CP)RBBA can change depending on the ordering of the variables used. We compare in Table 1 CPRBBA (local model) with 4 different orderings: order in which the points are read from the input file (input), average of 5 random orderings (random), nearest-neighbor separation as used in RBBA (NNS), and the furthest-point first ordering (FPF). We see that the best ordering can differ from dataset to dataset. In the following, we use the FPF strategy as it has the smallest average runtime.

We now compare CPRBBA to RBBA [6], to CPCLustering using CP [9] and CCCG using column generation [4]. Other unconstrained exact methods have no publicly available implementation, but the respective experiments point to RBBA as being the fastest for small values of k , as is typical in data mining.

The results are shown in Table 2. We can see that both RBBA and CPRBBA are better than the recent CPCLustering and CCCG methods in case no constraints are added, and that the difference in runtime between RBBA and CPRBBA is in accordance to the difference in ordering used as reported in Table 1.

6.2 Clustering with User-Constraints

We compare CPRBBA with CPCLustering and CCCG, supporting also user constraints.

³ <http://www.psiheart.net/QuantPsych/monograph.html>

⁴ <http://www.cp4clustering.com/>

⁵ <https://dtai.cs.kuleuven.be/CP4IM/cccg/>

⁶ <http://www.gecode.org>

⁷ <http://archive.ics.uci.edu/ml/>

dataset	N	K	input	random	NNS	FPF
ruspini	75	4	0.06	0.00	0.01	0.01
soybean	47	4	773.91	10.01	0.80	1.28
hatco	100	2	0.19	0.02	0.07	0.05
hatco	100	3	4.68	0.69	0.55	0.20
hatco	100	4	980.35	556.33	78.37	7.52
hatco	100	5	1800+	1800+	1800+	1636.41
iris	150	3	1800+	0.95	2.30	1.33
wine	178	3	1800+	1800+	16.37	53.57
seeds	210	3	1800+	491.03	1353.26	170.67
breast	569	2	1167.62	1800+	1800+	1800+
average			1012.7	645.9	505.2	367.1

Table 1. Runtimes in seconds of CPRBBA for different point orderings.

	K	CCCG	CPClustering	RBBA	CPRBBA
ruspini	4	1800+	0.41	0.01	0.01
soybean	4	1800+	1.21	0.38	1.28
hatco	2	1800+	1.74	0.03	0.05
hatco	3	1800+	186.18	0.29	0.20
hatco	4	1800+	1800+	53.95	7.52
hatco	5	1800+	1800+	1800+	1636.41
iris	3	1800+	583.19	1.14	1.33
wine	3	1800+	1800+	7.86	53.57
seeds	3	1800+	1800+	542.74	170.67
breast	2	1800+	1800+	1800+	1800+

Table 2. Runtimes in seconds of different exact methods

Instance-level constraints We randomly sampled a number of must-link (ML) and cannot-link (CL) constraints from the true class labels of the datasets. Two points are randomly taken and depending on whether they have the same label or not, a ML or a CL constraint is created. This is repeated until the required ML/CL number is reached.

ML constraints only. We observe in Table 3 that CPRBBA outperforms the other two exact constrained clustering methods, CCCG and CPCLustering. For must-link constraints, there is no difference between using -full or -local models because of the use of must-link blocks. In only one case (a 50-constraint set for the wine dataset), CPRBBA is not able to find a solution within the timeout.

	#c	CCCG	CPClustering	CPRBBA-local	CPRBBA-full
iris	10	1800+ (5)	341.59 (0)	0.81 (0)	0.86 (0)
iris	50	1800+ (5)	135.32 (0)	0.23 (0)	0.25 (0)
iris	100	47.20 (0)	1.20 (0)	0.01 (0)	0.01 (0)
iris	150	0.20 (0)	0.07 (0)	0.01 (0)	0.01 (0)
wine	10	1800+ (5)	1800+ (5)	258.54 (0)	259.30 (0)
wine	50	1800+ (5)	1800+ (5)	363.34 (1)	363.62 (1)
wine	100	1800+ (5)	1800+ (5)	1.19 (0)	1.23 (0)
wine	150	10.60 (0)	18.92 (0)	0.13 (0)	0.13 (0)

Table 3. Runtimes averaged over 5 random samples of #c must-link constraints; between brackets number of runs that timed-out (counted as 1800 seconds in average).

CL constraints only. The results for cannot-link constraints are shown in Table 4. Adding CL constraints can make the problem much harder. Here too CPRBBA outperforms the others, which is in line with the time difference in the unconstrained case. As more constraints are added, an optimal solution can be found in the given timeout for fewer sampled constraint sets (see number between brackets), leading to higher average runtimes.

	#c	CCCG	CPClustering	CPRBBA-local	CPRBBA-full
iris	10	1800+ (5)	727.32 (0)	1.69 (0)	1.79 (0)
iris	50	1800+ (5)	1694.03 (4)	63.94 (0)	64.07 (0)
iris	100	1800+ (5)	497.90 (0)	368.41 (1)	15.40 (0)
iris	150	1800+ (5)	643.72 (1)	721.29 (2)	361.57 (1)
iris	250	1800+ (5)	1094.49 (3)	1080.66 (3)	0.74 (0)
wine	10	1800+ (5)	1800+ (5)	622.89 (1)	625.64 (1)
wine	25	1800+ (5)	1800+ (5)	1310.99 (2)	1326.51 (2)
wine	50	1800+ (5)	1800+ (5)	1697.94 (4)	1706.36 (4)
wine	100	1800+ (5)	1800+ (5)	1800+ (5)	1800+ (5)

Table 4. Runtimes averaged over 5 random samples of #c cannot-link constraints; between brackets number of runs that timed-out (counted as 1800 seconds in average).

These results extend to the combination of must-link and cannot-link constraints (not shown).

Cluster-level constraints Table 5 shows runtimes for different datasets when adding a minimal or a maximal cluster size constraint. We can see that CPRBBA can handle such constraints well, and better than CPCLustering. CPRBBA-full considers more constraints than CPRBBA-local in between iterations, and can hence provide tighter bounds. However, we observe that for some datasets, obtaining tighter bounds requires more search in one iteration to get them, thus loosing the benefits of the tighter bounds in subsequent iterations, and thus leading to overhead. For the iris dataset, the effort of searching for a tighter bound does pay off in the experiments. We observe similar results for a maximum cluster size constraint.

	K	min size	cpclus.	cprbba-local	cprbba-full
ruspini	4	17	1.08	0.02	1.17
ruspini	4	18	270.00	9.00	24.06
soybean	4	10	1.28	1.39	1.78
soybean	4	11	1800+	1563.12	1652.13
iris	3	38	564.86	1.32	1.67
iris	3	42	693.38	9.23	2.45
iris	3	46	933.23	341.23	18.46
iris	3	50	1508.77	1800+	294.75
	K	max. size	cpclus.	cprbba-local	cprbba-full
ruspini	4	20	0.54	0.01	0.05
ruspini	4	19	1800+	602.82	794.83
soybean	4	14	1.28	1.32	1.83
soybean	4	13	17.52	13.19	17.44
iris	3	62	589.92	1.31	1.67
iris	3	58	723.63	3.95	3.04
iris	3	54	973.09	96.78	18.31
iris	3	50	1483.88	1800+	158.75

Table 5. Runtime in seconds for clustering with minimum (top) and maximum (bottom) size constraint

6.3 Multi-Objective Constrained Clustering

Constraints offer a way to find solutions that better fit the problem at hand. Changing the objective function is another way. Curiously, whereas the aim of clustering is to find homogeneous as well as well-separated clusters, most measures, including WCSS, express only homogeneity. One solution is to use multi-objective optimization, with one measure for homogeneity and one for well-separatedness. The result is a set of Pareto optimal solutions, where a Pareto optimal solution is one for which it is not possible to improve the value of one criterion without degrading the value of the other one.

We propose an algorithm (Algorithm 4) to compute an exact set of Pareto solutions for bi-objective WCSS/Split optimization, so as to obtain both homogeneous and well-separated clusterings. It is based on the ϵ -constraint algorithm [22] and is applicable to any complete method that can optimize WCSS under must-link constraints. In this algorithm, constrained single objective optimization (WCSS) is iterated, each time with a condition on the best value of the other objective (minimal split) found so far. This minimal-split constraint can in turn be translated into must-link constraints.

Algorithm 4: Bi-objective WCSS/Split

```

1  $Pareto\_sols \leftarrow \emptyset$ 
2  $min\_split \leftarrow 0$ 
3 repeat
4    $\Delta \leftarrow \text{Minimize\_WCSS}(\mathcal{O}, \{Split > min\_split\})$ 
5    $min\_split \leftarrow Split(\Delta)$ 
6   if  $\Delta$  is not dominated in  $Pareto\_sols$  then
7      $Pareto\_sols \leftarrow Pareto\_sols \cup \{\Delta\}$ 
8 until no  $\Delta$  was found;
```

In [12, 28, 27] the problem of finding the Pareto optimal solutions for minimizing the maximal diameter of the clusters and maximizing the minimal split between clusters is addressed, but without user-constraints. To our best knowledge the only work that handles user-constraints inside a multi-objective clustering problem is [8]. That work does not consider the WCSS criterion, and the criteria used often lead to thousands of equivalent clusterings corresponding to each Pareto point. Algorithm 4 can be easily modified to incorporate user-constraints, in case the Minimize_WCSS algorithm supports it: another set of user-constraints can simply be added to the split constraint at line 4.

Experiments Table 6 presents runtimes in seconds, number of Pareto solutions and the maximal number of clusterings Δ' corresponding to each Pareto solution Δ (i.e. $WCSS(\Delta') = WCSS(\Delta)$ and $Split(\Delta') = Split(\Delta)$). We can see here (last column) that for each Pareto solution, there is always only one corresponding clustering, which contrasts with the thousands of equivalent solutions found in [8] for the Diameter/Split measure.

	K	time (s)	#sols	#c/s
ruspini	4	0.01	1	1
soybean	4	1.58	4	1
hatco	4	32.52	24	1
hatco	5	1979.38	22	1
iris	3	1.11	10	1
wine	3	100.58	9	1
seeds	3	178.62	17	1

Table 6. Runtime, # Pareto solutions, maximal number of clusterings for each Pareto solution

Our framework can also be used for bi-objective WCSS/Split under user constraints. To the best of our knowledge, it is the first method to support this bi-criterion optimization both for instance- and cluster-level constraints. Table 7 shows the results for different use cases on the Iris dataset. For four of these cases, the exact Pareto fronts are shown in Figure 1 (the two cases for 20 ML/CL constraints with and without the minimal size constraint have the

same Pareto front). We can see here the interest of being able to handle user-constraints during the optimization process. Indeed, in this dataset, each ground truth cluster is of size 50, whereas in the unconstrained use case, the Pareto solutions can give clusterings with unbalanced clusters. For instance, the last point in the Pareto front corresponds to a clustering with clusters of size 2, 50 and 98. The constrained cases have the last Pareto solution with $WCSS=86.5396$ and $Split=0.412311$. This solution is common to all the 4 cases, and the only corresponding clustering has clusters of size 49, 50, 51.

Use case	time (s)	#sols	#c/s
unconstrained	1.11	10	1
20 ML/CL	13.68	7	1
40 ML/CL	9.66	8	1
size minimal 38	1.6	7	1
size minimal 40	1.8	4	1
20 ML/CL, size min 40	13.80	7	1
40 ML/CL, size min 40	9.75	8	1

Table 7. Results on Iris for bi-criterion constrained clustering cases

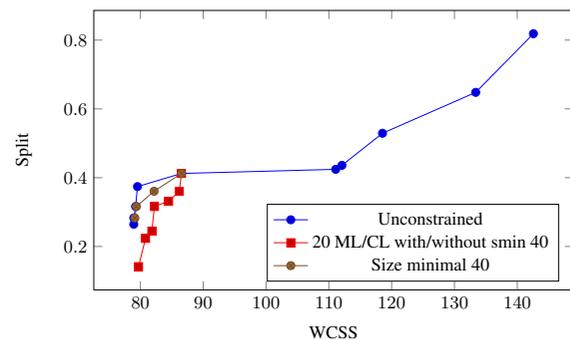


Figure 1. Pareto fronts for different cases on Iris

7 CONCLUSION

In this paper, we address one of the most popular constrained clustering task, the constrained minimum sum-of-squares clustering (MSSC). We extend the Repetitive Branch-and-Bound Algorithm, one of the best method for MSSC without user constraints, to integrate user constraints. The framework we propose is based on Constraint Programming (CP), which is used in each internal branch-and-bound step, as well as in the computation of upper and lower bounds. We propose two different CP models in order to have tight lower bounds and construct a specific propagation mechanism to make better use of the computed bounds. Experiments on classic datasets show that our approach, even though being generic, is competitive compared to a dedicated implementation of RBBA in the unconstrained case. For constrained cases, our approach outperforms the existing state-of-the-art exact approaches. Furthermore, we show how its generality allows it to be used in a bi-objective constrained clustering setting.

To further enhance the efficiency of the framework, one may have to consider other ordering heuristics, including dynamic ones. Moreover, RBBA has been applied to clustering tasks with other optimization criteria such as WCSS, to which our approach can be extended as well. Our bi-objective approach can also be used with non-exact constrained clustering methods, though the resulting Pareto front will be an approximation. Lastly, a mix of Russian Doll Search and our approach may lead to advances for both valued CSPs and clustering.

REFERENCES

- [1] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat, 'NP-hardness of Euclidean Sum-of-squares Clustering', *Machine Learning*, **75**(2), 245–248, (2009).
- [2] Daniel Aloise and Pierre Hansen, 'An branch-and-cut SDP-based algorithm for minimum sum-of-squares clustering', *Pesquisa Operacional*, **29**(3), 503–516, (2009).
- [3] Daniel Aloise, Pierre Hansen, and Leo Liberti, 'An improved column generation algorithm for minimum sum-of-squares clustering', *Mathematical Programming*, **131**(1-2), 195–220, (2012).
- [4] Behrouz Babaki, Tias Guns, and Siegfried Nijssen, 'Constrained clustering using column generation', in *Proceedings of the 11th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pp. 438–454, (2014).
- [5] Michael J. Brusco, 'An enhanced branch-and-bound algorithm for a partitioning problem', *British Journal of Mathematical and Statistical Psychology*, **56**(1), 83–92, (2003).
- [6] Michael J. Brusco, 'A repetitive branch-and-bound procedure for minimum within-cluster sums of squares partitioning', *Psychometrika*, **71**(2), 347–363, (2006).
- [7] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain, 'A Declarative Framework for Constrained Clustering', in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 419–434, (2013).
- [8] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain, 'Constrained clustering by constraint programming', *Artificial Intelligence*, DOI: 10.1016/j.artint.2015.05.006, (2015).
- [9] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain, 'Constrained minimum sum of squares clustering by constraint programming', in *Principles and Practice of Constraint Programming, CP 2015, Proceedings*, pp. 557–573, (2015).
- [10] Ian Davidson and S. S. Ravi, 'Clustering with Constraints: Feasibility Issues and the k-Means Algorithm', in *Proceedings of the 5th SIAM International Conference on Data Mining*, pp. 138–149, (2005).
- [11] Ian Davidson, S. S. Ravi, and Leonid Shamis, 'A SAT-based Framework for Efficient Constrained Clustering', in *Proceedings of the 10th SIAM International Conference on Data Mining*, pp. 94–105, (2010).
- [12] M. Delattre and P. Hansen, 'Bicriterion cluster analysis', *IEEE Trans. Pattern Anal. Mach. Intell.*, (4), 277–291, (1980).
- [13] O. du Merle, P. Hansen, B. Jaumard, and N. Mladenovic, 'An interior point algorithm for minimum sum-of-squares clustering', *SIAM Journal on Scientific Computing*, **21**(4), 1485–1505, (1999).
- [14] A. W. F. Edwards and L. L. Cavalli-Sforza, 'A method for cluster analysis', *Biometrics*, **21**(2), 362–375, (1965).
- [15] T. Gonzalez, 'Clustering to minimize the maximum intercluster distance', *Theoretical Computer Science*, **38**, 293–306, (1985).
- [16] Pierre Hansen and Brigitte Jaumard, 'Cluster analysis and mathematical programming', *Mathematical Programming*, **79**(1-3), 191–215, (1997).
- [17] Robert E. Jensen, 'A dynamic programming algorithm for cluster analysis', *Journal of the Operations Research Society of America*, **7**, 1034–1057, (1969).
- [18] W. L. G. Koontz, P. M. Narendra, and K. Fukunaga, 'A branch and bound clustering algorithm', *IEEE Trans. Comput.*, **24**(9), 908–915, (1975).
- [19] Marianne Mueller and Stefan Kramer, 'Integer Linear Programming Models for Constrained Clustering', in *Proceedings of the 13th International Conference on Discovery Science*, pp. 159–173, (2010).
- [20] Dan Pelleg and Dorit Baras, 'K-means with large and noisy constraint sets', in *Machine Learning: ECML 2007*, volume 4701 of *Lecture Notes in Computer Science*, pp. 674–682. Springer Berlin Heidelberg, (2007).
- [21] Douglas Steinley, 'k-means clustering: A half-century synthesis', *British Journal of Mathematical and Statistical Psychology*, **59**(1), 1–34, (2006).
- [22] Vincent T'kindt and Jean-Charles Billaut, *Multicriteria Scheduling, Theory, Models and Algorithms*, Springer, 2nd edn., 2005.
- [23] B.J. van Os and J.J. Meulman, 'Improving Dynamic Programming Strategies for Partitioning', *Journal of Classification*, (2004).
- [24] Gérard Verfaillie, Michel Lemaître, and Thomas Schiex, 'Russian doll search for solving constraint optimization problems', in *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96*, pp. 181–187, (1996).
- [25] K. Wagstaff and C. Cardie, 'Clustering with instance-level constraints', in *Proceedings of the 17th International Conference on Machine Learning*, pp. 1103–1110, (2000).
- [26] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl, 'Constrained K-means Clustering with Background Knowledge', in *Proceedings of the 18th International Conference on Machine Learning*, pp. 577–584, (2001).
- [27] J. Wang and J. Chen, 'Clustering to maximize the ratio of split to diameter', in *Proceedings of the 29th International Conference on Machine Learning*, (2012).
- [28] Y. Wang, H. Yan, and C. Sriskandarajah, 'The weighted sum of split and diameter clustering', *Journal of Classification*, **2**(12), 231–248, (1996).
- [29] Y. Xia and J. Peng, 'A cutting algorithm for the minimum sum-of-squared error clustering', in *SDM*, (2005).

Is Spearman’s Law of Diminishing Returns (SLODR) Meaningful for Artificial Agents?

José Hernández-Orallo¹

Abstract. The progress of artificial intelligence is reaching a point that some research questions that were only relevant for human and other animal agents are becoming relevant for artificial agents as well. One of those questions comes from human intelligence research and is known as Spearman’s Law of Diminishing Returns (SLODR). Charles Spearman, the father of factor analysis and the g factor (a dominant factor explaining most of the variance in cognitive tests for human populations), observed that when the analysis was restricted to the subpopulation of most able subjects, the relevance of this dominant factor diminished, as if the power of general intelligence were saturated or not fully used by the most able individuals. In about a century, there have been numerous theoretical explanations and experiments to confirm or reject Spearman’s hypothesis. However, all of them have been based on human or animal populations. In this paper, we analyse for the first time whether the SLODR makes sense for artificial agents and what its role should be in the analysis of general-purpose AI. We use a synthetic scenario based on modified elementary cellular automata (ECA) where the ECA rules work as tasks and the population of agents is generated with an agent policy language. Different slices of the population by ability and of the tasks by difficulty are analysed, showing that SLODR does not really appear. Indeed, even if very slightly, we find the reverse, i.e., that more correlation takes place for more able subpopulations, what we conjecture as the Universal Law of Augmenting Returns (ULOAR).

1 INTRODUCTION

While more and more specific applications are being successfully solved by AI systems, the field is also progressing in the development of systems that are able to solve a wider range of problems, usually after a long training. Reinforcement learning [49], cognitive developmental robotics [6] and machine learning in general incarnate—or are integrated into—autonomous agents that solve a range of tasks. Some recent developments have displayed significant performance in a variety of tasks, at least in some domains. For instance, [39] combine reinforcement learning and deep learning to attempt a diverse set of Atari 2600 videogames. Although the system has to relearn from scratch when the game changes slightly, it is still a general-purpose technique, which can be evaluated for a range of tasks.

Despite the great number of competitions in AI for particular applications, some competitions and benchmarks are also moving in the direction of more general-purpose systems (see [23] for a full account). These include the general game playing AAAI Competition [17, 18], the reinforcement learning competition [54, 10] (including, e.g., the ‘polyathlon’, with several domains), the genetic programming benchmarks [38, 53], the general video game competition

[43, 42], and the arcade learning environment [2, 43] (including the Atari 2600 videogames mentioned above).

However, there are some questions that arise when one considers a wide range of tasks, both when designing systems to behave well for them or when designing tests to evaluate these systems. This discussion is especially controversial when one wants to consider *all possible tasks*. On one hand, if one considers every possible problem’s output as equally likely (technically known as ‘block uniformity’ [29], with the uniform distribution being a special case) then we have the conditions for the so-called no-free-lunch theorems [57, 56], leading to the conclusion that, on average, no method can be better than any other. According to this, a general-purpose system and, indeed, the very concept of ‘general intelligence’ would be impossible [14]. On the other hand, if one considers problems as programs, then a uniform distribution is not possible. Instead, any universal distribution can be assumed, which leads to the theory of universal prediction using algorithmic probability developed by Solomonoff in the 1960s [44, 45]. This has influenced several approaches based on algorithmic information theory about how tasks can be generated and weighted in definitions of intelligence [11, 19, 33, 25, 13, 30] and how theoretically general agents can be defined [28], only if weakly optimal or suboptimal in general [41, 34]. Nevertheless, the idea of general intelligence makes sense theoretically in this context: some agents can be better than others in general.

The experimental and theoretical analysis of AI agents that are devised and evaluated for a range of tasks has led to an approaching to some similar ideas from the area of human intelligence evaluation. The use of IQ tests for the evaluation of AI systems has been advocated for by some [5, 4] but it has been criticised by others for being anthropocentric [12, 27]). But other concepts and tools from psychometrics, such as item response theory and the use of task difficulty to analyse the landscape of problems, are being vindicated in artificial intelligence as well, under the term universal psychometrics [26, 24]. In fact, one of the problems of the use of a universal distribution of tasks for defining the general problem of intelligence can be addressed differently if one considers a uniform distribution of difficulties, a uniform distribution of policies per difficulty with finally leaving the universal distribution to the conditional probability of a task given an acceptable policy [21]. This replaces the notion of a task-general intelligence (the so-called universal intelligence) as addressing a diversity of tasks to that of a policy-general addressing a diversity of solutions, expressed under a policy description language.

This debate replicates the controversy in psychometrics between the IQ scores (results of the IQ tests, which depend on the task distribution used in a test) and the g scores (a magnitude derived from the estimated value of a latent factor, the g factor, which is more independent to the particular task distribution used in a test). The g factor

¹ DSIC, Universitat Politècnica de València, email: jorallo@dsic.upv.es

derives from the so-called positive manifold, one of the most replicated experimental findings in the analysis of human intelligence. The positive manifold indicates that given any test composed of a set of (abstract) cognitive tasks we will find a high correlation in the results produced by a human population. In other words, those who perform well on some tasks will usually perform well on any other. This supports the idea of general intelligence.

For artificial intelligence, this suggests the following question: if we aim at building more general AI systems, will it be the case that those that are better for some tasks will also be better for other tasks? Is that a necessary or a contingent *property*? In order to study this question, however, there is an important consideration: two equally-general systems may have different levels of ability (or general intelligence). It is for those that are more intelligent we expect this property to hold stronger. To put an extreme negative case, a random agent is completely general, but not intelligent. We do not expect this property to hold for random agents, despite their 'generality'. This suggests that the positive manifold will only start to be observed for artificial agents when we have a population of minimally intelligent agents. As long as AI progresses towards more generally intelligent agents, this positive manifold would start to appear and then become stronger. Surprisingly, in the realm of human intelligence, Spearman found exactly the reverse observation. By taking subpopulations of more able humans, the positive manifold was weaker, something that was later known as Spearman's Law of Diminishing Returns (SLODR).

In this paper, we introduce a simple, but effective, setting to analyse these questions for artificial agents. We adapt a class of tasks consistent of elementary cellular automata (ECA) where we have introduced an agent that interacts within these worlds (the ECA rules), as done in [20]. Using this simple world and an elementary agent language, we can analyse *all tasks* and all possible policies (solutions) up to a certain size (determining the difficulty of the policy), so really having a diversity of solutions to analyse whether some degree of positive manifold appears. More interestingly, we can easily analyse different subsets according to their average performance on all policies (or slices of appropriate difficulty) and study whether the SLODR holds or not.

The rest of the paper is organised as follows. Section 2 reviews the notions of positive manifold, g factor and Spearman's Law of Diminishing Returns (SLODR) and some of the explanations and experiments performed to support or reject the law. Section 3 introduces the environments and agents used for the analysis, describing how they work and showing a few examples. Section 4 performs two different experiments with the goal of discovering whether the SLODR holds or not. Section 5 discusses the results and its implications. Finally, Section 6 closes the paper with new questions and future work.

2 SPEARMAN'S LAW OF DIMINISHING RETURNS (SLODR)

There are many kinds of cognitive tests that can be applied to humans. Some of them compose the popular IQ tests, whose development started about a century ago. Charles Spearman was one of the pioneers of a numerical analysis of human intelligence, by compiling the results of several tests on human populations. He started to use the recently introduced notion of correlation to analyse the results. He found one important phenomenon: when he analysed a set of different tests taken by the same population, he found a positive average correlation in their results. In other words, the individuals that obtained good results for some tests usually obtained good results

for the rest. This correlation was stronger the more culture-fair and abstract the tests were. This phenomenon was known as the 'positive manifold' [46, 47].

It is important to clarify that this phenomenon is not a property of the tests alone nor a property of the population alone. A correlation is clearly an effect that takes place for two subjects for a set of tests, but the average correlation is calculated from the correlation matrix, thereby involving both the population and the tests. Nevertheless, the positive manifold appeared again and again for different human populations and different sets of tests, provided they were not too linked to particular cultural or educational backgrounds (e.g., a chess-playing test and a Korean vocabulary test). Spearman tried to understand the findings through the invention of a rudimentary factor analysis. He identified a dominant *latent factor* that explained much of the variance, and called it the g factor. Since then, this factor has been one of the most relevant (and replicated) findings in psychometrics [31, 48] and has been found to predict many facets of life, from academic performance to (lack of) religiosity in humans.

The dominance of g and its explanatory character for the positive manifold led to the association of g with general intelligence, a latent factor that pervaded or dominated all other factors and facets of intelligence. Of course, this interpretation has been challenged many times, even if g appears again and again.

Still more controversial than the interpretation of the g factor is another finding that Spearman discovered. He calculated the strength of g on subpopulations of different abilities. In particular, in one of the analysis, he separated the results of several tests on a human population into two groups, group A with normal abilities and B with low abilities. After the split, he analysed the correlation matrices separately. The result was that the mean correlations for group A were 0.47 but the mean correlations for group B was 0.78. Note that this does not mean that group A had worse results (in fact, it was precisely the group with highest average results), but rather that the *proportion of the variance* explained by g for the low-ability group was much higher than for the normal-ability group. This result was striking, especially if g is understood as general intelligence. It looked as if the more intelligent a population is, the less important g would be, in relative terms, to explain its variability. This observation turned to be known as Spearman's Law of Diminishing Returns (SLODR). The finding was replicated many times since then with different experimental settings [9, 8, 50].

Spearman looked for an explanation and found it in the *law of diminishing returns* in economics. Many processes that are affected by many factors do not grow continuously as the result of the increase of one factor, so the influence of a single, albeit dominant, factor can become less relevant at a given point, being saturated. Spearman expressed it in this way: "the more 'energy' a person has available already, the less advantage accrues to his ability from further increments of it" [47, p. 219].

But this simile was not an explanation. Spearman postulated the "ability level differentiation", which considered that challenging items (those that can only solve the more able individuals) require the combination of many skills, and the prevalence of g would be slower. Basically, for the easy items, the general intelligence or some general resources would be the only available skills for low-ability subpopulations. Detterman and Daniel [9] argued similarly that "central processes are deficient, they limit the efficiency of all other processes in the system. So all processes in subjects with deficits tend to operate at the same uniform level. However, subjects without deficits show much more variability across processes because they do not have deficits in important central processes". Other explana-

tions were introduced, such as that the “genetic contribution is higher at low-ability levels” [8].

On the other hand, not only the above explanations but the experimental evidence itself have been contested. One common counter-explanation of the phenomenon argues that it is not that g is less important for able subjects, but that they find many of the problems in the tests less challenging than the normal population and then they are not forced to use general intelligence as they can solve the problems without deep thinking, i.e., more mechanically. In other words, the use of the same tests for both groups would be creating the effect. In fact, Fogarty and Stankov [16] performed an experiment where the more able group had to solve problems of higher difficulty whereas the less able group had to solve problems of lower difficulty. Under these conditions SLODR did not only appear but even the more able group showed higher correlations! This seems to agree with the idea that general intelligence is used when the individual finds a problem challenging. It is important, however, to check that the difficult problems are created without the use of spurious complications, in order to prevent that more difficult items are more specialised than the simple items. For instance, in number series problems, one can create a complex series by using the Fibonacci series. This, however, will just assess whether the subject has some particular mathematical knowledge, not really expecting that the subject is going to discover the Fibonacci series from scratch. This was already warned by Jensen, pointing out “that it is the highly g -loaded tests that differ the least in their loadings across different levels of ability, whereas the less g -loaded tests differ the most” [32]. Usually, problems featuring abstract thinking (inductive inference, analogies, etc.) are those with higher g loadings.

Nevertheless, one of the most relevant criticisms (or explanations), which will reappear later on in this paper, had a more statistical character. Jensen [31, p. 587] argued that the subgroups with higher abilities had lower variance than the subgroups with lower variance. This may be caused by the way the tests are designed to cover a wide range of subjects or the way the two groups are split, but the different variances were generally the case. As a consequence, the relative relevance of g would be lower for more able groups as there is less variance to explain.

All of the above suggests that there are several methodological problems about the analysis of SLODR in human intelligence, starting from putting into question all results for which both groups do not have the same variance and also those that include spurious problems or sample the populations in ways to get the same variance by introducing some other confounding factors. In the end, Murray et al. argue that SLODR could just be “a statistical artifact” [40].

In what follows we take a different perspective of the debate by using artificial tasks and artificial subjects. This can help us to rule out some of the confounding factors by focussing on a controlled experiment, where we can play with the population of agents and the choice of tasks more freely. Nevertheless, our interest is to analyse whether SLODR happens or not for artificial agents, and see whether the results can tell us something about the construction and evaluation of general-purpose AI agents.

3 A SETTING FOR ARTIFICIAL TASKS AND AGENTS

In this section, we are going to adapt the simple setting introduced in [20]. This is an appropriate scenario for practical reasons. First, it is more illustrative to use minimalistic environments where the number of observations and actions are extremely reduced, while still

having some relatively rich phenomena with very simple transition functions. Second, we are interested in simplistic policy languages in order to be able to evaluate a large amount of agents quickly.

3.1 Agent-populated elementary cellular automata: definition and examples

The environments we will work with are composed of an elementary cellular automaton (ECA) [55] for the space S and the transition function τ , but we will let an agent see and modify part of the usual behaviour of the automaton. The following definition specifies the complete behaviour of this kind of environment:

Definition 1 A single-agent elementary cellular automaton (SAECA) is a tuple $\langle S, \tau, \rho, \pi, \vec{\sigma}^0, \nu, p^0 \rangle$. The state space S is represented by a one-dimensional array of bits or cells $\vec{\sigma} = \langle \sigma_1, \sigma_2, \dots, \sigma_m \rangle$, also known as configuration. We consider the array to be finite ($|\vec{\sigma}| = m$) but circular in terms of neighbourhood ($\sigma_0 = \sigma_m$ and $\sigma_{m+1} = \sigma_1$). There is an initial array $\vec{\sigma}^0$, also known as seed. The transition function τ is given by a number ν , as any of the $2^{2^3} = 256$ rules that can be defined looking at each cell and its two neighbours according to the numbering scheme convention introduced in [55]. For instance, the following transitions for each triplet define an ECA rule:

111	110	101	100	011	010	001	000
0	1	1	0	1	1	0	1

The digits of the second row represent the new state for the middle cell after each transition, depending on the triplet. In the above case, 01101101, in binary, corresponds to decimal number 109, the ECA rule number with Wolfram's convention. Given this rule, the array 01100 would evolve in the following way, looping at the end:

01100
01101
01111
11001
01001
11001



Given the behaviour of the space, we consider just one agent π . The agent is located at one cell (its position p) with $1 \leq p \leq m$, which is initially p^0 . The set of observations \mathcal{O} is given by two bits $\langle \sigma_{p-1}, \sigma_{p+1} \rangle$ representing the contents of the left and right neighbouring cells respectively, i.e., σ_{p-1} and σ_{p+1} . The actions \mathcal{A} are given by a ‘move’ and an ‘upshot’, denoted by the pair $\langle V, U \rangle$. The ordered set of moves is given by $\{ \text{left}=0, \text{stay}=1, \text{right}=2 \}$, and the ordered set of upshots is $\{ \text{keep}=0, \text{swap}=1, \text{set0}=2, \text{set1}=3 \}$, which respectively mean that the content of the cell where the agent is does not change, the content of the cell is swapped ($0 \rightarrow 1, 1 \rightarrow 0$), the content is set to 0 and the content is set to 1. The rewards are calculated in the following way. If the agent is at position p at time t , then we use this formula:

$$r^t \leftarrow \sum_{j=1..[m/2]} \frac{\sigma_{p+j}^t + \sigma_{p-j}^t}{2^{j+1}}$$

which counts the number of 1s which are in the neighbourhood of the agent, weighted by their proximity. It is easy to see that $0 \leq r^t \leq 1$. Basically, the goal of the agent is to be surrounded by the highest number of 1s possible, by creating them or by exploiting the changes performed by the ECA rule.

The order of events for each step in the system is: observations are produced, actions are performed, the automaton is updated and finally, rewards are produced.

Note that the environment is parametrised by the original contents of the array σ^0 , the ECA rule number ν , and the original position of the agent p^0 . Given an environment and a computable agent, the evolution of the system is computable and deterministic.

Let us see a few examples of how these environments work. Figure 1 shows the evolution of several environments with seed "01010101010101010101", and several values of ν . We do not include any agent in the trials in this first figure. As a result, the space-time diagram after 200 iterations is the same as a classical elementary cellular automaton with each number ν (see, e.g., [55]).

3.2 Including agents: an agent policy language

Let us now explore what happens when we include agents in these environments. We new a language for expressing the agents. There are a few agent languages in the literature (see, e.g., [3, 35, 1]), but they are too oriented towards the architecture, are too focussed on Markov Decision Processes or are not sufficiently minimalistic for bounding their size and having some interesting programs. We present a very minimalist language, also taking into account the minimalist environment.

Definition 2 *The agent policy language APL is given by a memory (or history) binary array mem , initially empty (and not circular), and an ordered set of instructions $\mathcal{I} = \{ \text{back}=0, \text{fwd}=1, \text{Vaddm}=2, \text{Vadd1}=3, \text{Uaddm}=4, \text{Uadd1}=5 \}$. The numbers on the right will be used as shorthand for the instruction. For instance, the string 22142335 represents a program in APL. A program or policy π is a sequence of instructions $\iota_1, \iota_2, \dots, \iota_{|mem|}$ in \mathcal{I} . The interpreter works on its memory by using two accumulators V and U , and the action is given by the result of the accumulators at the end of the process. Namely:*

1. Read the observation $\langle \sigma_{p-1}, \sigma_{p+1} \rangle$ and its elements being appended to the history array mem .
2. Place the memory pointer b at the end of mem .
3. $V \leftarrow \text{stay}$
4. $U \leftarrow \text{keep}$
5. forall $\iota \in \pi$
6. case ι :
7. back : $b \leftarrow \max(b - 1, 1)$
8. fwd : $b \leftarrow \min(b + 1, |mem|)$
9. Vaddm : $V \leftarrow (V + mem_b) \bmod 3$
10. Vadd1 : $V \leftarrow (V + 1) \bmod 3$
11. Uaddm : $U \leftarrow (U + mem_b) \bmod 4$
12. Uadd1 : $U \leftarrow (U + 1) \bmod 4$
13. end case
14. endfor
15. return $\langle V, U \rangle$

Let us see an example. If an agent is located at the fifth position of the configuration 000110111 and has a current history $mem = 111010$ then the observations 1 and 0 will be appended to mem , leading to $mem = 11101010$. If the policy 20242335 is applied, we start with $b = 8$, $V = 0 = \text{stay}$ and $U = 0 = \text{keep}$, and we have the following execution:

1. $\iota_1 = 2 = \text{Vaddm}, V \leftarrow (V + mem_8) \bmod 3 = 1 = \text{stay}$.
2. $\iota_2 = 0 = \text{back}, b \leftarrow \max(8 - 1, 1) = 7$.
3. $\iota_3 = 2 = \text{Vaddm}, V \leftarrow (V + mem_7) \bmod 3 = 2 = \text{right}$.
4. $\iota_4 = 4 = \text{Uaddm}, U \leftarrow (U + mem_7) \bmod 4 = 1 = \text{swap}$.
5. $\iota_5 = 2 = \text{Vaddm}, V \leftarrow (V + mem_7) \bmod 3 = 0 = \text{left}$.

6. $\iota_6 = 3 = \text{Vadd1}, V \leftarrow (V + 1) \bmod 3 = 1 = \text{stay}$.
7. $\iota_7 = 3 = \text{Vadd1}, V \leftarrow (V + 1) \bmod 3 = 2 = \text{right}$.
8. $\iota_8 = 5 = \text{Uadd1}, U \leftarrow (U + 1) \bmod 4 = 2 = \text{set0}$.

After this program, which is run internally, we obtain the action that the agent will perform on the environment, which is given by $\langle V, U \rangle = \langle 2, 2 \rangle = \langle \text{right}, \text{set0} \rangle$. This means that the agent will move right and set the content of the cell to 0.

While the class of policies generated by this language is infinite, the language is still not universal, and all (finite) programs end. The goal of this language is to be able to express some simple policies that may be useful in the environment.

Figure 2 shows how the environment with elementary cellular automaton number 110 varies for several agent policies. The resulting space-time diagram patterns are different. Similar things (where differences are more visible with respect to the corresponding diagram in Figure 1) happen with rule number 164 (Figure 3).

We define \mathbb{R} as the (expected) response (the result) of agent π in task μ , which is calculated as an average of the rewards r^t for the 200 steps t . For instance, in Figure 3 policy 23555 for rule 164 seems to have higher \mathbb{R} than policy 24 for the same rule.

After introducing the environments (tasks) and agents (policies), in the following section we explore SLODR using subpopulations.

4 ANALYSIS OF SUBPOPULATIONS

Using the agent policy language APL defined above we generated 400 agents with their instructions chosen uniformly from the instruction set and a program length also uniformly distributed between 1 and 20. We evaluated each agent with all the 256 possible ECA rules, with 21 cells, fixed initialisation (seed) of "01010101010101010101", using 100 of iterations per trial.

4.1 Experiment 1: confounding factors

From the 256×400 results, we scaled them task per task so that for each task (ECA rule) we had mean 0 and standard deviation 1. As we will work with Pearson (linear) correlations, this scaling does not affect the correlations, but allows a better aggregation to determine the abilities of each agent. Also from the results, we calculated the 256×256 correlation matrix for the 256 rules. From all the correlations ($\frac{256 \times 255}{2} = 32640$), 29612 were positive. The average correlation was 0.146. Then we averaged the results for each agent to get their average score. We sorted agents per score and split the agent population according to different quantiles (from best to worst). This is shown in Figure 4.

Different size of the bins (subpopulations) were used for the quantiles. On the left figure, the black cross on top represents one bin with the whole population (400 agents), with an average correlation of 0.146, as said above. The second shape (using orange triangles) is formed by the 51 possible bins using agents 1..350, 2..351, ..., 51..400. For smaller bins underneath we see that the average correlation decreases (in other colours). If we look at the concave shapes we clearly see that the average correlation is not the same for the whole range, with smaller values for middle quantiles (around 0.5 on the x -axis). In fact, we see correlations are higher for the more able group (high performance, lower quantiles) and the less able group (low performance, higher quantiles).

Trying to interpret these first results, we can recall Jensen's criticism in section 2. What we observe can be easily explained by the choice of best or worst subsamples. These have a tendency to agree

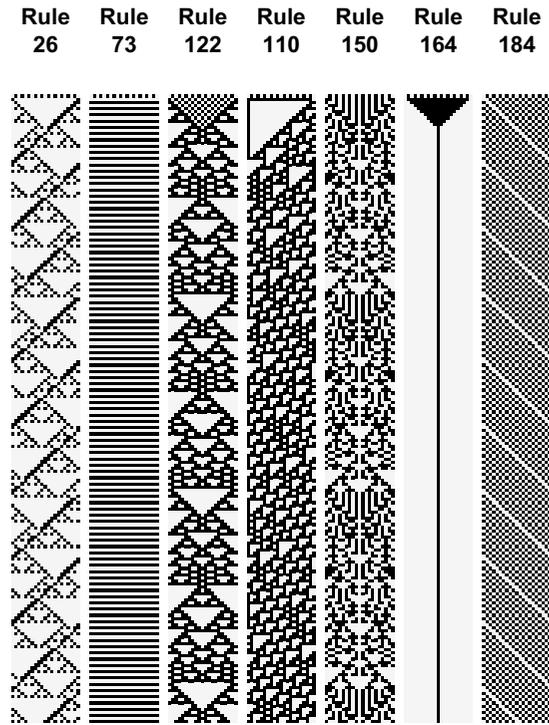


Figure 1. Space-time diagram (evolution for $t = 200$ steps) of several elementary cellular automata without agent. The initial array (seed) is always 0101010101010101010, whose length is 21 bits.

on more tasks and should not be interpreted as any common factor. In fact, the right plot shows the variance for each bin, which explains most of what happens on the left plot.

Nevertheless, this first experiment still shows several things. Using some quantiles split by performance, there are important differences. Of course this is not supporting any law of diminishing returns for the middle quantiles, but just a consequence of the different variances, as argued by Jensen. Also interesting is the fact that we get some positive, albeit small, average correlations, even if we are using randomly-generated agents for all possible tasks (all ECA rules). This is given by the reward mechanism, which is the same for all tasks (having 1s in the surrounding cells) and there are some agents that go well for this reward criterion disregarding the task.

In order to analyse the relevance of the reward criterion, we perform a second experiment where the reward mechanism is being mirrored half of the times (so agents cannot specialise to it). By mirroring we mean changing the sign of the reward, so now the goal is to be surrounded by as many 0s as possible. Also, the agent policy language is modified so that agents can now see the rewards. These small changes lead to very important changes in Figure 5 (left), where we now used 256 agents instead of 400. The top black cross uses all tasks (256) and all agents (256) together. The second shape (orange triangles) shows 17 bins, using agents 1..240, 2..241, ..., 17..256, and so on for the other shapes, according to the sizes shown in the legend.

4.2 Experiment 2: variance and difficulty

The average correlation almost disappears. It is now just 0.004. Again, Figure 5 (left) slices the agents in bins by their average abilities and we have shapes that are similar to the previous experiment.

However, we now do an extra change in our analysis. Figure 5

(right) also slices the tasks by *difficulty*. We evaluate the more able agents with more difficult tasks. In order to do this, we calculate difficulty of a task following [22], where we simplify the estimation of difficulty here by only considering the length of the policies (and not the execution steps as all policies have a finite execution time):

$$\bar{h}^{[\epsilon]}(\mu) \triangleq \min_{\pi \in \mathcal{A}^{[\epsilon]}(\mu)} L(\pi) \quad (1)$$

i.e., the difficulty of a task μ is the length of shortest policy π that is acceptable for the task. Note that this is not the Kolmogorov complexity of the task (i.e., the shortest description for the task) but rather the shortest description of any (acceptable) solution for the task. Acceptability is defined using a tolerance ϵ :

$$\mathcal{A}^{[\epsilon]}(\mu) \triangleq \{\pi : \mathbb{R}(\pi, \mu) \geq 1 - \epsilon\} \quad (2)$$

i.e., the set of all acceptable policies for a task μ is given by those policies whose expected response is above a threshold, given by the tolerance ϵ . Recall that we defined expected response \mathbb{R} as the average reward result of agent π in task μ .

Given this approximation to difficulty, we chose tolerance to be the response that separates the 10% best agents for each task and we sliced tasks by difficulty, using the same bin size than for the agents. As we only generated 256 agents in this experiment, the sizes where also the same. In summary, Figure 5 (right) shows different shapes. As mentioned above, the top black cross uses all tasks (256) and all agents (256) together, with 0.004 correlation, and the second shape (orange triangles) shows 17 bins, using agents 1..240, 2..241, ..., 17..256, and so on. But now, for each of the bins, we also slice the problems (tasks) according to their difficulty. For instance, for the first bin of the orange triangles, the most able agents 1..204, we calculate the correlation with only the most difficult tasks 1..204.

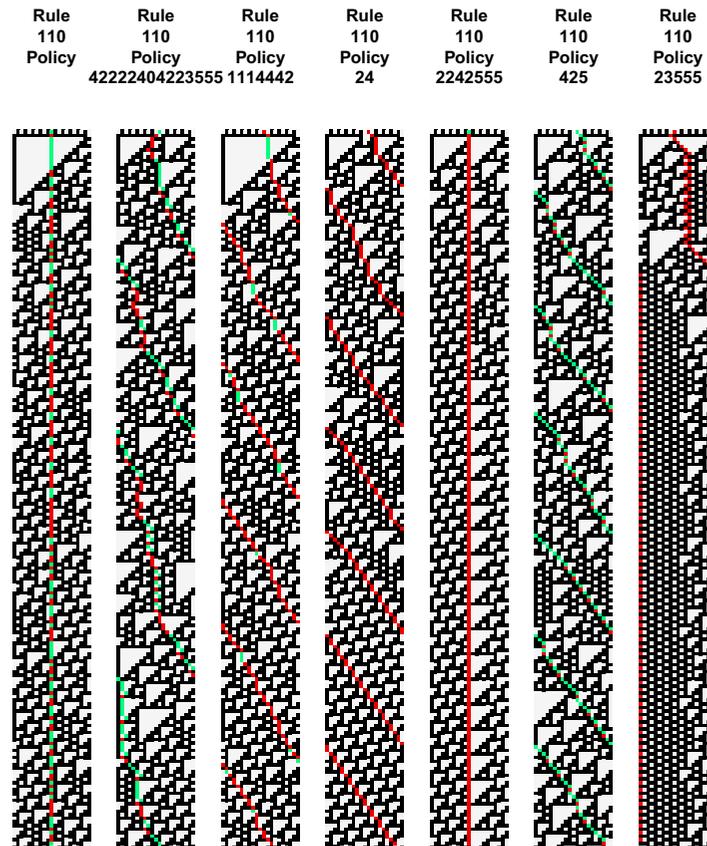


Figure 2. Space-time diagram (evolution for $t = 200$ steps) with different agent policies for the elementary cellular automaton with rule 110. The initial array (seed) is always 0101010101010101010, whose length is 21 bits. The agent is represented by a red dot when the cell has a 0 (like the white ones) and by a green dot when the cell has a 1 (like the black ones). The leftmost diagram is the empty policy ($\langle\langle\text{stay, keep}\rangle\rangle$).

The slicing by ability and corresponding difficulty for each group now shows a very different picture. Figure 5 (right) shows some slope in the distribution of results, where we find higher correlations for higher abilities (lower quantiles). This is exactly the reverse of Spearman's Law of Diminishing Returns.

5 DISCUSSION

Before jumping into any conclusion, let us first analyse the results of this particular experiment. We are generating agents with a very simple policy language. Still, it can now access the rewards and compute actions with them so that meaningful policies are generated. For instance, the policy that repeats the previous action if the reward is good and do another action otherwise can be coded with a relatively short program in this language. Nevertheless, we cannot expect any agent that is especially good. Accordingly, many agents are completely lost in the environments. However, it is precisely a basic scenario we wanted to explore first, resembling some kind of minimal artificial life situation where we can consider all agents up to a certain complexity and see if any correlation appears, even if small. The simplicity of the policies was also useful for a second, very important thing. Difficulties are estimated from first principles also using the agent policy language. As seen in eq. 1, difficulty is calculated as the length of the shortest acceptable policy. This can only be estimated in a reasonable amount of time with standard hardware if programs do not get very large. Of course, the simple scenario leads to very small correlations, since we have very simple agents, and even

very small correlations (once the rewards were mirrored), but the results are consistent to a low expectation about the abilities of these agents. This low correlation is also consistent to the very intuition under ULOAR.

However, the interesting point is that we can study task correlation in a very controlled experiment and find some trends by slicing per ability and difficulty. If we focus on the more able agents, it is not that they are just better for a random sample of tasks, but that they have a slight higher chance of getting more difficult problems right. The positive manifold starts appearing, the embryo of some kind of general ability may be appearing here. This suggests the hypothesis that given a population of agents, the more generally intelligent and diverse they are, the stronger the positive manifold will be. We can call this hypothesis the **universal law of augmenting returns** (ULOAR) as the opposite to SLODR. In fact, as argued before, the ULOAR makes more sense for AI, there is no reason to think that for artificial agents we may find some kind of saturation, once the tendency is initially found at very low degrees of general ability.

Of course, we cannot extrapolate from a single experiment that just shows a slightly higher (yet very small) correlation for the more able groups, but this contributes to the intuition that, for artificial agents, SLODR may not hold in general. Also, we cannot extrapolate this for human populations, and it is still unclear whether SLODR holds for humans or, more precisely, whether it holds for some human populations with some distributions of tests. But when one conceives artificial agents of a wide range of resources and algorithms, the existence of SLODR looks very counterintuitive in hindsight.

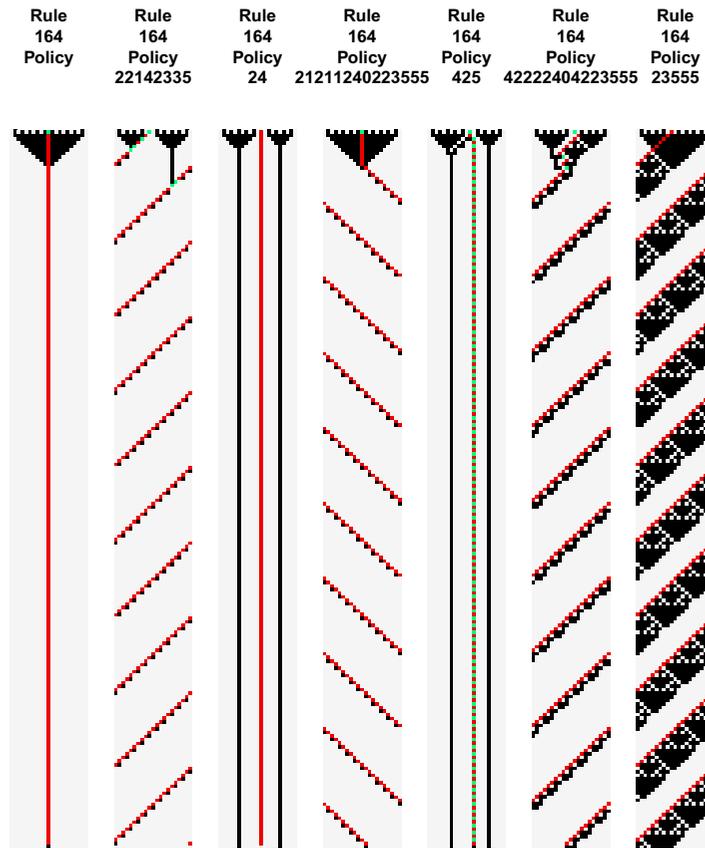


Figure 3. Same as Figure 2, with rule 164 and other policies.

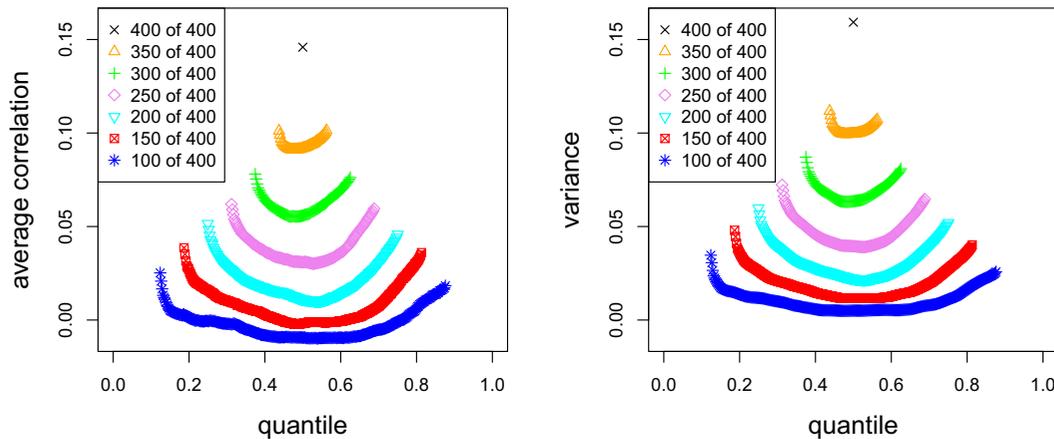


Figure 4. Average correlations for 400 agents (randomly-generated programs) and 256 tasks (all the ECA rules). Left: average correlation per quantiles using several bin sizes where agents are sorted by overall performance. Right: the variance of the bins.

6 CONCLUSION

In this paper we have argued that some questions that have been relevant for human intelligence may become soon important for artificial intelligence as well. One of these questions is the existence of general intelligence and how it can be measured and distinguished from the performance in particular skills. Given the notion of general intelligence as performance in a range of tasks, we have followed the recent theoretical and experimental analysis of the problem in AI (from the no-free-lunch theorems to algorithmic information theory) and fo-

cussed on one particular phenomenon found in human populations, known as Spearman's Law of Diminishing Returns: the positive manifold (the positive correlation of results for a set of cognitive tasks) has been shown to be stronger for less able subpopulations than more able subpopulations.

The choice of this phenomenon responds to its controversy in human intelligence research but also to the counterintuitive character that it would have for artificial intelligence. If SLODR were true in AI we would have that as long as we construct more general-purpose AI systems, we would have that they show less correlation in perfor-

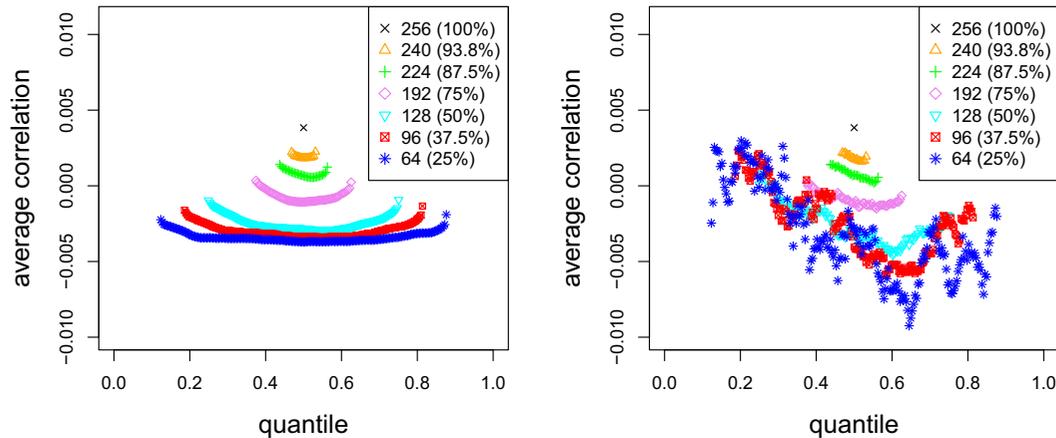


Figure 5. Average correlations for 256 agents (randomly-generated programs) and 256 tasks (all the ECA rules), using mirrored rewards for half of the trials. Left: average correlation per quantiles using several bin sizes, where results are sorted by agent performance. Right: each bin is only evaluated with the tasks of that quantile of difficulty.

mance among a range of tasks thus, in a way, becoming less general.

We have not only challenged that possibility but also analysed whether a reverse universal law of augmenting returns (ULOAR) may appear in a very simple setting of tasks and agents, even with agents of very reduced ability. Closely related to the methodological issues of the analysis of SLODR in human intelligence, we have seen how important it is to perform the analysis in the right way, by using difficulty to have an appropriate level of challenging tasks for each subpopulation to account for the variances. The advantages of this artificial experimentation setting is that we can rule out many other confounding factors that appear in human intelligence, such as the existence of some tasks that have been more common in our evolutionary history or culture, the existence of more efficient specialised modules in our brain predisposed for them, etc.

It is too soon to see whether the current questions and the methodology used here can have any effect in the way general-purpose AI agents will be developed and evaluated in the future, including multi-agent architectures, for which the environment and policies can be extended relatively easily [20]. However, there are some areas in AI that can benefit from some of the issues raised in this paper more immediately. For instance, the ULOAR can suggest new ways of empirically analysing AI systems, to devise new benchmarks and competitions and, most especially, to analyse their results. Of course, in these cases, the tasks and agents would be less minimal (more realistic), but would have more issues about how arbitrarily they have been chosen: many benchmarks include many tasks for which researchers have specialised during decades, and the agents would be a biased subpopulation composed of the participants of the competitions. Another issue for real competitions would be the estimation of difficulty, which is necessary to make the analysis properly. We advocate for principled approaches, based on the policy descriptions, as done here, but other approaches such as Item Response Theory could be used [15, 7, 37].

Some competitions in AI would be better suited than others to the concept of generality. For instance, while we can understand the notion of generality for a planning competition [36] (i.e., a general planner would be the one that is good for a wide range of planning problems), it is for general-purpose agents where the notion of a general factor is more intuitive and closer to the original notions in human intelligence. For instance, the reinforcement learning competi-

tion [54, 10] or the general video game competition [43, 42] would have similar interpretations of results as those discussed here. Nevertheless, the use of correlation matrices for whatever AI competition may show some general factors appearing. We could also investigate whether they grow stronger or not, as the discipline advances.

Finally, an area that can be particularly suitable for this kind of analysis is machine learning. There are already several ‘experiment databases’ [51, 52] whose results can be used to analyse correlations, positive manifolds and whether SLODR (or ULOAR) is taking place there. The interpretation of the results will likely be intriguing but the scope and implications may be fascinating.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their comments. This work has been partially supported by the EU (FEDER), by Generalitat Valenciana PROMETEOII/2015/013 and Spanish MINECO grant TIN2015-69175-C4-1-R.

REFERENCES

- [1] D. Andre and S.J. Russell, ‘State abstraction for programmable reinforcement learning agents’, in *Proceedings of the National Conference on Artificial Intelligence*, pp. 119–125, (2002).
- [2] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, ‘The arcade learning environment: An evaluation platform for general agents’, *Journal of Artificial Intelligence Research*, **47**, 253–279, (06 2013).
- [3] C. Boutilier, R. Reiter, M. Soutchanski, S. Thrun, et al., ‘Decision-theoretic, high-level agent programming in the situation calculus’, in *Proceedings of the National Conference on Artificial Intelligence*, pp. 355–362, (2000).
- [4] S. Bringsjord, ‘Psychometric artificial intelligence’, *Journal of Experimental & Theoretical Artificial Intelligence*, **23**(3), 271–277, (2011).
- [5] S. Bringsjord and B. Schimanski, ‘What is artificial intelligence? Psychometric AI as an answer’, in *International Joint Conference on Artificial Intelligence*, pp. 887–893, (2003).
- [6] A. Cangelosi and M. Schlesinger, *Developmental robotics: From babies to robots*, MIT Press, 2015.
- [7] Rafael Jaime De Ayala, *Theory and practice of item response theory*, Guilford Publications, 2009.
- [8] I. J. Deary, V. Egan, G. J. Gibson, E. J. Austin, C. R. Brand, and T. Kellaghan, ‘Intelligence and the differentiation hypothesis’, *Intelligence*, **23**(2), 105–132, (1996).

- [9] D. K. Detterman and M. H. Daniel, 'Correlations of mental tests with each other and with cognitive variables are highest for low IQ groups', *Intelligence*, **13**(4), 349–359, (1989).
- [10] C. Dimitrakakis, G. Li, and N. Tziortziotis, 'The reinforcement learning competition 2014', *AI Magazine*, **35**(3), 61–65, (2014).
- [11] D. L. Dowe and A. R. Hajek, 'A computational extension to the Turing Test', in *Proceedings of the 4th Conference of the Australasian Cognitive Science Society, University of Newcastle, NSW, Australia, also as Technical Report #97/322, Dept Computer Science, Monash University, Australia*, (1997).
- [12] D. L. Dowe and J. Hernández-Orallo, 'IQ tests are not for machines, yet', *Intelligence*, **40**(2), 77–81, (2012).
- [13] D. L. Dowe, J. Hernández-Orallo, and P. K. Das, 'Compression and intelligence: social environments and communication', in *Artificial General Intelligence*, eds., J. Schmidhuber, K.R. Thórisson, and M. Looks, volume 6830, pp. 204–211. LNAI series, Springer, (2011).
- [14] B. Edmonds, 'The social embedding of intelligence', in *Parsing the Turing Test*, eds., Robert Epstein, Gary Roberts, and Grace Beber, 211–235, Springer, (2009).
- [15] S. E. Embretson and S. P. Reise, *Item response theory for psychologists*, L. Erlbaum, 2000.
- [16] G. J. Fogarty and L. Stankov, 'Challenging the "law of diminishing returns"', *Intelligence*, **21**(2), 157–174, (1995).
- [17] M. Genesereth, N. Love, and B. Pell, 'General game playing: Overview of the AAAI competition', *AI Magazine*, **26**(2), 62, (2005).
- [18] M. Genesereth and M. Thielscher, 'General game playing', *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **8**(2), 1–229, (2014).
- [19] J. Hernández-Orallo, 'Beyond the Turing Test', *J. Logic, Language & Information*, **9**(4), 447–466, (2000).
- [20] J. Hernández-Orallo, 'On environment difficulty and discriminating power', *Autonomous Agents and Multi-Agent Systems*, 1–53, (2014).
- [21] J. Hernández-Orallo, 'C-tests revisited: Back and forth with complexity', in *Artificial General Intelligence - 8th International Conference, AGI 2015, Berlin, Germany, July 22-25, 2015, Proceedings*, eds., J. Bieger, B. Goertzel, and A. Potapov, pp. 272–282. Springer, (2015).
- [22] J. Hernández-Orallo, 'Stochastic tasks: Difficulty and Levin search', in *Artificial General Intelligence - 8th International Conference, AGI 2015, Berlin, Germany, July 22-25, 2015, Proceedings*, eds., J. Bieger, B. Goertzel, and A. Potapov, pp. 90–100. Springer, (2015).
- [23] J. Hernández-Orallo, 'Evaluation in artificial intelligence: From task-oriented to ability-oriented measurement', *Artificial Intelligence Reviews*, (2016).
- [24] J. Hernández-Orallo, *The Measure of All Minds: Evaluating Natural and Artificial Intelligence*, Cambridge University Press, 2016.
- [25] J. Hernández-Orallo and D. L. Dowe, 'Measuring universal intelligence: Towards an anytime intelligence test', *Artificial Intelligence*, **174**(18), 1508 – 1539, (2010).
- [26] J. Hernández-Orallo, D. L. Dowe, and M. V. Hernández-Lloreda, 'Universal psychometrics: Measuring cognitive abilities in the machine kingdom', *Cognitive Systems Research*, **27**, 5074, (2014).
- [27] J. Hernández-Orallo, F. Martínez-Plumed, U. Schmid, M. Siebers, and D. L. Dowe, 'Computer models solving human intelligence test problems: progress and implications', *Artificial Intelligence*, (2016).
- [28] M. Hutter, *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*, Springer, 2005.
- [29] C. Igel and M. Toussaint, 'A no-free-lunch theorem for non-uniform distributions of target functions', *Journal of Mathematical Modelling and Algorithms*, **3**(4), 313–322, (2005).
- [30] J. Insa-Cabrera, D. L. Dowe, and J. Hernández-Orallo, 'Evaluating a reinforcement learning algorithm with a general intelligence test', in *Current Topics in Artificial Intelligence. CAEPIA 2011*, eds., J.A. Lozano, J.A. Gamez, and J.A. Moreno. LNAI Series 7023, Springer, (2011).
- [31] A. R. Jensen, *The g factor: The science of mental ability*, Westport, Praeger, 1998.
- [32] Arthur R Jensen, 'Regularities in Spearman's law of diminishing returns', *Intelligence*, **31**(2), 95–105, (2003).
- [33] S. Legg and M. Hutter, 'Universal intelligence: A definition of machine intelligence', *Minds and Machines*, **17**(4), 391–444, (2007).
- [34] J. Leike and M. Hutter, 'Bad universal priors and notions of optimality', in *Conference on Learning Theory, CoLT*, (2015).
- [35] M. Leonetti and L. Iocchi, 'Improving the performance of complex agent plans through reinforcement learning', in *Proceedings of the 2010 International Conference on Autonomous Agents and Multiagent Sys-*
- tems: volume 1*, pp. 723–730, (2010).
- [36] D. Long and M. Fox, 'The 3rd international planning competition: Results and analysis', *J. Artif. Intell. Res. (JAIR)*, **20**, 1–59, (2003).
- [37] Fernando Martínez-Plumed, Ricardo B. C. Prudêncio, Adolfo Martínez-Usó, and José Hernández-Orallo, 'Making sense of item response theory in machine learning', in *ECAI*, IOS Press, (2016).
- [38] J. McDermott, D. R. White, S. Luke, L. Manzoni, M. Castelli, L. Vaneschi, W. Jaśkowski, K. Krawiec, R. Harper, K. De Jong, and U.-M. O'Reilly, 'Genetic programming needs better benchmarks', in *Proceedings of the 14th international conference on genetic and evolutionary computation conference*, pp. 791–798. ACM, (2012).
- [39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, 'Human-level control through deep reinforcement learning', *Nature*, **518**(26 February), 529–533, (2015).
- [40] A. L. Murray, H. Dixon, and W. Johnson, 'Spearman's law of diminishing returns: A statistical artifact?', *Intelligence*, **41**(5), 439–451, (2013).
- [41] L. Orseau, 'Asymptotic non-learnability of universal agents with computable horizon functions', *Theoretical Computer Science*, **473**, 149–156, (2013).
- [42] D. Perez, S. Samothrakis, J. Togelius, T. Schaul, S. Lucas, A. Couëtoux, J.I Lee, C.-U. Lim, and T. Thompson, 'The 2014 general video game playing competition', *IEEE Transactions on Computational Intelligence and AI in Games*, (2015).
- [43] T. Schaul, 'An extensible description language for video games', *Computational Intelligence and AI in Games, IEEE Transactions on*, **6**(4), 325–331, (2014).
- [44] R. J. Solomonoff. A preliminary report on a general theory of inductive inference, 1960. Report V-131, Zator Co., Cambridge, Ma. Feb 4, revision, Nov.
- [45] R. J. Solomonoff, 'A formal theory of inductive inference. Part I', *Information and control*, **7**(1), 1–22, (1964).
- [46] C. Spearman, 'General Intelligence, Objectively Determined and Measured', *The American Journal of Psychology*, **15**(2), 201–92, (1904).
- [47] C. Spearman, *The abilities of man: Their nature and measurement*, Macmillan, New York, 1927.
- [48] R. J. Sternberg, *Handbook of intelligence*, Cambridge University Press, 2000.
- [49] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, MIT press, 1998.
- [50] E. M. Tucker-Drob, 'Differentiation of cognitive abilities across the life span', *Developmental psychology*, **45**(4), 1097, (2009).
- [51] J. Vanschoren, H. Blockeel, B. Pfahringer, and G. Holmes, 'Experiment databases', *Machine Learning*, **87**(2), 127–158, (2012).
- [52] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, 'Openml: networked science in machine learning', *ACM SIGKDD Explorations Newsletter*, **15**(2), 49–60, (2014).
- [53] D. R. White, J. McDermott, M. Castelli, L. Manzoni, B. W. Goldman, G. Kronberger, W. Jaśkowski, U.-M. O'Reilly, and S. Luke, 'Better GP benchmarks: Community survey results and proposals', *Genetic Programming and Evolvable Machines*, **14**, 3–29, (2013).
- [54] S. Whiteson, B. Tanner, and A. White, 'The Reinforcement Learning Competitions', *The AI magazine*, **31**(2), 81–94, (2010).
- [55] S. Wolfram, *A new kind of science*, Wolfram media, Champaign, IL, 2002.
- [56] D. H. Wolpert, 'What the no free lunch theorems really mean; how to improve search algorithms', Technical report, Santa fe Institute Working Paper, (2012).
- [57] D. H. Wolpert and W. G. Macready, 'No free lunch theorems for search', Technical report, SFI-TR-95-02-010 (Santa Fe Institute), (1995).

Efficient Computation of Exact IRV Margins

Michelle Blom and Vanessa Teague and Peter J. Stuckey and Ron Tidhar¹

Abstract. Computing the margin of victory (MOV) in an Instant Runoff Voting (IRV) election is NP-hard. In an IRV election with winning candidate w , the MOV defines the smallest number of cast votes that, if modified, result in the election of a candidate *other than* w . The ability to compute such margins has significant value. Arguments over the correctness of an election outcome usually rely on the size of the electoral margin. Risk-limiting audits use the size of this margin to determine how much post-election auditing is required. We present an efficient branch-and-bound algorithm for computing exact margins that substantially improves on the current best-known approach. Although exponential in the worst case, our algorithm runs efficiently in practice, computing margins in instances that could not be solved by the current state-of-the-art in a reasonable time frame.

1 Introduction

Instant Runoff Voting (IRV) is a system of preferential voting in which voters rank candidates in order of preference. IRV is used for all parliamentary lower house elections in Australia, parliamentary elections in Fiji and Papua New Guinea, presidential elections in Ireland and Bosnia/Herzegovina, and local elections in numerous locations world-wide, including the UK and United States [19]. Given candidates c_1 , c_2 , c_3 , and c_4 , each vote in an IRV election is a (*possibly partial*) ranking of these candidates. A vote with the ranking $[c_1, c_2, c_3]$ expresses a first preference for candidate c_1 , a second preference for c_2 , and a third for c_3 . The tallying of votes proceeds by distributing each vote to its first ranked candidate. The candidate with the smallest number of votes is eliminated, with their votes redistributed to subsequent, less preferred candidates. Elimination proceeds in this fashion, until a single candidate w remains, who is declared the winner. The *margin of victory* (MOV) of the election is the smallest number of cast votes that must be modified (their ranking replaced with an alternate ranking) to ensure that a candidate *other than* w is the last candidate standing and is elected.

Exact computation of IRV electoral margins is NP-hard [22]. It is difficult to compute either the true runner-up of an IRV election, or the margin by which they lost. Disputing an election outcome, or proving that it is correct, generally requires some argument comparing the electoral margin to the precision of the process. For example, risk-limiting audits require knowledge of the MOV to determine how much auditing is required [15]. A close election, in which the MOV is small, requires more auditing than one with a large margin. As more jurisdictions move toward electronic voting or post-election digitisation of votes, software for election analysis, including MOV computation, will become increasingly important.

Automatic recounting of ballots, for example, is triggered in many jurisdictions if the last round margin (the difference between the tallies of the last two remaining candidates, divided by two and rounded

up) of an IRV election falls below a threshold. The 2013 federal election for the Australian seat of Fairfax, in Queensland, for example, had a last round margin of just 4 votes, triggering a recount. The actual margin of victory for an IRV election, however, may be much lower than its last round margin. The 2011 federal election for the Australian seat of Balmain, New South Wales, had a last round margin of 1239 votes, with 2477 votes separating the last two remaining candidates – a Liberal and a Green. The actual margin of victory, however, was at most 388 votes, with 775 votes separating the Greens and Labor in a prior round of elimination. Last round margins will therefore trigger recounts in only a portion of eligible IRV elections.

This paper contributes an efficient algorithm, denoted *margin*, for exact IRV margin computation that substantially improves on the current best-known approach by Magrino et al. [16], denoted MRSW throughout this paper. On a data set of 29 IRV elections held in the United States between 2007 and 2014, MRSW computes margins in several hundred seconds in 26 of the 29 instances, but fails to compute a margin within 72 hours in the remainder. Although exponential in the worst case, our algorithm runs efficiently in practice on all real IRV election instances for which we could obtain data. On all IRV instances in our data set, our algorithm computes exact margins in less than 6 seconds. The *significance* of our improved algorithm is that MOV computation is *now practical* in elections with large numbers of candidates. In the 2007 San Francisco Mayoral election (with 18 candidates) our algorithm computes the MOV is less than 2 seconds while MRSW times out after 72 hours. We compute the MOV in the Minneapolis 2013 Mayoral election (36 candidates) and the Oakland 2014 Mayoral election (17 candidates) in less than 5 seconds, while MRSW times out after 72 hours. In the 2015 Australian New South Wales (NSW) state election (lower house), 93 IRV elections were held to elect a member of parliament in 93 distinct electorates. Our algorithm computes the MOV in each of these elections in less than 0.04 seconds. MRSW requires up to several hundred seconds.

An obvious, but inefficient, algorithm for computing exact IRV margins is to consider every possible order in which candidates could be eliminated, and use a linear program (LP) solver such as CPLEX to compute the exact number of manipulations (vote modifications) necessary to achieve it. A manipulation replaces the ranking of a vote (e.g., $[c_1, c_2, c_3]$) with a different ranking (e.g., $[c_2, c_1]$). The MOV is the smallest number of such modifications required to realise the election of a different candidate. An insight of Magrino et al. [16] is that this LP can be used to compute a *lower bound* on the number of manipulations required to realise an elimination order *ending* in a particular sequence of candidates. The branch-and-bound algorithm of Magrino et al. [16] guides a search of the space of partial orders, using these lower bounds, for a complete elimination order (involving all candidates) requiring fewest vote manipulations to realise.

Our algorithm has the same basic structure as MRSW but introduces a new, easily computed, and often *tighter*, lower bound on the

¹ The University of Melbourne, Australia, email: michelleb@unimelb.edu.au

number of vote manipulations required to realise an elimination order ending in a specific candidate sequence. Computing this lower bound does not require the solving of an LP, and is often higher in value than that computed by MRSW. Combining our new lower bounds with those generated by the LP of MRSW allows us to prune larger portions of the space of possible elimination orders, earlier in search, and disregard many partial orders without the need to solve an LP. This significantly reduces the time required to compute the MOV. Like Magrino et al. [16], we compute margins under the assumption that any manipulation applied to cast votes must leave the number of votes *unchanged*. We further extend the work of Magrino et al. [16] by presenting two variations of our algorithm in which this assumption is not required. This allows us to answer important practical questions. If there were lost votes, could their inclusion have altered the election outcome? If some people voted twice, could it have made a difference to the outcome? The answer to these questions can be obtained by calculating the MOV under the assumption that votes can only be *added* to the election (*addition only*) or removed (*deletion only*). This type of manipulation is known as *voter control*. We consider both settings in this paper.

The problem of computing margins in elections is related to that of *bribery* in the literature (see [11, 10, 12, 4, 5, 13, 6]). In the bribery problem, voters can be bribed in order to change their votes. Much work has analysed the susceptibility of various voting rules (e.g., Condorcet-based or plurality voting) to bribery, and the complexity of manipulating an election with bribery. In the shift bribery problem, for example, each voter has prices (bribes) for which they are willing to shift the position of a candidate in their vote forward by i positions [5]. The bribery problem seeks to find a lowest cost set of bribes such that an alternate candidate (to the original winner) is elected. If each voter will change their vote at a price of 1, this lowest cost is equivalent to the electoral MOV. While the complexity of the bribery problem, and election manipulation in general, has been well-studied, the work presented in this paper differs in that it presents a practical and implementable algorithm for *computing* electoral margins in IRV elections, outperforming the current state-of-the-art.

The key contributions of this paper are as follows. We present: a new, efficient method of computing a lower bound on the number of vote manipulations required to realise the elimination of candidates in an IRV election in a specific order; a modification of the MOV-calculation algorithm of Magrino et al. [16] in which this bounding procedure is used; a comparison of our approach with that of Magrino et al. [16] on 29 IRV elections held in the United States between 2007 and 2014 (25 of which appear in the work of Magrino et al. [16]), and 93 IRV elections from the 2015 NSW state election; and two adaptations of our algorithm for settings in which votes can only be *added* or *removed*, but not both.

This paper is structured as follows. Section 2 examines related work in the complexity and computation of IRV margins. Definitions and concepts underlying our approach are presented in Section 3. Section 4 describes our improved algorithm for the computation of margins, highlighting where it deviates from that of Magrino et al. [16]. Section 5 evaluates our algorithm on a suite of IRV instances. Section 6 examines two variations of our algorithm in which votes may be *deleted* from or *added* to an election profile, but not both.

2 Related Work

Computing the exact MOV in an IRV election is NP-hard [22]. Determining whether a single voter can manipulate an IRV election to achieve a desired outcome is also NP-hard [2]. This result has been

extended by Conitzer et al. [8, 9] to show that for a weighted variant of IRV in which there are more than 3 candidates (and votes are weighted), finding a manipulation for which a specific candidate is elected (constructive manipulation) is NP-complete. For elections of more than 4 candidates, finding a destructive manipulation (ensuring that a specific candidate is not elected) is also NP-complete. The complexity of manipulating plurality and Condorcet-based elections by adding or deleting voters (equivalent to the addition or deletion of votes considered in this paper) is examined by Bartholdi III et al. [3]. The complexity of strategic voting in schemes for which votes can be *partial* rankings over candidates is investigated by Narodytska and Walsh [17]. See Rothe and Schend [20] for a review of such complexity results. In some of these works, IRV is referred to as a form of Single Transferable Vote (STV) with a single winner.

In this paper we seek to determine the smallest number of votes cast in an IRV election that, if modified, will result in a different outcome (a different winner). Similar questions have been considered for alternate voting rules. The complexity of manipulating an election with bribery is considered by Faliszewski et al. [12], under a number of voting schemes: Condorcet-based; approval voting; scoring rules; veto rules; and plurality. Their aim is to find a manipulation to achieve a desired election result, while minimising the cost of bribes given to voters for changing their vote.

An algorithm for computing upper and lower bounds on the IRV MOV has been developed by Cary [7]. The “Winner Elimination” upper bound, for example, finds the most efficient way to eliminate the winner in each round, returning the least-cost (involving fewest vote changes) of these. Bounds on the MOV for IRV and other voting schemes have also been provided by Sarwate et al. [21]. A lower bound is computed by picking sets of candidates to eliminate in order to maximise the difference between the number of votes allocated to the candidates in these sets, and to the remaining candidate with the fewest votes. The bounds defined by Cary [7] and Sarwate et al. [21] can be computed in polynomial time, but are not necessarily tight (i.e., they may differ significantly from the true margin).

In 31 IRV elections conducted in the United States and Ireland, Sarwate et al. [21] compare their computed bounds to known exact margins. Lower bounds equaled exact margins in 18 elections, and in the others fell below exact margins by 0.6% to 19% of the total votes cast. Computed upper bounds were typically within a few votes of exact margins, with a number of exceptions. For the 2009 Aspen City Council election, the lower and upper bound of Sarwate et al. [21] differ from the exact margin by 2.5% (62 votes) and 9.9% (254 votes) of the total number of votes cast. Our algorithm finds the exact MOV in this election within 1.5 seconds. In the 2008 race for Pierce County assessor, their lower and upper bound differ from the exact margin by 0.6% (1945 votes) and 1.6% (5079 votes) of the total number of votes. Our algorithm computes the MOV within 0.02 seconds.

Magrino et al. [16] present a branch-and-bound algorithm for computing exact IRV margins. Applied to 25 IRV elections in the United States, this approach successfully computes exact margins in all but one instance. This algorithm, referred to as MRSW in this paper, considers the space of possible alternate elimination orders of a set of candidates \mathcal{C} , in which the actual winner $c_w \in \mathcal{C}$ is *not* the last remaining candidate. Given one such order, a linear program (LP) computes the smallest number of votes (of those cast) that must be modified in order to realise this elimination order. When applied to a partial sequence of candidates, π' , this LP computes the smallest number of vote changes required to achieve this order of elimination in a *reduced election profile*, in which all candidates not in π' have been eliminated (and their votes redistributed). It is clear that this

Initially, all candidates remain standing (are not eliminated)
While there is *more than one* candidate standing
 For every candidate c standing
 Tally (count) the votes in which c is the highest-ranked
 candidate of those standing
 Eliminate the candidate with the smallest tally
The winner is the one candidate not eliminated

Figure 1. An informal definition of the IRV counting algorithm.

number is a lower bound on the number of vote changes required to achieve any *complete* elimination order (involving all candidates) ending in π' . The MRSW algorithm builds a tree of partial elimination orders, with the smallest LP evaluation obtained for each visited leaf (a complete order) providing an upper bound on the electoral margin. Partial orders whose associated lower bound is *larger* than the best known upper bound are pruned from the search.

The main restricting cost of MRSW is the number of nodes that are explored and evaluated via the LP. Our algorithm dramatically reduces the number of partial elimination orders (nodes) explored, relative to MRSW, through the use of a bounding rule assigning tighter (higher) lower bounds to nodes close to the root of the tree. We are thus able to prune larger portions of the search space, earlier.

3 Preliminaries

The tallying of votes in an IRV election proceeds by a series of rounds in which the candidate with the lowest number of votes is eliminated (see Figure 1) with the last remaining candidate declared the winner. All votes in an eliminated candidate's tally are distributed to the next most-preferred (remaining) candidate in their ranking.

Let \mathcal{C} be the set of candidates in an IRV election \mathcal{B} . We refer to sequences of candidates π in list notation (e.g., $\pi = [c_1, c_2, c_3, c_4]$), and use such sequences to represent both votes and elimination orders. We will often treat a sequence as the set of elements it contains. An election \mathcal{B} is defined as a multiset² of votes, each vote $b \in \mathcal{B}$ a sequence of candidates in \mathcal{C} , with no duplicates, listed in order of preference (most preferred to least preferred). Let $first(\pi)$ denote the first candidate appearing in sequence π (e.g., $first([c_2, c_3]) = c_2$). In each round of vote counting, there are a current set of eliminated candidates \mathcal{E} and a current set of candidates still standing $\mathcal{S} = \mathcal{C} \setminus \mathcal{E}$. The winner c_w of the election is the last standing candidate.

Definition 1 Projection $p_{\mathcal{S}}(\pi)$ We define the projection of a sequence π onto a set \mathcal{S} as the largest subsequence of π that contains only elements of \mathcal{S} . (The elements keep their relative order in π).

For example: $p_{\{c_2, c_3\}}([c_1, c_2, c_4, c_3]) = [c_2, c_3]$ and $p_{\{c_2, c_3, c_4, c_5\}}([c_6, c_4, c_7, c_2, c_1]) = [c_4, c_2]$.

Each candidate $c \in \mathcal{C}$ has a *tally* of votes. Votes are added to this tally upon the elimination of a candidate $c' \in \mathcal{C} \setminus c$, and are redistributed from this tally upon the elimination of c .

Definition 2 Tally $t_{\mathcal{S}}(c)$ Given candidates $\mathcal{S} \subseteq \mathcal{C}$ are still standing in an election \mathcal{B} , the tally for a candidate $c \in \mathcal{C}$, denoted $t_{\mathcal{S}}(c)$,

² A multiset allows for the inclusion of duplicate items.

Ranking	Count
$[c_2, c_3]$	4
$[c_1]$	20
$[c_3, c_4]$	9
$[c_2, c_3, c_4]$	6
$[c_4, c_1, c_2]$	15
$[c_1, c_3]$	6

(a)

Candidate	Rnd1	Rnd2	Rnd3
c_1	26	26	26
c_2	10	10	—
c_3	9	—	—
c_4	15	24	30

(b)

Table 1. An example IRV election profile, stating (a) the number of votes cast with each listed ranking over candidates c_1, c_2, c_3 , and c_4 , and (b) the tallies after each round of vote counting and elimination.

is defined as the number of votes $b \in \mathcal{B}$ for which c is the most-preferred candidate of those remaining. Recall that $p_{\mathcal{S}}(b)$ denotes the sequence of candidates mentioned in b that are also in \mathcal{S} .

$$t_{\mathcal{S}}(c) = |\{b \mid b \in \mathcal{B}, c = first(p_{\mathcal{S}}(b))\}| \quad (1)$$

Definition 3 Margin of Victory (MOV) The MOV in an election with candidates \mathcal{C} and winner $c_w \in \mathcal{C}$, is the *smallest* number of votes whose ranking must be modified (by an adversary) so that a candidate $c' \in \mathcal{C} \setminus c_w$ is elected.

If several candidates receive the same number of votes, at any stage of the IRV count, we assume that the adversary can decide which of the candidates is eliminated. This assumption is made by Magrino et al. [16]. If this is not the case, the MOV of Definition 3 slightly underestimates (but never overestimates) the true margin.

Definition 4 Last Round Margin ($LRM_{\mathcal{B}}$) The last round margin of election \mathcal{B} , in which two candidates $\mathcal{S} = \{c, c'\}$ remain with $t_{\mathcal{S}}(c)$ and $t_{\mathcal{S}}(c')$ votes in their tallies, is equal to half the difference between the tallies of c and c' rounded up.

$$LRM_{\mathcal{B}} = \lceil \frac{|t_{\mathcal{S}}(c) - t_{\mathcal{S}}(c')|}{2} \rceil \quad (2)$$

Example 1 Consider the IRV election of Table 1. The tallies of candidates c_1, c_2, c_3 , and c_4 , in the 1st counting round are 26, 10, 9, and 15 votes. Candidate c_3 is eliminated, and 9 votes are distributed to c_4 , who now has a tally of 24. Candidate c_2 , on 10 votes, is eliminated next with 6 of their votes distributed to c_4 (the remainder have no subsequent preferences and are exhausted). Candidates c_1 and c_4 remain with tallies of 26 and 30. The last round margin is 2 votes. Candidate c_1 is eliminated from consideration and c_4 elected.

4 A Fast Algorithm for Calculating Margins

We present a branch-and-bound algorithm for computing the MOV in IRV elections. This algorithm has the same basic structure as that of MRSW [16], being a traversal of the tree of possible orders of candidate elimination. Our algorithm incorporates a substantially improved pruning rule allowing us to dramatically reduce the portion of this tree we must traverse to compute the MOV. In this section, we describe our algorithm in detail and contrast its performance against MRSW on 29 IRV elections held in the United States between 2007 and 2014, and the 2015 NSW lower house election held in Australia. In the latter election, 93 IRV elections were held to elect a member of parliament in 93 electorates.

Magrino et al. [16] define an LP, DISTANCETO, shown below, for computing the minimum number of votes cast in an election \mathcal{B} that, if

manipulated, realises a specific complete elimination order π (involving all candidates \mathcal{C}). When applied to a partial order π' ($\pi' \subset \mathcal{C}$), DISTANCETO computes a *lower bound* on the number of votes that, if manipulated, will realise an elimination order *ending* in π' . In this paper, we define a *bounding rule* that, when applied to a partial order π' , computes an alternative, often tighter (higher), lower bound.

Let \mathbf{R} denote the set of possible (partial and total) rankings R of candidates \mathcal{C} that could appear on a vote, N_R the number of votes cast in the election with ranking $R \in \mathbf{R}$, and N the total number of votes cast. For each ranking $R \in \mathbf{R}$, we define variables:

- q_R number of votes to be changed into R ;
- m_R number of votes with ranking R in the unmodified election to be changed into something other than R ; and
- y_R number of votes in the modified election with ranking R .

Given a partial or complete order π , the DISTANCETO LP is:

$$\min \sum_{R \in \mathbf{R}} q_R$$

$$N_R + q_R - m_R = y_R \quad \forall R \in \mathbf{R} \quad (3)$$

$$\sum_{R \in \mathbf{R}} q_R = \sum_{R \in \mathbf{R}} m_R \quad (4)$$

$$\sum_{R \in \mathcal{R}_{i,i}} y_R \leq \sum_{R \in \mathcal{R}_{j,i}} y_R \quad \forall c_i, c_j \in \pi \cdot i < j \quad (5)$$

$$n \geq y_R \geq 0, \quad N_R \geq m_R \geq 0, \quad q_R \geq 0 \quad \forall R \in \mathbf{R} \quad (6)$$

Constraint (3) states that the number of votes with ranking $R \in \mathbf{R}$ in the new election is equal to the sum of those with this ranking in the unmodified election and those whose ranking has *changed to* R , minus the number of votes whose ranking has been *changed from* R . Constraint (5) defines a set of *special elimination constraints* which force the candidates in π to be eliminated in the stated order. $\mathcal{R}_{j,i}$ denotes the subset of rankings in \mathbf{R} ($\mathcal{R}_{j,i} \subset \mathbf{R}$) in which c_j is the most preferred candidate still standing (i.e., that will count toward c_j 's tally) at the end of round i (in which candidate c_i is eliminated). Constraint (4) ensures that the total number of votes cast in the election does not change as a result of the manipulation.

4.1 Two New Lower-Bounding Rules

Let us consider a partial elimination order $\pi' \subset \mathcal{C}$. Each candidate $e \in \mathcal{C} \setminus \pi'$ must be eliminated before every candidate $c \in \pi'$ (recall that we are computing a lower bound on the number of votes that must be manipulated to realise an elimination order *ending* in π'). We define $\Delta(c, e)$ as the number of votes $b \in \mathcal{B}$ for which c is ranked higher than e , or c appears and e does not. This is equal to the number of votes with rankings $[c, e]$ or $[c]$ when all candidates apart from c and e are removed. At any time e is eliminated before c , c has a tally of *at most* $\Delta(c, e)$ votes at the moment e is eliminated, with all other votes assigned to e , or another candidate. Recall that $p_{\mathcal{S}}(b)$ denotes the *projection* of b onto set \mathcal{S} (i.e., the ranking of vote b with all candidates not in the set \mathcal{S} removed).

$$\Delta(c, e) = |\{b \mid b \in \mathcal{B}, p_{\{c,e\}}(b) \in \{[c, e], [c]\}\}| \quad (7)$$

The *primary vote* of candidate $c \in \mathcal{C}$, denoted $f(c)$, is the number of votes $b \in \mathcal{B}$ for which c is ranked highest.

$$f(c) = |\{b \mid b \in \mathcal{B}, c = \text{first}(b)\}| \quad (8)$$

To ensure that candidate e is eliminated *before* candidate c , we require that $f(e) \leq \Delta(c, e)$. In other words, we require that the primary vote of e is less than or equal to the number of votes in which c is ranked higher than e , or c appears and e does not. If it is the case that $f(e) > \Delta(c, e)$, we need to change the relative counts by the amount $f(e) - \Delta(c, e)$ for this order of elimination to be feasible. Let $l_1(c, e)$ denote a lower bound on the number of votes that must be modified to achieve the elimination of e before c .

$$l_1(c, e) = \max(0, \lceil \frac{f(e) - \Delta(c, e)}{2} \rceil) \quad (9)$$

Example 2 Consider the partial elimination order $\pi' = [c_2]$ in the election of Table 1. To realise an elimination order ending in c_2 , all other candidates must be eliminated prior to c_2 's election. To ensure that c_1 appears before c_2 in the elimination sequence, we count all votes that could possibly be in c_2 's tally at the point at which c_1 is eliminated – this is denoted $\Delta(c_2, c_1)$ and defined in Equation 7. In this example, $\Delta(c_2, c_1) = 10$. The smallest number of votes that c_1 could have in their tally upon elimination is their initial tally (or primary vote) $f(c_1)$, defined in Equation 8. Here, $f(c_1) = 26$. For c_1 to appear before c_2 in the elimination sequence, we must change the votes so that, at the very least, the minimum number of votes that c_1 could have (upon elimination) is less than the maximum number of votes c_2 could have. Equation 8 computes this ‘minimal number’ of required vote changes, $l_1(c_2, c_1)$. Here, $l_1(c_2, c_1) = 8$.

Since each candidate $e \in \mathcal{C} \setminus \pi'$ has been eliminated prior to each $c \in \pi'$, we can compute a lower bound on the number of votes that must be modified to realise an elimination order ending in π' , $b_1(\pi')$, as shown in Equation 10. In contrast to the DISTANCETO LP, our lower bound does not consider the order in which candidates are eliminated in π' , but computes a lower bound on the number of votes we must alter to ensure that the candidates in π' are the last candidates standing. The DISTANCETO LP, however, operates on a reduced election profile in which all candidates not in π' have been eliminated, and their votes redistributed. It computes the manipulation required to realise π' in this setting.

$$b_1(\pi') = \max\{l_1(c, e) \mid c \in \pi', e \in \mathcal{C} \setminus \pi'\} \quad (10)$$

Example 3 (Example 2 cont.) For the partial order $\pi' = [c_2]$, we compute lower bounds on the smallest number of vote changes required to eliminate each candidate c_i ($i \neq 2$) prior to c_2 's election, $l_1(c_2, c_i)$. The largest $l_1(c_2, c_i)$ becomes our lower bound, $b_1(\pi')$, on the number of votes we must change to realise an elimination order ending in $[c_2]$. In this example, $b_1(\pi') = 8$. DISTANCETO would assign a lower bound of 0 to π' as in a reduced election involving only c_2 , no votes need be changed to ensure they are elected.

The bound b_1 can be tightened. Consider the partial elimination order π' , for which all candidates $e \in \mathcal{C} \setminus \pi'$ are eliminated before all $c \in \pi'$. We know that e has at least $f(e)$ votes in its tally. Candidate c may not have, in their tally, all votes which have been counted toward $\Delta(c, e)$ (those in which c appears before e , or c appears, but e does not). Some of these votes may lie in the tallies of other candidates in π' , who have not yet been eliminated. We define $\Delta_{\mathcal{S}}(c, e)$ as the maximum number of votes that c can have in their tally at the time e is eliminated, where $\mathcal{S} = \{e\} \cup \pi'$ denotes the minimal set of candidates that must be ‘still standing’ at this time.

$$\Delta_{\mathcal{S}}(c, e) = |\{b \mid b \in \mathcal{B}, c = \text{first}(p_{\mathcal{S}}(b))\}| \quad (11)$$

To realise a situation in which candidate $e \in \mathcal{C} \setminus \pi'$ is eliminated prior to candidate $c \in \pi'$, we require that $f(e) \leq \Delta_S(c, e)$. If $f(e) > \Delta_S(c, e)$ then we must modify at least $l_2(c, e, \pi')$ votes.

$$l_2(c, e, \pi') = \max(0, \lceil \frac{f(e) - \Delta_S(c, e)}{2} \rceil) \quad (12)$$

Equation 13 defines a tighter lower bound on the number of votes in \mathcal{B} that must be changed to ensure that $\pi' \subset \mathcal{C}$ are the last remaining candidates, $b_2(\pi')$. Note that $l_2(c, e, \pi') \geq l_1(c, e)$, for all $\pi' \subset \mathcal{C}$, $c \in \pi'$, and $e \in \mathcal{C} \setminus \pi'$. Hence, $b_2(\pi') \geq b_1(\pi')$ for all $\pi' \subset \mathcal{C}$. Both $b_1(\pi')$ and $b_2(\pi')$ are independent of the order of candidates in π' .

$$b_2(\pi') = \max\{l_2(c, e, \pi') \mid c \in \pi', e \in \mathcal{C} \setminus \pi'\} \quad (13)$$

Example 4 Consider the partial elimination order $\pi' = [c_3, c_1]$. Our first lower bound on the number of manipulations required to realise an elimination order ending in π' , $b_1(\pi')$, equals 0. To compute this we evaluate $l_1(c_i, c_j)$ for $i = 1, 3$, and $j = 2, 4$, and take the maximum result. These values represent the smallest number of manipulations required to eliminate c_2 and c_4 before candidates c_3 and c_1 . To compute our second lower bound, $b_2(\pi')$, we evaluate $l_2(c_i, c_j, \pi')$ for $i = 1, 3$, and $j = 2, 4$ (via Equation 12), and take the maximum result. We find that $l_2(c_i, c_j, \pi') = 0$ for all i and j with the exception of $l_2(c_3, c_2, \pi')$ which equals 1. Hence, $b_2(\pi') = 1$. DISTANCETO assigns π' a lower bound of 0.

4.2 A Branch-and-Bound Algorithm: margin

Figure 2 outlines our algorithm, denoted `margin`, for computing exact margins in IRV elections. This algorithm shares the basic structure as MRSW [16], both being branch-and-bound algorithms.

An initially empty priority queue Q of partial elimination orders is maintained throughout the algorithm (step 1). An upper bound on the number of votes that must be modified to realise a winning candidate other than c_w is initialised to the last round margin of the election, $LRM_{\mathcal{B}}$ (step 2). A partial order $\pi' = \{c\}$ for all $c \in \mathcal{C} \setminus c_w$ (i.e., we do not consider orders that will end in the winning candidate) is inserted into Q with a score given by $b_2(\pi')$ of Equation 13 iff that score is lower than the current upper bound (steps 5–7). If this score is larger or equal to the upper bound, π' and all its descendants are pruned (will not be explored). MRSW adds these orders to Q with a score of 0 (as in a reduced election involving only one candidate c , no votes need to be altered to ensure that c wins).

We repeatedly select the partial order π' in Q with the smallest score (i.e., the smallest lower bound on the size of the manipulation required to realise an elimination order ending in π'). This order is removed from Q (step 10), and is expanded. It is at this point that we evaluate π' with the DISTANCETO LP (step 13). We have found, from experimentation, that DISTANCETO can provide a higher bound than b_2 , albeit infrequently. If the revised bound from DISTANCETO is larger or equal to the current upper bound, π' is pruned from the tree in step 14–15 (i.e., it is not added to Q). Otherwise, we consider each candidate $c \in \mathcal{C} \setminus \pi'$ and create new order π in which c is eliminated just prior to the first candidate in π' (step 17). If π is a complete order, containing all candidates, we compute the exact number of vote changes required to realise π with DISTANCETO. If this number is lower than the current upper bound, U is replaced with the smaller number (step 19). If π is a partial order, we compute $b_2(\pi)$ as defined in Equation 13 (step 20). If this lower bound is lower than the current upper bound, π is inserted into Q (steps 21–22), otherwise it and its descendants are pruned from the search.

```

margin( $\mathcal{C}, \mathcal{B}, c_w$ )
1   $Q := \emptyset$ 
2   $U := LRM_{\mathcal{B}}$ 
3  for( $c \in \mathcal{C} \setminus \{c_w\}$ )
4     $\pi' := [c]$ 
5     $l := b_2(\pi')$ 
6    if( $l < U$ )
7       $Q := Q \cup \{(l, \pi')\}$ 
8  while  $Q \neq \emptyset$ 
9    ( $l, \pi'$ ) := arg min  $Q$ 
10    $Q := Q \setminus \{(l, \pi')\}$ 
11    $U := \text{expand}(l, \pi', U, Q, \mathcal{C}, \mathcal{B})$ 
12  return  $U$ 

expand( $l, \pi', U, Q, \mathcal{C}, \mathcal{B}$ )
13   $l' := \max\{l, \text{DISTANCETO}(\pi', \mathcal{C}, \mathcal{B})\}$ 
14  if( $l' \geq U$ )
15    return  $U$ 
16  for( $c \in \mathcal{C} \setminus \pi'$ )
17     $\pi := [c] ++ \pi'$ 
18  if( $|\pi| = |\mathcal{C}|$ )
19    return  $\min\{U, \text{DISTANCETO}(\pi, \mathcal{C}, \mathcal{B})\}$ 
20   $l'' = \max\{l', b_2(\pi)\}$ 
21  if( $l'' < U$ )
22     $Q := Q \cup \{(l'', \pi)\}$ 
23  return  $U$ 

```

Figure 2. MOV computation for an IRV election \mathcal{B} with candidates \mathcal{C} and winner $c_w \in \mathcal{C}$; $\boxed{\#}$ denotes where our algorithm differs from MRSW [16].

The b_2 bound is not guaranteed to generate a tighter bound than DISTANCETO (although in practice we find that it *is* tighter in a majority of instances). In using our lower bounding rules to *select* partial orders for expansion, and evaluating DISTANCETO only on these selected orders, we reduce the number of LPs solved by our algorithm. MRSW evaluates the DISTANCETO LP for each child formed upon the expansion of a node.

When all elimination orders have been examined, or pruned, `margin` terminates (step 12), returning U , which now equals the smallest number of vote changes required to alter the outcome of election \mathcal{B} .

4.3 Comparing MRSW and margin: An Example

Consider the IRV election of Table 1. Figure 3a records the partial elimination orders considered by MRSW when computing the MOV. Each node denotes an elimination order that is traversed and evaluated by MRSW, with its score recorded. MRSW first considers the partial orders $[c_2]$, $[c_3]$, and $[c_4]$, assigning each a score of 0. The upper bound on the manipulation size required to change the election outcome is set to 2 votes (the last round margin). MRSW considers the children of node $[c_3] - [c_2, c_3]$ with a score of 5, $[c_1, c_3]$ with a score of 11, and $[c_4, c_3]$ with a score of 0. These scores are obtained by solving DISTANCETO. Nodes $[c_2, c_3]$ and $[c_1, c_3]$ can be pruned as their scores are higher than the current upper bound. Node $[c_4, c_3]$ is expanded, creating children $[c_2, c_4, c_3]$ with a score of 0 and $[c_1, c_4, c_3]$ with a score of 6 (consequently pruned). The leaf node $[c_1, c_2, c_4, c_3]$ is then visited and assigned a score of 11 (also pruned). MRSW continues to expand nodes in this manner as shown

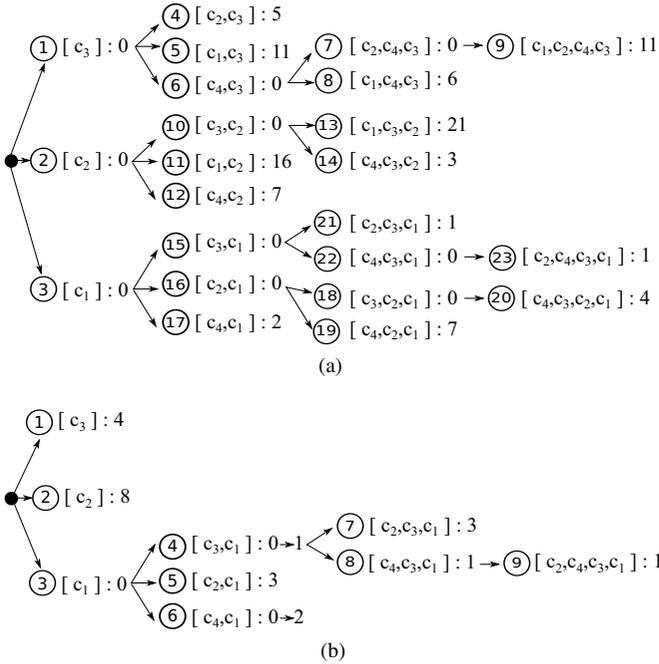


Figure 3. Traversal of elimination orders by (a) MRSW, and (b) margin, recording the sequence in which orders are evaluated (circled), and the score given to each order, for the election of Table 1.

in Figure 3a, visiting 23 nodes and solving 20 LPs (the children of the root node have a value of 0 as in a reduced election with one candidate, no votes need to be changed to ensure they are elected).

Our algorithm visits and evaluates the nodes shown in Figure 3b, reporting beside each node the score we assign to it. Nodes $[c_3]$, $[c_2]$, and $[c_1]$, are assigned scores of 4, 8, and 0, respectively. Nodes $[c_3]$ and $[c_2]$ are immediately pruned as their scores are larger than the current upper bound of 2. This allows us to concentrate on elimination orders ending in c_1 . Nodes $[c_3, c_1]$, $[c_2, c_1]$, and $[c_4, c_1]$, are assigned scores of 0, 3, and 0 (when $[c_3, c_1]$ and $[c_4, c_1]$ are selected for expansion, DISTANCETO is solved for these nodes and their scores revised to 1 and 2, respectively). Node $[c_3, c_1]$ is expanded, visiting nodes $[c_2, c_3, c_1]$ and $[c_4, c_3, c_1]$ with scores of 3 and 1. The leaf $[c_2, c_4, c_3, c_1]$ is given a score of 1 by solving DISTANCETO. The upper bound is updated to 1. DISTANCETO is then solved for node $[c_4, c_1]$, assigned a score of 0 by our bounding rules, obtaining a revised lower bound of 2. As this is greater than the current upper bound, $[c_4, c_1]$ can be pruned immediately. Our algorithm assigns scores to 9 nodes, but solves only 4 LPs in the process – DISTANCETO is solved at a node only when it has been selected for expansion, or if it is a leaf node. Scores assigned by our lower bounding rules are used to determine which nodes to select for expansion.

5 Computational Results

We have evaluated our improved algorithm (Figure 2) on 29 IRV elections held in the United States, and 93 IRV elections involved in the 2015 NSW state election (lower house). We contrast the performance of our approach on this data set with that of MRSW. Execution was performed on a machine with four 2.10 GHz CPUs, 7.7 GB of memory, and with a 72 hour timeout. CPLEX 12.5.1 was used to solve all LPs. Table 2 reports, for each election considered, the number of candidates and votes cast, the number of calls to DISTANCETO

made by MRSW and by our algorithm (denoted margin), the computation time (in milliseconds) of the two algorithms, the MOV and the last-round margin. Of the 93 IRV elections in the 2015 NSW state election, 4 reported a MOV that differed from its last round margin.

Our algorithm substantially reduces both the number of calls to DISTANCETO and computation time. For example, we are able to compute the MOV of the 2007 San Francisco Mayoral election, where MRSW timed out after 72 hours. In generating these results, our algorithm uses the tighter b_2 pruning rule of Equation 13. In the majority of instances considered, pruning with b_2 was either faster, or as fast, as pruning with b_1 . The b_2 rule is more costly to compute, however, than b_1 . In the 2007 San Francisco Mayoral election, for example, 1300 ms are used to compute the margin when pruning with b_2 (solving 94 LPs). In contrast, 1139 ms are used when pruning with b_1 , even though 970 LPs are solved in the process. For the Aspen 2009 City Council race, however, 1241 ms are used when pruning with b_1 and 1039 ms when pruning with b_2 . For the Pierce 2008 County Assessor instance, 17 ms and 9 ms are used when pruning with b_1 and b_2 , respectively. The full table of results comparing the performance of b_1 and b_2 has been omitted for brevity.

In the 93 IRV elections of the 2015 NSW state election, MRSW computed the MOV in 14 to 354,007 ms, solving 21 to 23,768 LPs. Our margin algorithm computed margins in 1 to 35 ms, solving 1 to 13 LPs. Table 2 reports the results of 16 of these IRV elections – in 4 of which (Ballina, Maitland, Lismore, and Willoughby) the MOV differs from the last round margin. The number of candidates in each election range from 5 to 8. MRSW, in general, requires more time to compute margins in elections with more candidates.

We have additionally applied margin to all instances of the PreLib data set³ that can be interpreted as an election (261 instances). The number of candidates and votes cast in these instances range from 3 to 2819, and 4 to 15,101, respectively. Margin computation in all instances with more than 500 candidates is trivial, with only 4 votes cast. With a 30 minute time limit, margin computes margins in all but 10 instances, while MRSW fails in 30 instances. In the 10 instances for which margin fails to compute a margin in 30 minutes, it finds lower and upper bounds on the margin that differ by up to 9 votes in 7/10 instances and by 22 to 789 votes in the remainder.

6 Variations: Voter Control

Suppose some votes are lost during an election. We extend margin to determine the minimum number of votes that must be added to change an election outcome as follows. We first remove the division by two when calculating the last round margin (Definition 4), and l_1 , b_1 , l_2 , and b_2 of Equations 9–13 (in this setting, manipulations can only add votes). We then modify the DISTANCETO LP to calculate the minimum number of vote additions required to enforce a certain elimination order. To do so, we interpret variable q_R as the number of votes with ranking $R \in \mathbf{R}$ added to the election. We set $m_R = 0$ for $R \in \mathbf{R}$, and remove Constraint (4) which forces the number of votes to remain constant. If the computed MOV is larger than the number of lost votes, then their inclusion could not have altered the election outcome. In the 2013 election of candidates to six seats in Western Australia’s Senate a discrepancy of 1,375 initially verified votes was discovered during a recount (resulting from a lost ballot box) [18]. The election result was overturned, and a repeat election held in 2014. While Single Transferable Vote (STV) is used in Australian Senate elections – a more complex scheme than IRV – this case demonstrates the impact of such mistakes when they occur.

³ <http://www.preflib.org/>

$ C $	# Votes Cast	MRSW LPs	margin LPs	MRSW Time (ms)	margin Time (ms)	MOV	LRM	Election Name
2	45,986	1	0	1	1	15,356	15,356	Berkeley 2010 Auditor
2	15,243	1	0	1	1	4,830	4,830	Oakland 2010 D2 School Board
2	14,040	1	0	1	1	4,826	4,826	Oakland 2010 D6 School Board
2	23,494	1	0	1	1	8,338	8,338	San Leandro 2010 D3 City Council
3	122,268	6	1	1	1	17,081	17,081	Oakland 2010 Auditor
3	15,243	6	1	1	1	2,175	2,175	Oakland 2010 D2 City Council
3	23,494	6	1	1	1	742	742	San Leandro 2010 D5 City Council
4	4,862	22	1	4	1	364	364	Berkeley 2010 D7 City Council
4	5,333	23	2	4	1	878	878	Berkeley 2010 D8 City Council
4	14,040	24	2	4	1	2,603	2,603	Oakland 2010 D6 City Council
4	43,661	19	1	3	1	2,007	2,007	Pierce 2008 City Council
4	159,987	19	1	3	1	8,396	8,396	Pierce 2008 County Auditor
5	312,771	49	4	9	1	2,027	2,027	Pierce 2008 County Executive
5	2,544	65	1	12	1	89	89	Aspen 2009 Mayor
5	6,426	85	1	17	1	1,174	1,174	Berkeley 2010 D1 City Council
5	5,708	64	1	12	1	517	517	Berkeley 2010 D4 City Council
5	13,482	49	1	9	1	486	486	Oakland 2012 D5 City Council
5	28,703	65	2	13	1	2,332	2,332	San Leandro 2012 D4 City Council
7	23,494	292	1	81	1	116	116	San Leandro 2010 Mayor
7	312,771	312	19	98	9	1,111	3,650	Pierce 2008 County Assessor
7	26,761	351	19	111	8	386	684	Oakland 2012 D3 City Council
8	23,884	4,989	2	3,905	2	2,329	2,329	Oakland 2010 D4 City Council
8	57,492	7,737	2	6,772	2	8,522	8,522	Berkeley 2012 Mayor
8	34,180	1,301	2	666	2	423	423	Oakland 2012 D1 City Council
11	122,268	26,195	4	90,988	18	1,013	1,013	Oakland 2010 Mayor
11	2,544	15,109	224	64,705	1,039	35	162	Aspen 2009 City Council
17	101,431	—	234	timeout	5,067	10,201	10,201	Oakland 2014 Mayor
18	149,465	—	94	timeout	1,300	50,837	50,837	San Francisco 2007 Mayor
36	79,415	—	2	timeout	1,173	6,949	6,949	Minneapolis 2013 Mayor
5	44797	130	1	37	2	8235	8235	Seat of Lakemba, NSW 2015 lower house
5	45467	1,071	1	2,326	2	8,495	8495	Seat of Liverpool, NSW 2015 lower house
5	47348	130	1	36	2	10,806	10,806	Seat of Manly, NSW 2015 lower house
6	47,933	222	13	161	12	4,012	5,446	Seat of Maitland, NSW 2015 lower house
6	47,208	173	8	107	9	209	1,173	Seat of Lismore, NSW 2015 lower house
6	47,370	655	12	502	11	10,160	10,247	Seat of Willoughby, NSW 2015 lower house
7	47,865	380	11	652	35	1,130	1,267	Seat of Ballina, NSW 2015 lower house
7	48,358	867	1	1,693	9	3,132	3,132	Seat of Newcastle, NSW 2015 lower house
7	45497	710	1	1,323	9	3,536	3,536	Seat of Newtown, NSW 2015 lower house
7	48065	1071	1	2,326	10	4253	4253	Seat of Lake Macquarie, NSW 2015 lower house
8	47,590	7,091	1	79,637	22	4,069	4,069	Seat of Clarence, NSW 2015 lower house
8	47,803	21,054	1	190,066	18	7,311	7,311	Seat of Hawkesbury, NSW 2015 lower house
8	48,571	9,923	1	100,143	21	4,974	4,974	Seat of Swansea, NSW 2015 lower house
8	46,756	23,768	2	354,007	34	8,574	8,574	Seat of Murray, NSW 2015 lower house
8	48,002	4,106	1	29,211	20	2,576	2,576	Seat of Penrith, NSW 2015 lower house
8	42,892	2,369	1	11,243	18	2,864	2,864	Seat of Sydney, NSW 2015 lower house

Table 2. Running times and margins computed for 29 IRV elections in the United States, and 16/93 IRV elections held in the 2015 NSW state election (lower house), using MRSW and margin. Cases where the margin of victory (MOV) differs from the last round margin (LRM) are in bold

In Australian state and federal elections, each polling station has a book containing the names and addresses of all voters in the region. As each voter casts their vote, their name is struck off by hand. This does not prevent a voter from voting more than once at multiple polling stations. In the 2013 Australian federal election, the Australian Electoral Commission (AEC) ‘investigated almost 19,000 instances of multiple voting’ [1]. In this situation we know the number of invalid votes, but not *which* votes are invalid. If this total exceeds the minimum number of votes that, if removed, change the result of the election, we know these invalid votes may have influenced the outcome. To determine this number, we extend `margin` as follows. The division by two in our definition of last round margin, and bounding rules, is removed. Variable m_R becomes the number of

cast votes with ranking $R \in \mathbf{R}$ that we will delete. We set $q_R = 0$ for $R \in \mathbf{R}$, and remove Constraint (4). We replace q_R with m_R in the `DISTANCETO` objective, as we seek to minimise the number of deleted votes required to realise an alternate outcome.

Our `margin` algorithm, when applied to our suite of IRV instances, is able to find the MOV in both the *addition-* and *deletion-only* settings, with runtimes similar to those in Table 2. Table 3 reports, for each election in Table 2, the number of candidates and votes cast, the number of calls to `DISTANCETO` made by MRSW and `margin`, in the *addition-only* setting, together with the computation time (in milliseconds) of the two algorithms, the MOV and the last-round margin.

$ C $	# Votes Cast	MRSW LPs	margin LPs	MRSW Time (ms)	margin Time (ms)	MOV	<i>LRM</i>	Election Name
2	45,986	1	0	1	1	30,711	30,711	Berkeley 2010 Auditor
2	15,243	1	0	1	1	9,660	9,660	Oakland 2010 D2 School Board
2	14,040	1	0	1	1	9,651	9,651	Oakland 2010 D6 School Board
2	23,494	1	0	1	1	16,675	16,675	San Leandro 2010 D3 City Council
3	122,268	6	1	1	1	34,162	34,162	Oakland 2010 Auditor
3	15,243	6	1	1	1	4,349	4,349	Oakland 2010 D2 City Council
3	23,494	6	1	1	1	1,484	1,484	San Leandro 2010 D5 City Council
4	4,862	22	1	2	1	728	728	Berkeley 2010 D7 City Council
4	5,333	24	2	2	1	1,756	1,756	Berkeley 2010 D8 City Council
4	14,040	24	2	2	1	5,205	5,205	Oakland 2010 D6 City Council
4	43,661	19	1	2	1	4,014	4,014	Pierce 2008 City Council
4	159,987	19	1	2	2	16,792	16,792	Pierce 2008 County Auditor
5	312,771	49	4	4	1	4,054	4,054	Pierce 2008 County Executive
5	2,544	65	1	6	1	177	177	Aspen 2009 Mayor
5	6,426	79	1	7	1	2,348	2,348	Berkeley 2010 D1 City Council
5	5,708	62	1	5	1	1,033	1,033	Berkeley 2010 D4 City Council
5	13,482	49	1	4	1	972	972	Oakland 2012 D5 City Council
5	28,703	62	2	6	1	4,664	4,664	San Leandro 2012 D4 City Council
7	23,494	292	1	35	1	232	232	San Leandro 2010 Mayor
7	312,771	312	19	53	7	2,221	7,299	Pierce 2008 County Assessor
7	26,761	351	19	70	6	771	1367	Oakland 2012 D3 City Council
8	23,884	3,801	2	1,714	2	4,657	4,657	Oakland 2010 D4 City Council
8	57,492	5,693	2	2,465	2	17,044	17,044	Berkeley 2012 Mayor
8	34,180	1,186	2	315	2	845	845	Oakland 2012 D1 City Council
11	122,268	23,541	4	45,285	21	2,025	2,025	Oakland 2010 Mayor
11	2,544	13,943	220	50,117	862	70	323	Aspen 2009 City Council
17	101,431	—	224	timeout	4,812	20,402	20,402	Oakland 2014 Mayor
18	149,465	—	94	timeout	1,273	101,674	101,674	San Francisco 2007 Mayor
36	79,415	—	2	timeout	1,176	13,898	13,898	Minneapolis 2013 Mayor
5	44797	123	1	36	1	16,470	16,470	Seat of Lakemba, NSW 2015 lower house
5	45467	121	1	35	1	16,989	16,989	Seat of Liverpool, NSW 2015 lower house
5	47348	116	1	30	1	21,612	21,612	Seat of Manly, NSW 2015 lower house
6	47,933	213	13	194	15	8,023	10,892	Seat of Maitland, NSW 2015 lower house
6	47,208	173	8	134	11	417	2345	Seat of Lismore, NSW 2015 lower house
6	47,370	503	12	449	14	20319	20493	Seat of Willoughby, NSW 2015 lower house
7	47,865	385	11	945	50	2,259	2,534	Seat of Ballina, NSW 2015 lower house
7	48,358	819	1	2,182	8	6,264	6,264	Seat of Newcastle, NSW 2015 lower house
7	45,497	632	1	1,624	14	7,072	7,072	Seat of Newtown, NSW 2015 lower house
7	48,065	1,005	1	3,031	14	8,506	8,506	Seat of Lake Macquarie, NSW 2015 lower house
8	47,590	4,663	1	50,983	33	8,137	8,137	Seat of Clarence, NSW 2015 lower house
8	47,803	12,963	1	130,655	27	14,621	14,621	Seat of Hawkesbury, NSW 2015 lower house
8	48,571	6,679	1	62,190	31	9,948	9,948	Seat of Swansea, NSW 2015 lower house
8	46,756	11,624	2	156,357	38	17,147	17,147	Seat of Murray, NSW 2015 lower house
8	48,002	3,389	1	25,774	33	5,151	5,151	Seat of Penrith, NSW 2015 lower house
8	42,892	2,304	1	14,119	26	5,727	5,727	Seat of Sydney, NSW 2015 lower house

Table 3. Running times and margins computed for each of the IRV elections of Table 2, using MRSW and *margin*, under the restriction that votes can only be *added* (not deleted or modified). Cases where the margin of victory (MOV) differs from the last round margin (LRM) are in bold.

7 Concluding Remarks

We have presented an algorithm, denoted *margin*, for computing IRV margins that significantly outperforms the current state-of-the-art. Our algorithm can efficiently compute the MOV in all IRV instances for which we could obtain data. This includes a number of instances for which the current state-of-the-art approach could not compute the margin, in a reasonable time frame of 72 hours. The significance of this work is that automated margin computation is now practical for IRV elections with a large number of candidates. We have presented two easily computed lower bounds on the degree of manipulation required to realise an elimination order ending in a specific sequence. This allows us to prune large portions of the space of

possible alternate elimination orders when computing IRV margins. Moreover, we have described how our *margin* algorithm can be used to determine whether lost votes, or invalid votes (e.g., from electors voting multiple times), could have influenced an election outcome.

IRV has several extensions, including various forms of the Single Transferable Vote (STV). STV is used to elect candidates to the Australian Senate, in all elections in Malta, and in most elections in the Republic of Ireland [14]. The extension of our algorithm for computing margins in IRV elections to STV elections, where candidates are elected to multiple seats, and the votes of elected candidates are redistributed at a fractional value, is a topic of future research. Preliminary results suggest that our *margin* algorithm can be adapted to apply to STV, using a non-linear version of *DISTANCETO*.

REFERENCES

- [1] ABC. Thousands admit to multiple votes in 2013 federal election. www.abc.net.au/news/2014-02-26/thousands-admit-to-multiple-votes-in-2013-federal-election/5284230, 2014. Accessed: August 2015.
- [2] J. J. Bartholdi III and J. B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.
- [3] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8):27–40, 1992.
- [4] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. Prices matter for the parameterized complexity of shift bribery. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1398–1404, 2014.
- [5] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. Large-scale election campaigns: combinatorial shift bribery. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 67–75. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
- [6] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. Complexity of shift bribery in committee elections. *arXiv preprint arXiv:1601.01492*, 2016.
- [7] D. Cary. Estimating the margin of victory for instant-runoff voting. In *USENIX Accurate Electronic Voting Technology Workshop: Workshop on Trustworthy Elections*, USENIX Association Berkeley, CA, USA, 2011.
- [8] V. Conitzer, J. Lang, and T. Sandholm. How many candidates are needed to make elections hard to manipulate? In *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 201–214, Bloomington, Indiana, USA, 2003. ACM.
- [9] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):14, 2007.
- [10] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Algorithmic Game Theory*, pages 299–310. Springer, 2009.
- [11] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. The complexity of bribery in elections. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, volume 6, pages 641–646, 2006.
- [12] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How Hard Is Bribery in Elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2011.
- [13] P. Faliszewski, Y. Reisch, J. Rothe, and L. Schend. Complexity of manipulation, bribery, and campaign management in Bucklin and fallback voting. *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 29(6):1091–1124, 2015.
- [14] D. Farrell and I. McAllister. Australia: The Alternative Vote in a Compliant Political Culture. In M. Gallagher and P. Mitchell, editors, *The Politics of Electoral Systems*, pages 79–97. Oxford University Press, Oxford, 2005.
- [15] M. Lindeman and P.B. Stark. A gentle introduction to risk-limiting audits. *IEEE Security and Privacy*, 10:42–49, 2012.
- [16] T. R. Magrino, R. L. Rivest, E. Shen, and D. A. Wagner. Computing the margin of victory in IRV elections. In *USENIX Accurate Electronic Voting Technology Workshop: Workshop on Trustworthy Elections*, USENIX Association Berkeley, CA, USA, 2011.
- [17] N. Narodytska and T. Walsh. The computational impact of partial votes on strategic voting. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 657–662, 2014.
- [18] Parliament of Australia. The disputed 2013 WA Senate election. www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/FlagPost/2013/November/The_disputed_2013_WA_Senate_election, 2013. Accessed: April 2016.
- [19] R. Richie. Instant Runoff Voting: What Mexico (and Others) Could Learn. *Election Law Journal*, 3:501–512, 2004.
- [20] J. Rothe and L. Schend. Challenges to complexity shields that are supposed to protect elections against manipulation and control: a survey. *Annals of Mathematics and Artificial Intelligence*, 68(1-3):161–193, 2013. ISSN 1012-2443.
- [21] A. Sarwate, S. Checkoway, and H. Shacham. Risk-limiting audits and the margin of victory in nonplurality elections. *Statistics, Politics, and Policy*, 4(1):29–64, January 2013.
- [22] L. Xia. Computing the margin of victory for various voting rules. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 982–999, 2012.

On Labelling Statements in Multi-Labelling Argumentation

Pietro Baroni¹ and Guido Governatori² and Régis Riveret³

Abstract. In computational models of argumentation, argument justification has attracted more attention than statement justification, and significant sensitivity losses are identifiable when dealing with the justification of statements by otherwise appealing formalisms. This paper reappraises statement justification as a formalism-independent component in argument-based reasoning. We introduce a novel general model of argument-based reasoning based on multiple stages of labellings, the last one being devoted to statement justification, identify two alternative paths from argument acceptance to statement justification, and compare their expressiveness. We then show that this model encompasses several prominent literature proposals as special cases, thereby enabling a systematic comparison of existing approaches to statement justification, evidencing their merits and limits. Finally we illustrate our model by specifying a generic ignorance-aware statement justification and showing how it can be seamlessly integrated into different formalisms.

1 INTRODUCTION

In studies of argument-based reasoning, argument justification has received far more attention than statement justification, often treated as a simple byproduct of the former. As a consequence, significant expressiveness and sensitivity problems can be identified in the treatment of statement justification by otherwise appealing formalisms. In particular, in a recent paper [4] we have pointed out that even in very simple common sense reasoning examples the statement justification outcomes produced by different argumentation formalisms may be significantly different and show counterintuitive aspects. To overcome these limitations, in [4] we have proposed a preliminary approach where statement justification is regarded as a formalism-independent tunable component of argument-based reasoning. The approach is based on a generic multi-labelling system and its application and relevant advantages have been exemplified in the case of *ASPIC*⁺ [11]. In this paper we provide a twofold advancement in this research direction. First we identify two alternative design choices in multi-labelling systems and compare their expressiveness. Second we illustrate the application of multi-labelling systems to model a representative set of argumentation formalisms, thus confirming their potential to provide a common formalism-independent reference for the investigation of statement justification principles and methods. The presentation is supported by a common sense example, illustrated below.

Example 1. Suppose that Dr. Smith says to you: “Given your clinical picture, you are affected by disease D1, not disease D2”. Suppose then

Dr. Jones, considered equally competent, says to you: “Given your clinical picture, you are affected by disease D2, not by disease D1”. Your view on the justification of the statements S1=“I am affected by disease D1” and S2=“I am affected by disease D2” may become quite uncertain. In a different situation, at home, you use an off-the-shelf test kit suggesting you have caught disease D3. You then undertake a serious and reliable clinical test, which excludes disease D3. Would you consider the same status for the statement S3=“I am affected by disease D3” and the statements S1, S2? What about the justification status of the statement S4=“I am affected by D4”, where D4 is a poorly studied and initially asymptomatic disease you never heard of? Intuitively, different justification statuses seem reasonable and useful. Actually, such distinctions may be decisive. Surprisingly, as will be discussed in the following, the current versions of several well-known structured argumentation formalisms fail to distinguish these justification statuses, equating, for instance, the justification status of S4 with the one of S3, or with that of S1 and S2, or even the justification status of S3 with that of S1 and S2.

We advocate that the loss of sensitivity to statements’ statuses is not intrinsic to the formalisms, but is rather due to the relatively limited attention paid to justification of statements, often treated as a sort of appendix of the notions of acceptance and justification of arguments. To address this limitation, we propose a multi-labelling model of the argumentation process, showing that starting from the common basis of argument production and acceptance two alternative approaches to statement justification can be considered.

The paper is organised as follows. In Section 2 we propose multi-labelling systems for argumentation, catering for an argument-focused approach and a statement-focused approach. In Section 3 we compare the expressiveness of the two approaches. We show in Section 4 that several literature proposals can be seen as instances of our model, and in Section 5 how it can support tunable statement justification labellings, before concluding in Section 6.

2 MULTI-LABELLING SYSTEMS

To investigate the different notions of justification involved in a generic argument-based reasoning system, one may define a generic multi-labelling model of the argumentation process. In [4] we preliminarily introduced a model consisting of four stages (or levels), namely argument production, argument acceptance, argument justification, statement justification. Here we extend our analysis with two variants of the model, called the *argument-focused* approach and the *statement-focused* approach. These two approaches differ at the third stage (see Fig. 1). In the *argument-focused* approach, argument acceptance gives rise to argument justification at a third stage, from which statement justification is derived at a fourth stage.

¹ DII, Univ. of Brescia, email: pietro.baroni@unibs.it

² Data61, CSIRO, email: guido.governatori@data61.csiro.au

³ Data61, CSIRO, email: regis.riveret@data61.csiro.au

In the *statement-focused* approach, argument acceptance is projected on statements, giving rise to statement acceptance at a third stage, from which statement justification is again derived at a fourth stage. The description and formal definitions of these different stages are provided in the sequel, preceded by some basic concepts. Due to space limits we cannot include illustrative examples in this section: they are provided with the analyses developed in Sections 3 and 4.

Basic concepts. Multi-labelling systems are based on the notion of labelling.

Definition 1 (Labelling). *Given a set of labels Λ and a set T , a Λ -labelling L of T is a (possibly partial) function $L : T \rightarrow \Lambda$.*

The idea is then that argument-based reasoning can be regarded as a sequence of labelling activities where the starting point consists in producing labellings for arguments and the final result is a labelling for the statements about which the arguments are built. Moving across the stages, the labellings produced at one stage are used as input to produce new labellings at the next stage, where the labels, their meaning and/or the labelled elements change. As we will see, it may happen that sets of labels are projected or transferred from the elements of a stage to those in the subsequent one (e.g. a statement may “receive” the set of labels associated to all the arguments concluding it). In these cases a synthesis (S) operator is required.

Definition 2 (S-operators). *Given two sets of labels Λ_1 and Λ_2 , a S-operator from Λ_1 to Λ_2 is a function $\mathfrak{S} : Pow(\Lambda_1) \rightarrow \Lambda_2$, where $Pow(T)$ denotes the powerset of T . A double S-operator from Λ_1 to Λ_2 is a function $\mathfrak{S} : Pow(\Lambda_1) \times Pow(\Lambda_1) \rightarrow \Lambda_2$.*

Intuitively, a synthesis operator associates with each set of labels in Λ_1 a single corresponding label in Λ_2 . Such an operator is naturally applied to projected or transferred sets of labels belonging to Λ_1 to obtain a synthetic representation in the context of Λ_2 . Double synthesis operators will have a role when contraries come into play.

Argument production. The first stage regards the production of a set of arguments \mathcal{A} whose structure and mutual relationships are left unspecified. The only relevant property for our purposes is that each argument $A \in \mathcal{A}$ has a conclusion, denoted as $Con(A)$, belonging to a language \mathcal{L} . We do not make any assumption on the set of arguments, while we assume that the language is equipped with a contrariness relation. In its simplest form the contrariness relation corresponds to the traditional notion of negation but other more general forms of contrariness have been considered in the literature [10, 3]. To encompass this wider view, we assume a contrariness relation Cnt , allowing the existence of multiple (or no) contraries for each statement, and hence being compatible with a variety of argumentation formalisms.

Definition 3 (Language). *A language \mathcal{L} is a set of statements equipped with a contrariness relation $Cnt : \mathcal{L} \rightarrow Pow(\mathcal{L})$. For all $\varphi \in \mathcal{L}$, $\psi \in Cnt(\varphi)$ is called a contrary of φ .*

The outcomes of the argument production stage can be summarised in an abstract form as an argument-conclusion structure.

Definition 4 (Argument-conclusion structure). *An argument-conclusion structure (ACS) is a triple $\langle \mathcal{L}, \mathcal{A}, Con \rangle$ where \mathcal{L} is a language, \mathcal{A} is a finite set of arguments and $Con : \mathcal{A} \rightarrow \mathcal{L}$ is a relation associating every argument with its conclusion.*

Note that some elements of \mathcal{L} may not play the role of conclusions, e.g. if \mathcal{L} encompasses negation as failure.

In general each statement $\varphi \in \mathcal{L}$ is supported by a (possibly empty) set of arguments denoted as $Arg(\varphi)$. This notion can obviously be extended to sets of statements as in the following definition.

Definition 5 (Supporting arguments). *Given an ACS $\langle \mathcal{L}, \mathcal{A}, Con \rangle$ and a set $\Phi \subseteq \mathcal{L}$, the set of supporting arguments of Φ is defined as*

$$Arg(\Phi) \triangleq \{A \in \mathcal{A} \mid Con(A) \in \Phi\}.$$

Argument acceptance. The second stage concerns the acceptance evaluation of a set of arguments, the outcome is a set of argument acceptance labellings using a set of labels Λ_{AA} . Each label in Λ_{AA} represents an individual argument acceptance status and a labelling L_{AA} altogether represents a “reasonable” viewpoint (in general among many possible ones) about the acceptance of the arguments in \mathcal{A} .

Definition 6 (Argument acceptance labelling and evaluation). *Given an ACS $\mathcal{AC} = \langle \mathcal{L}, \mathcal{A}, Con \rangle$ and a set of acceptance labels Λ_{AA} , an argument acceptance Λ_{AA} -labelling for \mathcal{AC} is a Λ_{AA} -labelling of \mathcal{A} .*

A Λ_{AA} -acceptance evaluation for \mathcal{AC} is a set of argument acceptance Λ_{AA} -labellings for \mathcal{AC} denoted as $\mathfrak{L}_{AA}(\mathcal{AC})$ or just \mathfrak{L}_{AA} where not ambiguous.

Different ways of using the set \mathfrak{L}_{AA} give rise to two alternatives for the subsequent stages. In a nutshell, in the *argument-focused* approach, the set of acceptance labelling is projected on arguments and then synthesised, giving rise to an argument justification stage, while in the *statement-focused* approach, the focus is transferred from arguments to their conclusions, giving rise to a statement acceptance stage.

Argument-focused (AF) approach

Argument justification. In the AF approach, the third stage deals with the definition of an argument justification labelling L_{AJ} using a set Λ_{AJ} of argument justification labels.

It is natural to assume that L_{AJ} is functionally dependent on \mathfrak{L}_{AA} and, in particular, we make two basic assumptions on the nature of this dependency. First, for each argument A , $L_{AJ}(A)$ depends only on the acceptance labels of A in \mathfrak{L}_{AA} ; second, cardinality does not count in this evaluation, i.e. for each label $\lambda \in \Lambda_{AA}$ it only matters whether there is any $L_{AA} \in \mathfrak{L}_{AA}$ such that $L_{AA}(A) = \lambda$. Following these assumptions, L_{AJ} is obtained by first projecting \mathfrak{L}_{AA} on arguments and then applying a synthesis operator from Λ_{AA} to Λ_{AJ} .

Definition 7 (Argument acceptance projection). *Let $\mathcal{AC} = \langle \mathcal{L}, \mathcal{A}, Con \rangle$ be an ACS and \mathfrak{L}_{AA} a Λ_{AA} -acceptance evaluation for \mathcal{AC} . For every $A \in \mathcal{A}$ the projection of \mathfrak{L}_{AA} on A is defined as*

$$\Sigma_{AA}(A) \triangleq \{\lambda \in \Lambda_{AA} \mid \exists L_{AA} \in \mathfrak{L}_{AA} : L_{AA}(A) = \lambda\}.$$

Definition 8 (Argument justification labelling and evaluation). *Given a set of justification labels Λ_{AJ} and an ACS $\mathcal{AC} = \langle \mathcal{L}, \mathcal{A}, Con \rangle$, an argument justification Λ_{AJ} -labelling for \mathcal{AC} is a Λ_{AJ} -labelling of \mathcal{A} . Given a Λ_{AA} -acceptance evaluation \mathfrak{L}_{AA} for \mathcal{AC} , an argument justification Λ_{AJ} -labelling L_{AJ} is the synthesis of the projection of \mathfrak{L}_{AA} based on a S-operator \mathfrak{S}_{AJ} from Λ_{AA} to Λ_{AJ} if for every argument $A \in \mathcal{A}$ it holds that $L_{AJ}(A) = \mathfrak{S}_{AJ}(\Sigma_{AA}(A))$.*

AF statement justification. The fourth stage in the AF approach caters for the justification status of statements, i.e. the elements of the language \mathcal{L} . We assume that this is functionally dependent on a given argument justification labelling, as defined above, and is represented by exactly one statement justification labelling L_{ASJ} using a set of

statement justification labels Λ_{SJ} . We assume that the contraries of a statement may play a role in the assesment of its justification and of course that L_{ASJ} depends on the outcome of the third stage: for each statement φ the justification labels are transferred from arguments to φ itself and to its contraries, yielding $\Upsilon_{AJ}(\varphi)$ and $\Upsilon_{\overline{AJ}}(\varphi)$ according to the following definition.

Definition 9 (Justification transfer). *Let $\mathcal{AC} = \langle \mathcal{L}, \mathcal{A}, Con \rangle$ be an ACS and L_{AJ} an argument justification Λ_{AJ} -labelling for \mathcal{AC} . For every statement $\varphi \in \mathcal{L}$ the supporting transfer and contrary-supporting transfer of L_{AJ} on φ are respectively defined as*

$$\Upsilon_{AJ}(\varphi) \triangleq \{\lambda \in \Lambda_{AJ} \mid \exists A \in Arg(\{\varphi\}) : L_{AJ}(A) = \lambda\}$$

$$\Upsilon_{\overline{AJ}}(\varphi) \triangleq \{\lambda \in \Lambda_{AJ} \mid \exists A \in Arg(Con(\varphi)) : L_{AJ}(A) = \lambda\}.$$

Based on $\Upsilon_{AJ}(\varphi)$ and $\Upsilon_{\overline{AJ}}(\varphi)$, assigning a justification label in Λ_{SJ} to each statement amounts to define a double S -operator $\mathfrak{S}_{ASJ} : Pow(\Lambda_{AJ}) \times Pow(\Lambda_{AJ}) \rightarrow \Lambda_{SJ}$.

Definition 10 (AF statement justification labelling and evaluation). *Given an ACS $\mathcal{AC} = \langle \mathcal{L}, \mathcal{A}, Con \rangle$ and a set of statement justification labels Λ_{SJ} , an AF statement justification Λ_{SJ} -labelling for \mathcal{AC} is a Λ_{SJ} -labelling of \mathcal{L} . Given an argument justification Λ_{AJ} -labelling L_{AJ} for \mathcal{AC} , an AF statement justification Λ_{SJ} -labelling L_{ASJ} is the synthesis of the transfer of L_{AJ} based on a double S -operator \mathfrak{S}_{ASJ} from Λ_{AJ} to Λ_{SJ} if for every statement $\varphi \in \mathcal{L}$ $L_{ASJ}(\varphi) = \mathfrak{S}_{ASJ}(\Upsilon_{AJ}(\varphi), \Upsilon_{\overline{AJ}}(\varphi))$.*

Statement-focused (SF) approach

Statement acceptance. Since multiple arguments may have the same conclusion, within each single labelling $L_{AA} \in \mathfrak{L}_{AA}(\mathcal{AC})$ one can transfer the acceptance labels from arguments to statements and then synthesise them to obtain a statement acceptance labelling. In this way, a set of statement acceptance labellings can be derived from an argument acceptance evaluation.

Definition 11 (Acceptance transfer). *Let $\mathcal{AC} = \langle \mathcal{L}, \mathcal{A}, Con \rangle$ be an ACS and L_{AA} an argument acceptance Λ_{AA} -labelling for \mathcal{AC} . For every statement $\varphi \in \mathcal{L}$ the supporting transfer and contrary-supporting transfer of L_{AA} on φ are respectively defined as*

$$\Upsilon_{AA}(\varphi) \triangleq \{\lambda \in \Lambda_{AA} \mid \exists A \in Arg(\{\varphi\}) : L_{AA}(A) = \lambda\}$$

$$\Upsilon_{\overline{AA}}(\varphi) \triangleq \{\lambda \in \Lambda_{AA} \mid \exists A \in Arg(Con(\varphi)) : L_{AA}(A) = \lambda\}.$$

Definition 12 (Statement acceptance labelling and evaluation). *Given a set of statement acceptance labels Λ_{SA} and an ACS $\mathcal{AC} = \langle \mathcal{L}, \mathcal{A}, Con \rangle$, a statement acceptance Λ_{SA} -labelling for \mathcal{AC} is a Λ_{SA} -labelling of \mathcal{L} . Given an argument acceptance Λ_{AA} -labelling L_{AA} for \mathcal{AC} , a statement acceptance Λ_{SA} -labelling L_{SA} is the synthesis of the transfer of L_{AA} based on a double S -operator \mathfrak{S}_{SA} from Λ_{AA} to Λ_{SA} if for every $\varphi \in \mathcal{L}$ $L_{SA}(\varphi) = \mathfrak{S}_{SA}(\Upsilon_{AA}(\varphi), \Upsilon_{\overline{AA}}(\varphi))$.*

A statement acceptance evaluation for \mathcal{AC} is a set of statement acceptance labellings for \mathcal{AC} denoted $\mathfrak{L}_{SA}(\mathcal{AC})$ or just \mathfrak{L}_{SA} where not ambiguous.

SF statement justification. In the SF approach, the statement acceptance evaluation is projected on each statement and on its contraries.

Definition 13 (Statement acceptance projection). *Let $\mathcal{AC} = \langle \mathcal{L}, \mathcal{A}, Con \rangle$ be an ACS and \mathfrak{L}_{SA} a statement acceptance evaluation for \mathcal{AC} . For every statement $\varphi \in \mathcal{L}$ the projection of \mathfrak{L}_{SA} on φ and on its contraries are respectively defined as*

$$\Sigma_{SA}(\varphi) \triangleq \{\lambda \in \Lambda_{SA} \mid \exists L_{SA} \in \mathfrak{L}_{SA} : L_{SA}(\varphi) = \lambda\}$$

$$\Sigma_{\overline{SA}}(\varphi) \triangleq \{\lambda \in \Lambda_{SA} \mid \exists \overline{\varphi} \in Con(\varphi), \exists L_{SA} \in \mathfrak{L}_{SA} : L_{SA}(\overline{\varphi}) = \lambda\}.$$

Then, for each statement φ , a statement justification labelling L_{SSJ} is a function of the acceptance labels of φ itself, and its contraries.

Definition 14 (SF statement justification labelling and evaluation). *Given an ACS $\mathcal{AC} = \langle \mathcal{L}, \mathcal{A}, Con \rangle$ and a set of statement justification labels Λ_{SJ} , a SF statement justification Λ_{SJ} -labelling for \mathcal{AC} is a Λ_{SJ} -labelling of \mathcal{L} . Given a statement acceptance evaluation \mathfrak{L}_{SA} for \mathcal{AC} , a SF statement justification Λ_{SJ} -labelling L_{SSJ} is the synthesis of the projection of \mathfrak{L}_{SA} based on a double S -operator \mathfrak{S}_{SSJ} from Λ_{SA} to Λ_{SJ} if for every statement $\varphi \in \mathcal{L}$ $L_{SSJ}(\varphi) = \mathfrak{S}_{SSJ}(\Sigma_{SA}(\varphi), \Sigma_{\overline{SA}}(\varphi))$.*

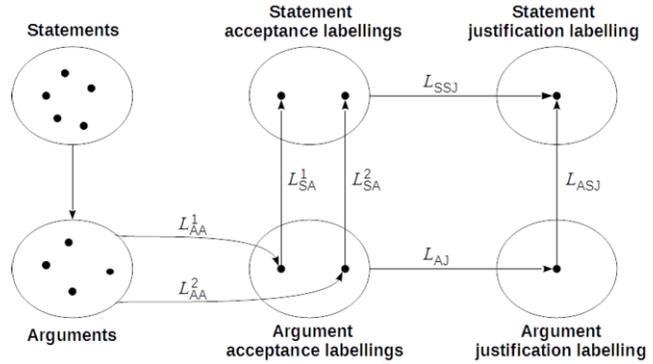


Figure 1. Overview of a multi-labelling system.

A multi-labelling system (MLS) consists of an ACS equipped with the relevant evaluations either in the AF or SF approach. Accordingly, two classes of multi-labelling systems can be identified:

- an AF MLS is a tuple $\mathfrak{L}^A = \langle \mathcal{AC}, \mathfrak{L}_{AA}, L_{AJ}, L_{ASJ} \rangle$;
- an SF MLS is a tuple $\mathfrak{L}^S = \langle \mathcal{AC}, \mathfrak{L}_{AA}, \mathfrak{L}_{SA}, L_{SSJ} \rangle$;

where all the symbols are interpreted as in the previous definitions.

3 COMPARING THE AF AND SF APPROACHES

In the AF approach the idea is that the outcomes of the second stage (i.e. the acceptance labellings of arguments) are first projected and synthesised on the argument themselves, giving rise to argument justification. Then argument justification outcomes are transferred to statements and synthesised in turn, taking contraries into account, to get statement justification. On the other hand in the SF approach the outcomes of the argument acceptance stage are immediately transferred and synthesised on statements, giving rise to statement acceptance. Then the acceptance outcomes of a statement and of its contraries are taken into account to derive statement justification.

One may wonder whether, under the assumptions we made, the two approaches feature the same expressiveness, i.e. whether any statement justification labelling produced by an AF MLS can be obtained by a corresponding SF MLS and vice versa.

We show that the answer is negative, using some simple examples with the set of argument acceptance labels $\Lambda_{AA} = \{IN, OUT\}$.

A distinction expressible only by the SF approach.

C1. Consider a first case C1 where there are (possibly among others) two arguments A and B such that, for some statement φ , $Arg(\{\varphi\}) = \{A, B\}$ (i.e. they have the same conclusion φ and no other arguments conclude φ). For simplicity, let us also assume that $Arg(Cnt(\varphi)) = \emptyset$. Suppose that the outcome of the argument acceptance stage consists of two labellings, i.e. $\mathfrak{L}_{AA}(\mathcal{AC}) = \{L_{AA}^1, L_{AA}^2\}$ such that $L_{AA}^1(A) = \text{IN}$ and $L_{AA}^1(B) = \text{OUT}$, while $L_{AA}^2(A) = \text{OUT}$ and $L_{AA}^2(B) = \text{IN}$.

- In the AF approach, at the argument justification stage, we get $\Sigma_{AA}(A) = \Sigma_{AA}(B) = \{\text{IN}, \text{OUT}\}$ and, whatever S-operator \mathfrak{S}_{AJ} is adopted it must be that $L_{AJ}(A) = L_{AJ}(B) = \mathfrak{S}_{AJ}(\{\text{IN}, \text{OUT}\}) = \lambda$ for some $\lambda \in \Lambda_{AJ}$. At the statement justification stage, we have $\Upsilon_{AJ}(\varphi) = \{\lambda\}$ and $\Upsilon_{\overline{AJ}}(\varphi) = \emptyset$. Then, whatever S-operator \mathfrak{S}_{ASJ} is adopted, we get that $L_{ASJ}(A)$ functionally depends on the pair $(\{\lambda\}, \emptyset)$ i.e. $L_{ASJ}(\varphi) = \mathfrak{S}_{ASJ}(\{\lambda\}, \emptyset)$.
- In the SF approach, at the statement acceptance stage, we get $\Upsilon_{AA}^1(\varphi) = \Upsilon_{AA}^2(\varphi) = \{\text{IN}, \text{OUT}\}$. Therefore, $L_{SA}^1(\varphi) = L_{SA}^2(\varphi) = \mathfrak{S}_{SA}(\{\text{IN}, \text{OUT}\}, \emptyset) = \lambda^0$ for some $\lambda^0 \in \Lambda_{SA}$. At the statement justification stage, $\Sigma_{SA}(\varphi) = \{\lambda^0\}$ and $L_{SSJ}(\varphi) = \mathfrak{S}_{SSJ}(\{\lambda^0\}, \emptyset)$.

C2. Consider a second case C2 where there is a single argument A with conclusion φ , i.e. $Arg(\{\varphi\}) = \{A\}$, and assume again that $Arg(Cnt(\varphi)) = \emptyset$ and that the outcome of the argument acceptance stage consists of two labellings, i.e. $\mathfrak{L}_{AA}(\mathcal{AC}) = \{L_{AA}^1, L_{AA}^2\}$ such that $L_{AA}^1(A) = \text{IN}$ while $L_{AA}^2(A) = \text{OUT}$.

- In the AF approach, at the argument justification stage, as in the case C1, $\Sigma_{AA}(A) = \{\text{IN}, \text{OUT}\}$ and then $L_{AJ}(A) = \mathfrak{S}_{AJ}(\{\text{IN}, \text{OUT}\}) = \lambda$. Hence, at the statement justification stage, we get $\Upsilon_{AJ}(\varphi) = \{\lambda\}$ and $\Upsilon_{\overline{AJ}}(\varphi) = \emptyset$ from which $L_{ASJ}(\varphi) = \mathfrak{S}_{ASJ}(\{\lambda\}, \emptyset)$ must be the same as in case C1.
- In the SF approach, at the statement acceptance stage, we get $\Upsilon_{AA}^1(\varphi) = \{\text{IN}\}$ and $\Upsilon_{AA}^2(\varphi) = \{\text{OUT}\}$. Therefore, $L_{SA}^1(\varphi) = \mathfrak{S}_{SA}(\{\text{IN}\}, \emptyset) = \lambda^1$ and $L_{SA}^2(\varphi) = \mathfrak{S}_{SA}(\{\text{OUT}\}, \emptyset) = \lambda^2$, for some $\lambda^1, \lambda^2 \in \Lambda_{SA}$. At the statement justification stage, $\Sigma_{SA}(\varphi) = \{\lambda^1, \lambda^2\}$, and $L_{SSJ}(\varphi) = \mathfrak{S}_{SSJ}(\{\lambda^1, \lambda^2\}, \emptyset)$, which may give rise to a different outcome than in case C1.

We observe that in the cases C1 and C2 the statement justification of φ must be the same by the AF approach, while the statement justification of φ may be different by the SF approach. Hence, we conclude that the AF approach is unable to capture some distinctions which can be captured by the SF approach.

A distinction expressible only by the AF approach.

C3. Consider a case C3, where, similarly to case C1, there are two arguments A and B such that, for some statement φ , $Arg(\{\varphi\}) = \{A, B\}$ and $Arg(Cnt(\varphi)) = \emptyset$. Suppose also that the outcome of the argument acceptance stage consists of two labellings, i.e. $\mathfrak{L}_{AA}(\mathcal{AC}) = \{L_{AA}^1, L_{AA}^2\}$ such that $L_{AA}^1(A) = \text{IN}$ and $L_{AA}^1(B) = \text{IN}$, while $L_{AA}^2(A) = \text{IN}$ and $L_{AA}^2(B) = \text{OUT}$.

- In the AF approach, at the argument justification stage, we get $\Sigma_{AA}(A) = \{\text{IN}\}$ and $\Sigma_{AA}(B) = \{\text{IN}, \text{OUT}\}$. Then let $L_{AJ}(A) = \mathfrak{S}_{AJ}(\{\text{IN}\}) = \lambda$ and $L_{AJ}(B) = \mathfrak{S}_{AJ}(\{\text{IN}, \text{OUT}\}) = \lambda'$. It follows that $\Upsilon_{AJ}(\varphi) = \{\lambda, \lambda'\}$ while $\Upsilon_{\overline{AJ}}(\varphi) = \emptyset$, from which $L_{ASJ}(\varphi) = \mathfrak{S}_{ASJ}(\{\lambda, \lambda'\}, \emptyset)$.
- In the SF approach, at the statement acceptance stage, we get $\Upsilon_{AA}^1(\varphi) = \{\text{IN}\}$, $\Upsilon_{AA}^2(\varphi) = \{\text{IN}, \text{OUT}\}$, and $L_{SA}^1(\varphi) =$

$\mathfrak{S}_{SA}(\{\text{IN}\}, \emptyset) = \lambda^1$, and $L_{SA}^2(\varphi) = \mathfrak{S}_{SA}(\{\text{IN}, \text{OUT}\}, \emptyset) = \lambda^2$ for some $\lambda^1, \lambda^2 \in \Lambda_{SA}$. Then $\Sigma_{SA}(\varphi) = \{\lambda^1, \lambda^2\}$ on which $L_{SSJ}(\varphi)$ functionally depends.

C4. Consider now a case C4 which differs from case C3 because there is an additional argument labelling L_{AA}^3 , namely $\mathfrak{L}_{AA}(\mathcal{AC}) = \{L_{AA}^1, L_{AA}^2, L_{AA}^3\}$ where L_{AA}^1 and L_{AA}^2 are as in case C3, while $L_{AA}^3(A) = \text{OUT}$ and $L_{AA}^3(B) = \text{IN}$.

- In the AF approach, at the argument justification stage, we get $\Sigma_{AA}(A) = \Sigma_{AA}(B) = \{\text{IN}, \text{OUT}\}$. Then $L_{AJ}(A) = L_{AJ}(B) = \mathfrak{S}_{AJ}(\{\text{IN}, \text{OUT}\}) = \lambda'$. It follows that $\Upsilon_{AJ}(\varphi) = \{\lambda'\}$ and $L_{ASJ}(\varphi) = \mathfrak{S}_{ASJ}(\{\lambda'\}, \emptyset)$, which may give rise to a different outcome than in case C3.
- In the SF approach, at the statement acceptance stage, we get $\Upsilon_{AA}^3(\varphi) = \Upsilon_{AA}^2(\varphi) = \{\text{IN}, \text{OUT}\}$, hence $L_{SA}^3(\varphi) = L_{SA}^2(\varphi) = \lambda^2$. Then also in the case C4 we get $\Sigma_{SA}(\varphi) = \{\lambda^1, \lambda^2\}$ and hence $L_{SSJ}(\varphi)$ must be the same as in case C3.

We observe that in the cases C3 and C4 the statement justification of φ may be different by the AF approach, while the statement justification of φ must be the same by the SF approach. Hence, we conclude that the SF approach is unable to capture some distinctions which can be captured by the AF approach.

In summary, the AF and the SF approaches are incomparable in terms of expressiveness.

4 ARGUMENTATION FORMALISMS AS MLSs

In this section we illustrate how to cast argumentation formalisms into MLSs. In particular we will consider $ASPIC^+$, Assumption-Based Argumentation, and Defeasible Logic Programming and examine for each formalism whether it can be seen as an instance of the argument-focused approach, the statement-focused approach, or both. This analysis involves identifying some argument and statement labellings and the relevant synthesis operators corresponding to the original definition of each formalism. We will also show how this reconstruction in terms of MLSs and associated properties can ease the analysis and comparison of argumentation formalisms.

Due to space limitations we will not recall the definitions of these formalisms, which are typically very rich, but we will rather focus on the properties relevant for our development. The reader is referred to the original references for all the details.

Some of the considered formalisms deal with infinite and/or circular arguments. As these kinds of argument would require a specific additional treatment, due to space limitations and in order to focus on the main message of the paper, we will consider only finite and non-circular arguments in the context of each of the reviewed formalisms.

Moreover, we will assume that the monotonic or strict part of the knowledge base is consistent, that is, it does not support the derivation of contrary conclusions.

Also, we will not formally develop Example 1 for each formalism, but will refer directly to the statement justification outcome, assuming that the underlying formalisation is quite immediate in each case. As a brief informal description, we assume that two mutually attacking arguments support the statements S1 and S2, that the argument supporting the statement S3 is defeated by another (stronger) argument supporting the negation of S3 (denoted $\neg S3$), and that there are no arguments supporting S4, nor its negation.

In the remainder, all proofs are omitted due to space limitations.

4.1 Basic properties of MLSs

MLSs are useful to analyse and compare actual argumentation formalisms on a common ground consisting of abstract general properties. In particular we will consider in this paper the notions of full coverage and contrary-sensitivity.

The first property requires that the relevant functions are total.

Definition 15 (Coverage). *An AF MLS $\mathcal{L}^A = \langle \mathcal{AC}, \mathcal{L}_{AA}, L_{AJ}, L_{ASJ} \rangle$ (resp. a SF MLS $\mathcal{L}^S = \langle \mathcal{AC}, \mathcal{L}_{AA}, \mathcal{L}_{SA}, L_{SSJ} \rangle$) is said to provide*

- a full coverage of argument acceptance if every $L_{AA} \in \mathcal{L}_{AA}$ is total,
- a full coverage of argument justification (resp. of statement acceptance) if L_{AJ} (resp. every $L_{SA} \in \mathcal{L}_{SA}$) is total,
- a full coverage of statement justification if L_{ASJ} (resp. L_{SSJ}) is total.

\mathcal{L}^A (resp. \mathcal{L}^S) provides an exhaustive coverage if it provides all the three levels of full coverage introduced above.

Sensitivity to contrariness concerns statement justification only: the idea is that the justification status of a statement φ is actually somehow affected also by the contraries of φ . Formally this amounts to require that they make some difference in the evaluation.

Definition 16 (Contrary-sensitivity). *Given an AF MLS $\mathcal{L}^A = \langle \mathcal{AC}, \mathcal{L}_{AA}, L_{AJ}, L_{ASJ} \rangle$, L_{ASJ} is contrary-sensitive iff $\exists \varphi, \psi \in \mathcal{L}$ such that $\Upsilon_{AJ}(\varphi) = \Upsilon_{AJ}(\psi)$, $\Upsilon_{\overline{AJ}}(\varphi) \neq \Upsilon_{\overline{AJ}}(\psi)$, and $L_{ASJ}(\varphi) \neq L_{ASJ}(\psi)$.*

Given a SF MLS $\mathcal{L}^S = \langle \mathcal{AC}, \mathcal{L}_{AA}, \mathcal{L}_{SA}, L_{SSJ} \rangle$, L_{SSJ} is contrary-sensitive iff $\exists \varphi, \psi \in \mathcal{L}$ such that $\Sigma_{SA}(\varphi) = \Sigma_{SA}(\psi)$, $\Sigma_{\overline{SA}}(\varphi) \neq \Sigma_{\overline{SA}}(\psi)$, and $L_{SSJ}(\varphi) \neq L_{SSJ}(\psi)$.

We will use the properties of coverage and contrary-sensitivity to analyse argumentation formalisms in the remainder of this section. So, an argumentation formalism \mathfrak{F} can be considered, at a very general level, as a mechanism to produce ACSs and the relevant labellings. The universe of all ACSs possibly produced by a formalism \mathfrak{F} is denoted as $\Omega(\mathfrak{F})$. Concepts concerning MLSs and their components can be applied to argumentation formalisms considering the elements of $\Omega(\mathfrak{F})$. For instance a formalism \mathfrak{F} is said to provide an exhaustive coverage if all of the MLSs associated to every ACS in $\Omega(\mathfrak{F})$ provide an exhaustive coverage.

4.2 ASPIC⁺

ASPIC⁺ (denoted as A+ for short) is a rule-based argumentation formalism which assumes the existence of a generic language \mathcal{L} equipped with a contrariness relation [12, 10, 11].

A+ arguments may attack each other, and argument acceptance is based on Dung's formalism of argumentation frameworks [7] and its semantics. Accordingly, it is possible to refer to the labelling-based version of Dung's semantics [2], where a set of three argument acceptance labels is adopted, namely $\Lambda_{AA}^{\text{IOU}} = \{\text{IN}, \text{OUT}, \text{UN}\}$. The $\Lambda_{AA}^{\text{IOU}}$ -based argument acceptance labellings prescribed by the various abstract argumentation semantics proposed in the literature are all total. Note however that *stable* semantics fails to produce any labelling in some cases. In general, the argument acceptance phase for an argument collection \mathcal{AC} in A+ produces the acceptance evaluation $\mathcal{L}_{AA}(\mathcal{AC})$ and the relevant projection Σ_{AA} (Definition 7).

Concerning the subsequent stage, A+ focuses on argument justification, hence it belongs to the AF approach, and adopts the traditional notion of skeptical and credulous justification (see Def. 3.1 of [11])

which says that an argument is skeptically justified (denoted SKJ) if it is labelled IN in all labellings prescribed by the adopted semantics, while it is credulously justified (denoted CRJ) if it is labelled IN in some labellings. It is easy to see that, with a small adjustment to keep the two notions disjoint, it fits Definition 8.

Proposition 1. *Given the set of argument justification labels $\Lambda_{AJ}^{A+} = \{\text{SKJ}, \text{CRJ}\}$, the argument justification labelling L_{AJ}^{A+} prescribed by A+ for every $\mathcal{AC} \in \Omega(\text{A}^+)$ is such that $L_{AJ}^{A+}(A) = \mathfrak{S}_{AJ}^{A+}(\Sigma_{AA}(A))$ where the S-operator \mathfrak{S}_{AJ}^{A+} from $\Lambda_{AA}^{\text{IOU}}$ to Λ_{AJ}^{A+} is defined for $T \in \text{Pow}(\Lambda_{AA}^{\text{IOU}})$ as*

- $\mathfrak{S}_{AJ}^{A+}(T) = \text{SKJ}$ iff $T = \{\text{IN}\}$;
- $\mathfrak{S}_{AJ}^{A+}(T) = \text{CRJ}$ iff $T \supsetneq \{\text{IN}\}$.

It can be immediately observed that literally L_{AJ}^{A+} does not provide full coverage, since it does not cover the cases where $\text{IN} \notin \Sigma_{AA}(A)$. This can be explained by the emphasis on acceptance in A+. It is anyway easy to recover a full coverage by defining a third status (let say *not justified*, denoted as NOJ), covering the remaining cases, i.e. letting $\Lambda_{AJ}^{A+} = \{\text{SKJ}, \text{CRJ}, \text{NOJ}\}$.

In A+, statements inherit directly the justification status of the “best justified” argument supporting them: a statement is skeptically justified if and only if it is the conclusion of a skeptically justified argument, while it is credulously justified if and only if it is not skeptically justified and it is the conclusion of a credulously justified argument. Again, it can be proved that this fits Definition 10, applying the transfer of Definition 9 to L_{AJ}^{A+} .

Proposition 2. *Given the set of statement justification labels $\Lambda_{SJ}^{A+} = \{\text{skj}, \text{crj}\}$, the statement justification labelling L_{ASJ}^{A+} prescribed by A+ for every $\mathcal{AC} \in \Omega(\text{A}^+)$ is such that $L_{ASJ}^{A+}(\varphi) = \mathfrak{S}_{ASJ}^{A+}(\Upsilon_{AJ}(\varphi), \Upsilon_{\overline{AJ}}(\varphi))$ where the double S-operator \mathfrak{S}_{ASJ}^{A+} from Λ_{AJ}^{A+} to Λ_{SJ}^{A+} is defined for $T, U \in \text{Pow}(\Lambda_{SJ}^{A+})$ as*

- $\mathfrak{S}_{ASJ}^{A+}(T, U) = \text{skj}$ iff $\text{SKJ} \in T$;
- $\mathfrak{S}_{ASJ}^{A+}(T, U) = \text{crj}$ iff $\text{CRJ} \in T$ and $\text{SKJ} \notin T$.

L_{ASJ}^{A+} does not provide full coverage since it does not cover the cases where $\Upsilon_{AJ}(\varphi) \cap \{\text{SKJ}, \text{CRJ}\} = \emptyset$, i.e. the justification status is left undefined for all the various cases where a statement is not supported by any justified argument. Again this can be easily fixed by defining a third statement label (denoted as noj) covering the remaining cases, i.e. letting $\Lambda_{SJ}^{A+} = \{\text{skj}, \text{crj}, \text{noj}\}$. Note, anyway that by the properties of the formalism, that we can not discuss in detail due to space limitation, not all cases are possible. For instance, under some well-formedness hypotheses of the set rules, if a statement is skeptically justified its contraries cannot be skeptically justified nor credulously justified.

With the above analysis and in particular with Proposition 1 and 2 we have built an AF MLS for ASPIC⁺. It can be shown (using the same line of reasoning presented in the second part of Section 3) that it is impossible to build a SF MLS which, starting from $\mathcal{L}_{AA}(\mathcal{AC})$, produces the same statement justification labelling as ASPIC⁺ in all cases. More precisely, the situation of the arguments A and B in the case C3 presented in Section 3 can be obtained, for instance⁴, with a Dung's argumentation framework consisting of three arguments, A, B, C, where B and C mutually attack each other. Applying stable semantics to this framework gives rise to two labellings L_{AA}^1 and L_{AA}^2 whose restriction on A and B is as described in Section 3. Similarly,

⁴ We omit the underlying rule based-reasoning and admit that these small ad-hoc examples can be felt as somehow unrealistic: the same situation for A and B could be obtained in a realistic rule-based reasoning scenario with a larger number of arguments.

the situation of the arguments A and B in the case C4 can be obtained with a Dung's argumentation framework consisting of four arguments, A, B, C, D where A and D mutually attack each other, B and C mutually attack each other, and in addition D attacks C . Again, applying stable semantics to this framework gives rise to three labellings $L_{AA}^1, L_{AA}^2, L_{AA}^3$ whose restriction on A and B is as described in Section 3. Under the assumption that $Arg(\{\varphi\}) = \{A, B\}$, we get that in case C3, $L_{AJ}^+(A) = \text{SKJ}$, $L_{AJ}^+(B) = \text{CRJ}$, from which $L_{SJ}^+(\varphi) = \text{skj}$. In case C4 we get $L_{AJ}^+(A) = L_{AJ}^+(B) = \text{CRJ}$ from which $L_{ASJ}^+(\varphi) = \text{crj}$. So the justification of the statement φ is different in the two cases, while as shown in Section 3 this difference can not be obtained in the statement focused model. This shows that the argument and statement justification mechanisms adopted in $ASPIC^+$, as, defined in [11], belong to the AF camp. Since $ASPIC^+$ is a generic formalism admitting many instances, note also that this does not show that it is in general impossible to reconstruct actual instances of $ASPIC^+$ in the SF approach: there can be some instance-specific constraints preventing cases like the ones illustrated above to actually occur.

Moreover it is evident that A^+ is not contrary-sensitive, given that \mathfrak{S}_{ASJ}^+ does not actually use its second parameter U , i.e. $\Upsilon_{\overline{AJ}}(\varphi)$ does not have any effect in the definition of $L_{SJ}^+(\varphi)$. As above, this can be explained by the focus on positive support in A^+ . Hence these limitations are certainly not intrinsic to A^+ , rather they can be overcome by providing more articulated definitions for the notions of justification, leaving unchanged all the rest of the formalism.

Example 2. Referring to Example 1, we observe that according to A^+ , with every semantics, $S3$ and $S4$ get the same justification status (undefined or noj), while $\neg S3$ would be skj. The status of $S1$ and $S2$ is semantics-dependent: both would get the status crj if a Dung multiple-status semantics (e.g. preferred or stable) is adopted, while they would be equated to $S3$ and $S4$ (undefined or noj) in the case of a Dung single-status semantics (e.g. grounded or ideal)⁵.

4.3 Assumption-Based Argumentation

Assumption-Based Argumentation (denoted as ABA for short) is a rule-based argumentation formalism which, similarly to $ASPIC^+$, assumes the existence of a generic language \mathcal{L} equipped with a contrariness relation, see [13] for a tutorial. Similarly to $ASPIC^+$, ABA uses Dung's argumentation frameworks and their semantics hence the set of argument acceptance labels $\Lambda_{AA}^{\text{IOU}} = \{\text{IN}, \text{OUT}, \text{UN}\}$ and the relevant considerations are the same as for $ASPIC^+$.

The situation is more articulated concerning the other stages since both a credulous and a skeptical⁶ stance [6] have been considered in the literature. In the credulous stance (see a detailed description in [13]) a statement φ is justified if there is at least one acceptance labelling supporting φ . This can be directly reconstructed in the AF approach: using the terminology of [13], an argument is justified if it belongs to at least one *winning* set. This corresponds to being labelled IN in at least one labelling in Definition 8.

Proposition 3. Given the set of argument justification labels $\Lambda_{AJ}^{\text{AB-cr}} = \{\text{WIN}\}$, the credulous argument justification labelling

⁵ An argumentation semantics is *single-status* if, $\forall \mathcal{AC} \mid \Omega_{AA}(\mathcal{AC}) = 1$, is *multiple-status* if $\exists \mathcal{AC}$ such that $|\Omega_{AA}(\mathcal{AC})| > 1$.

⁶ Unfortunately these terms are overloaded in the literature. In particular the skeptical stance of [6] is significantly different e.g. from the notion of skeptical justification in $ASPIC^+$. To avoid to introduce a new terminology we are forced to the use of the same term with different meanings in different formalisms.

$\Lambda_{AJ}^{\text{AB-cr}}$ prescribed by ABA for every $\mathcal{AC} \in \Omega(ABA)$ is such that $L_{AJ}^{\text{AB-cr}}(A) = \mathfrak{S}_{AJ}^{\text{AB-cr}}(\Sigma_{AA}(A))$ where the S -operator $\mathfrak{S}_{AJ}^{\text{AB-cr}}$ from $\Lambda_{AA}^{\text{IOU}}$ to $\Lambda_{AJ}^{\text{AB-cr}}$ is defined for $T \in \text{Pow}(\Lambda_{AA}^{\text{IOU}})$ as $\mathfrak{S}_{AJ}^{\text{AB-cr}}(T) = \text{WIN}$ iff $T \supseteq \{\text{IN}\}$.

This labelling does not allow full coverage (since it does not cover the cases where $\text{IN} \notin T$), but it is immediate to recover full coverage by introducing a complementary "not winning" label NOWIN .

In the credulous stance of ABA , statements inherit directly the justification status from arguments: a statement is winning if it is the conclusion of a winning argument. This easily fits Definition 10.

Proposition 4. Given the set of statement justification labels $\Lambda_{SJ}^{\text{AB-cr}} = \{\text{win}\}$, the statement justification labelling $L_{ASJ}^{\text{AB-cr}}$ prescribed by ABA for every $\mathcal{AC} \in \Omega(ABA)$ is such that $L_{ASJ}^{\text{AB-cr}}(\varphi) = \mathfrak{S}_{ASJ}^{\text{AB-cr}}(\Upsilon_{AJ}(\varphi), \Upsilon_{\overline{AJ}}(\varphi))$ where the double S -operator $\mathfrak{S}_{ASJ}^{\text{AB-cr}}$ from $\Lambda_{AJ}^{\text{AB-cr}}$ to $\Lambda_{SJ}^{\text{AB-cr}}$ is defined for $T, U \in \text{Pow}(\Lambda_{AJ}^{\text{AB-cr}})$ as $\mathfrak{S}_{ASJ}^{\text{AB-cr}}(T, U) = \text{win}$ iff $\text{WIN} \in T$.

Again, literally speaking, this labelling does not provide full coverage (which can however be easily recovered with an additional label NOWIN) and it is not contrary-sensitive. Here similar observations as for the case of $ASPIC^+$ apply.

It can be easily shown that the credulous stance can also be reconstructed in the SF model. We omit it due to space limitation.

Example 3. According to the credulous stance in ABA , with every semantics, $S3$ and $S4$ get the same justification status (undefined or nowin), while $\neg S3$ is win. The status of $S1$ and $S2$ is semantics-dependent: both get the status win (thus being equated to $\neg S3$) if a multiple-status semantics is adopted, while they are equated to $S3$ and $S4$ (undefined or nowin) in the case of a single-status semantics.

In [6], in addition to the credulous statement justification reviewed above, a skeptical notion of justification is considered: basically a statement φ is skeptically justified if all acceptance labellings support φ . On this basis, a more articulated classification of statement justification, distinguishing credulously, skeptically and not justified statements can be introduced. This stance, denoted as $\text{AB} \cdot \text{sk}$ can be reconstructed in the SF approach as we illustrate below.

Definition 17. Given the set of statement acceptance labels $\Lambda_{SA}^{\text{AB-sk}} = \{\text{in}, \text{nin}\}$, the double S -operator $\mathfrak{S}_{SA}^{\text{AB-sk}}$ from $\Lambda_{AA}^{\text{IOU}}$ to $\Lambda_{SA}^{\text{AB-sk}}$ for $\text{AB} \cdot \text{sk}$ is defined for $T, U \in \text{Pow}(\Lambda_{AA}^{\text{IOU}})$ as:

- $\mathfrak{S}_{SA}^{\text{AB-sk}}(T, U) = \text{in}$ iff $\text{IN} \in T$;
- $\mathfrak{S}_{SA}^{\text{AB-sk}}(T, U) = \text{nin}$ otherwise.

So, by applying $\mathfrak{S}_{SA}^{\text{AB-sk}}$, for a given statement φ and a single acceptance labelling L_{AA} , if at least one argument supporting φ is labelled IN in L_{AA} , then $L_{SA}(\varphi) = \text{in}$, else $L_{SA}(\varphi) = \text{nin}$.

The final stage of statement justification requires the statement justification labels corresponding to skeptical, credulous and no justification, and a double S -operator, which is rather simple since, as the other cases considered above, $\text{AB} \cdot \text{sk}$ is contrary insensitive.

Proposition 5. Given the set of statement justification labels $\Lambda_{SJ}^{\text{AB-sk}} = \{\text{skj}, \text{crj}, \text{noj}\}$, the statement justification labelling $L_{SSJ}^{\text{AB-sk}}$ prescribed by ABA for every $\mathcal{AC} \in \Omega(ABA)$ is such that $L_{SSJ}^{\text{AB-sk}}(\varphi) = \mathfrak{S}_{SSJ}^{\text{AB-sk}}(\Sigma_{SA}(\varphi), \Sigma_{\overline{SA}}(\varphi))$ where the double S -operator $\mathfrak{S}_{SSJ}^{\text{AB-sk}}$ from $\Lambda_{SA}^{\text{AB-sk}}$ to $\Lambda_{SJ}^{\text{AB-sk}}$ is defined for $T, U \in \text{Pow}(\Lambda_{SA}^{\text{AB-sk}})$ as

- $\mathfrak{S}_{SSJ}^{\text{AB-sk}}(T, U) = \text{skj}$ iff $T = \{\text{in}\}$;
- $\mathfrak{S}_{SSJ}^{\text{AB-sk}}(T, U) = \text{crj}$ iff $T \supseteq \{\text{in}\}$;
- $\mathfrak{S}_{SSJ}^{\text{AB-sk}}(T, U) = \text{noj}$ otherwise.

Proposition 5 completes the reconstruction of $AB \cdot sk$ in the SF approach. It is interesting to note that such a reconstruction is not possible in the AF approach: this can be proved with the same line of reasoning used in the first part of Section 3.

Example 4. $AB \cdot sk$ behaves similarly to $A+$: with every semantics, $S3$ and $S4$ get the same justification status (undefined or noj), while $\neg S3$ would be skj. The status of $S1$ and $S2$ is semantics-dependent: both would get the status crj if a Dung multiple-status semantics (e.g. preferred or stable) is adopted, while they would be equated to $S3$ and $S4$ (undefined or noj) in the case of a Dung single-status semantics.

4.4 Defeasible Logic Programming

Defeasible Logic Programming (denoted as *DeLP*) “provides a computational reasoning system that uses an argumentation engine to obtain answers from a knowledge base represented using a logic programming language extended with defeasible rules” [8]. *DeLP* encompasses two forms of contrariness, namely strong and default negation. On this basis, defeat relations between arguments can be defined. For language statements, a notion of complement is introduced, based on strong negation only.

Differently from the approaches surveyed in the previous sections, *DeLP* does not use Dung’s framework for argument acceptance evaluation, rather it adopts a dialectical procedure. This leads to a single status approach where each argument is marked as Defeated or Undefeated, hence the set of argument acceptance labels is defined as $\Lambda_{AA}^{De} = \{D, U\}$. In other words the dialectical procedure is guaranteed to produce a unique total Λ_{AA} -based acceptance labelling: for every $\mathcal{AC} \in \Omega(DeLP) \mid \mathfrak{L}_{AA}(\mathcal{AC}) = 1$. Based on this property, in the AF approach acceptance projection (Definition 7) and argument justification (Definition 8) can be defined essentially as identity.

Proposition 6. Given the set of argument justification labels $\Lambda_{AJ}^{De} = \{D, U\}$, the argument justification labelling L_{AJ}^{De} prescribed by *DeLP* for every $\mathcal{AC} \in \Omega(DeLP)$ is such that $L_{AJ}^{De}(A) = \mathfrak{S}_{AJ}^{De}(\Sigma_{AA}(A))$ where the S -operator \mathfrak{S}_{AJ}^{De} from Λ_{AA}^{De} to Λ_{AJ}^{De} is defined for $T \in Pow(\Lambda_{AA}^{De})$ as $\mathfrak{S}_{AJ}^{De}(T) = D$ iff $T = \{D\}$; $\mathfrak{S}_{AJ}^{De}(T) = U$ iff $T = \{U\}$.

A statement is said to be warranted if it is the conclusion of an argument whose justification label is U . On this simple basis, an articulated notion of justification status for a statement φ based on four labels (corresponding to the possible answers to a *DeLP* query) is introduced: yes if φ is warranted; no if the complement of φ is warranted; und(ecided) if neither φ nor its complement are warranted; unk(nown) if φ is not in the signature of the program. This fits Definition 10 as shown by Proposition 7 (see also Table 1).

Proposition 7. Given the set of statement justification labels $\Lambda_{SJ}^{De} = \{\text{yes, no, und, unk}\}$, for every $\mathcal{AC} \in \Omega(DeLP)$, the statement justification labelling L_{SJ}^{De} prescribed by *DeLP* is such that $L_{SJ}^{De}(\varphi) = \mathfrak{S}_{ASJ}^{De}(\Upsilon_{AJ}(\varphi), \Upsilon_{\overline{AJ}}(\varphi))$ where the double S -operator \mathfrak{S}_{ASJ}^{De} from Λ_{AJ}^{De} to Λ_{SJ}^{De} is defined for $T, U \in Pow(\Lambda_{SJ}^{De})$ as

- $\mathfrak{S}_{ASJ}^{De}(T, U) = \text{yes}$ iff $U \in T$;
- $\mathfrak{S}_{ASJ}^{De}(T, U) = \text{no}$ iff $U \in \overline{U}$;
- $\mathfrak{S}_{ASJ}^{De}(T, U) = \text{und}$ iff $T \cup U = \{D\}$;
- $\mathfrak{S}_{ASJ}^{De}(T, U) = \text{unk}$ iff $T \cup U = \emptyset$.

In contrast with the previous formalisms, this labelling provides a full coverage, since all possible cases for a statement are considered (note in particular that it is guaranteed that $U \notin \Upsilon_{AJ}(\varphi) \cap \Upsilon_{\overline{AJ}}(\varphi)$, i.e. a statement and its complement cannot be both warranted). In

$\Upsilon_{\overline{AJ}}(\varphi) \setminus \Upsilon_{AJ}(\varphi)$	\emptyset	$\{U\}$	$\{D\}$	$\{U, D\}$
\emptyset	unk	yes	und	yes
$\{U\}$	no	n.a.	no	n.a.
$\{D\}$	und	yes	und	yes
$\{U, D\}$	no	n.a.	no	n.a.

Table 1. Justification status of statement φ depending on $\Upsilon_{AJ}(\varphi)$ and $\Upsilon_{\overline{AJ}}(\varphi)$ in *DeLP*.

particular it distinguishes three cases of non acceptance (while non acceptance was overlooked in the previously surveyed formalisms). This is also due to the fact that this approach is contrary sensitive.

Example 5. *DeLP* would label $S3$ as no, $\neg S3$ as yes, both $S1$ and $S2$ as und, and $S4$ as unk.

Since *DeLP* is single-status at the stage of argument acceptance, it encompasses a single notion of positive justification, while, for instance, *ASPIC*⁺ distinguishes credulous and skeptical justification. This suggests that combining the most expressive aspects of different approaches may give rise to a more general treatment of the notion of argument and statement justification. Further, it can be shown that (omitted due to space limitations) the *DeLP* statement justification labelling can be reconstructed in the SF approach too.

5 TOWARDS TUNABLE JUSTIFICATION

From the analysis in the previous section, it emerges that different argumentation formalisms adopt quite different notions of justification, both concerning arguments and statements, featuring different properties and sometimes failing to satisfy some intuitive requirements like full coverage and contrary-sensitivity. However these differences do not seem to be caused by technical motivations, but rather to depend on arbitrary choices based on the intended use of the notion of justification in the presentation of the formalisms themselves. Moreover, some proposals (*ABA* in the credulous stance and *DeLP*) fit both the AF and SF approach, while others (*ABA* in the sceptical stance and *ASPIC*⁺) fit only one (the SF and AF approach respectively). These observations back up our claim that the notion of justification (and in particular of statement justification) has been somehow neglected in the development of argumentation formalisms, often more focused on the notion of argument acceptance. Moreover they suggest that justification notions, instead of being “hardwired” in the definitions, could better be conceived as tunable components of any argumentation formalism, with a role similar to those played by argumentation semantics in *ASPIC*⁺ and *ABA*. These formalisms do not stick to a single argumentation semantics, rather they assume that one is chosen among the various available ones (including possibly those to be developed in the future).

We aim now at illustrating how our model can be used to build alternative options for statement justification, by providing an example of a generic approach to statement justification, the so-called ignorance-aware labelling. Due to space limits, we do this for the AF approach only, analogous ideas are applicable to the SF approach too.

The ignorance-aware labelling captures different reasons for which a statement is not justified: this may be because it is falsified in some way, or due to some lack of knowledge or because the available knowledge carries some undecidedness. To support this distinction, we assume the set of labels $\Lambda_{SJ}^a = \{\text{yes, fal, unk, ni}\}$, where the label yes indicates that the statement is justified, the label fal indicates that the statement is falsified, unk stands for an ‘unknown’ statement, while ni captures other cases of indecision about the statement.

Assuming accordingly the set of labels $\Lambda_{S_j}^a$ and full coverage as a basic requirement, the adoption of the ignorance-aware labelling in a formalism essentially amounts to specify for each possible pair $(\Upsilon_{A_j}(\varphi), \Upsilon_{\bar{A}_j}(\varphi))$ the corresponding yes, fal, unk or ni label, i.e a double S-operator $\mathfrak{S}_{AS_j}^{la}$ from Λ_{A_j} to $\Lambda_{S_j}^a$, where Λ_{A_j} is the set of justification labels adopted in the formalism. For the sake of conciseness, in the following we will leave implicit the definition of the S-operator $\mathfrak{S}_{AS_j}^{la}$ and will express directly the dependence of the statement labelling on $\Upsilon_{A_j}(\varphi)$ and $\Upsilon_{\bar{A}_j}(\varphi)$.

Let us consider $ASPIC^+$ first and assume $\Lambda_{A_j}^{A^+} = \{SKJ, CRJ, NOJ\}$. A first, skeptically oriented, option corresponds to the idea that a statement is labelled yes if it is supported by a skeptically justified argument. A second, credulously oriented, option labels a statement yes if it is supported by a skeptically or credulously justified argument.

Definition 18. *The skeptically ignorance-aware labelling for $ASPIC^+$ is defined as follows:*

- $L_{AS_j}^{la-A^+-sk}(\varphi) = \text{yes}$ iff $SKJ \in \Upsilon_{A_j}(\varphi)$;
- $L_{AS_j}^{la-A^+-sk}(\varphi) = \text{fal}$ iff $SKJ \in \Upsilon_{\bar{A}_j}(\varphi)$ ⁷;
- $L_{AS_j}^{la-A^+-sk}(\varphi) = \text{unk}$ iff $\Upsilon_{\bar{A}_j}(\varphi) \cup \Upsilon_{A_j}(\varphi) = \emptyset$;
- $L_{AS_j}^{la-A^+-sk}(\varphi) = \text{ni}$ otherwise.

Definition 19. *The credulous ignorance-aware labelling for $ASPIC^+$ is defined as follows*

- $L_{AS_j}^{la-A^+-cr}(\varphi) = \text{yes}$ iff $\{SKJ, CRJ\} \cap \Upsilon_{A_j}(\varphi) \neq \emptyset$;
- $L_{AS_j}^{la-A^+-cr}(\varphi) = \text{fal}$ iff $\{SKJ, CRJ\} \cap \Upsilon_{A_j}(\varphi) = \emptyset$ and $\{SKJ, CRJ\} \cap \Upsilon_{\bar{A}_j}(\varphi) \neq \emptyset$;
- $L_{AS_j}^{la-A^+-cr}(\varphi) = \text{unk}$ iff $\Upsilon_{\bar{A}_j}(\varphi) \cup \Upsilon_{A_j}(\varphi) = \emptyset$;
- $L_{AS_j}^{la-A^+-cr}(\varphi) = \text{ni}$ otherwise.

Example 6. *In the skeptically labelling, $S1$ and $S2$ are labelled as ni, $S3$ as fal, $\neg S3$ as yes, while in the credulous labelling $S1$ and $S2$ are labelled as yes, $S3$ as fal, $\neg S3$ as yes. In both the skeptically and credulous version, $S4$ is labelled unk.*

Turning to ABA (in the credulous stance), the same observations as for $ASPIC^+$ apply to the ignorance-aware labelling counterpart.

Definition 20. *The skeptically ignorance-aware labelling for ABA is defined as follows*

- $L_{AS_j}^{la-AB-sk}(\varphi) = \text{yes}$ iff $WIN \in \Upsilon_{A_j}(\varphi)$ and $WIN \notin \Upsilon_{\bar{A}_j}(\varphi)$;
- $L_{AS_j}^{la-AB-sk}(\varphi) = \text{fal}$ iff $WIN \notin \Upsilon_{A_j}(\varphi)$ and $WIN \in \Upsilon_{\bar{A}_j}(\varphi)$;
- $L_{AS_j}^{la-AB-sk}(\varphi) = \text{unk}$ iff $\Upsilon_{\bar{A}_j}(\varphi) \cup \Upsilon_{A_j}(\varphi) = \emptyset$;
- $L_{AS_j}^{la-AB-sk}(\varphi) = \text{ni}$ otherwise.

Definition 21. *The credulous ignorance-aware labelling for ABA is defined as follows*

- $L_{AS_j}^{la-AB-cr}(\varphi) = \text{yes}$ iff $WIN \in \Upsilon_{A_j}(\varphi)$;
- $L_{AS_j}^{la-AB-cr}(\varphi) = \text{fal}$ iff $WIN \notin \Upsilon_{A_j}(\varphi)$ and $WIN \in \Upsilon_{\bar{A}_j}(\varphi)$;
- $L_{AS_j}^{la-AB-cr}(\varphi) = \text{unk}$ iff $\Upsilon_{\bar{A}_j}(\varphi) \cup \Upsilon_{A_j}(\varphi) = \emptyset$;
- $L_{AS_j}^{la-AB-cr}(\varphi) = \text{ni}$ otherwise.

Example 7. *In the skeptically labelling for ABA , $S1$ and $S2$ are labelled as ni, $S3$ as fal, $\neg S3$ as yes, $S4$ as unk, while in the credulous case, $S1$ and $S2$ are labelled as yes, $S3$ as fal, $\neg S3$ as yes, $S4$ as unk.*

As to $DeLP$, the original labelling reified in Proposition 7 is ignorance-aware, modulo two label names: no for fal, und for ni.

Definition 22. *The (skeptical) ignorance-aware labelling for $DeLP$ is defined as follows*

- $L_{AS_j}^{la-De}(\varphi) = \text{yes}$ iff $U \in \Upsilon_{A_j}(\varphi)$;
- $L_{AS_j}^{la-De}(\varphi) = \text{fal}$ iff $U \in \Upsilon_{\bar{A}_j}(\varphi)$;
- $L_{AS_j}^{la-De}(\varphi) = \text{unk}$ iff $\Upsilon_{\bar{A}_j}(\varphi) \cup \Upsilon_{A_j}(\varphi) = \emptyset$;
- $L_{AS_j}^{la-De}(\varphi) = \text{ni}$ otherwise.

Example 8. *According to Definition 22, $S1$ and $S2$ are labelled as ni, $S3$ as fal, $\neg S3$ as yes, $S4$ as unk.*

It can be noted that the credulous and skeptical versions of the ignorance-aware labelling provide (respectively) coincident results for all the formalisms considered, while the outcomes were different (not only formally but also substantially) with the original definitions.

Example 9. *For every statement S in $\{S1, S2, S3, \neg S3, S4, S5\}$:*

- $L_{AS_j}^{la-A^+-sk}(S) = L_{AS_j}^{la-AB-sk}(S) = L_{AS_j}^{la-De}(S)$, and
- $L_{AS_j}^{la-A^+-cr}(S) = L_{AS_j}^{la-AB-cr}(S)$.

6 CONCLUSION

Argument-based reasoning is a complex activity which is based on, but is not limited to, the tasks of argument production and argument acceptance evaluation, on which many current literature formalisms are mainly focused. In particular, treating statement justification as a simple byproduct of the previous reasoning stages tends to hide the conceptual richness of this task too. As we have shown, this gives rise to disagreements and/or losses of sensitivity in some well-known formalisms even in simple common-sense examples.

To overcome these limits, we have proposed the novel notion of multi-labelling system for argument-based reasoning, which restores statement justification as a first-class formalism-independent component of the overall process and promotes the idea that it is tunable, much in the way argumentation semantics is a tunable component in several formalisms. We have shown that at least two approaches, namely the AF and the SF approach, can be considered in this context and that they are incomparable in terms of expressiveness. We have then shown how the process leading from argument acceptance to statement justification in three well-known argumentation formalisms can be regarded as a kind of either AF or SF multi-labelling system. Multi-labelling systems can be tuned, and we illustrated this by ‘plugging’ a (so-called ignorance-aware) labelling for statement justification into the three considered formalisms, thus achieving agreement in the example used in the paper.

Overall, the paper provides a first foundational contribution towards a deeper study of statement justification in argument-based reasoning and opens the way to several future research directions. In particular, we mention a systematic study of general principles and properties for statement labellings. This would represent a complementary contribution with respect to the literature works on rationality postulates [5, 1] which deal with the collective properties (e.g. consistency and closure) of the conclusions of a set of arguments rather than with the notion of justification of the conclusions themselves. Moreover we will consider the investigation of other pluggable statement justification methods and of their relationships, and the revision of some of the assumptions underlying the approaches considered in this paper. For instance, revising some of these assumptions may allow one to overcome the expressiveness gaps evidenced in Section 3. Finally, a first analysis carried out on Defeasible Logic [9] shows the necessity, in this context, of more articulated approaches taking into account the different types of arguments and attacks present in the underlying formalism.

⁷ We don't specify the additional condition $SKJ \notin \Upsilon_{A_j}(\varphi)$ since, by the properties of $ASPIC^+$, it is already implied by $SKJ \in \Upsilon_{\bar{A}_j}(\varphi)$.

REFERENCES

- [1] L. Amgoud and P. Besnard, 'Logical limits of abstract argumentation frameworks', *Journal of Applied Non-Classical Logics*, **23**(3), 229–267, (2013).
- [2] P. Baroni, M. Caminada, and M. Giacomin, 'An introduction to argumentation semantics', *Knowledge Eng. Review*, **26**(4), 365–410, (2011).
- [3] P. Baroni, M. Giacomin, and B. Liao, 'Dealing with generic contrariness in structured argumentation', in *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence, IJCAI 2015*, pp. 2727–2733, (2015).
- [4] P. Baroni, G. Governatori, H.-P. Lam, and R. Riveret, 'On the justification of statements in argumentation-based reasoning', in *Proc. of the 15th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2016)*, pp. 521–524, (2016).
- [5] M. Caminada and L. Amgoud, 'On the evaluation of argumentation formalisms', *Artif. Intell.*, **171**(5-6), 286–310, (2007).
- [6] Y. Dimopoulos, B. Nebel, and F. Toni, 'On the computational complexity of assumption-based argumentation for default reasoning', *Artif. Intell.*, **141**(1-2), 57–78, (2002).
- [7] P. M. Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games', *Artif. Intell.*, **77**(2), 321–358, (1995).
- [8] A. J. Garcia and G. R. Simari, 'Defeasible logic programming: DeLP servers, contextual queries, and explanations for answers', *Argument & Computation*, **5**(1), 63 – 88, (2014).
- [9] G. Governatori, M.J. Maher, G. Antoniou, and D. Billington, 'Argumentation semantics for defeasible logic', *J. Log. Comput.*, **14**(5), 675–702, (2004).
- [10] S. Modgil and H. Prakken, 'A general account of argumentation with preferences', *Artif. Intell.*, **195**, 361 – 397, (2013).
- [11] S. Modgil and H. Prakken, 'The ASPIC⁺ framework for structured argumentation: a tutorial', *Argument & Computation*, **5**(1), 31 – 62, (2014).
- [12] H. Prakken, 'An abstract framework for argumentation with structured arguments', *Argument & Computation*, **1**(2), 93–124, (2010).
- [13] F. Toni, 'A tutorial on assumption-based argumentation', *Argument & Computation*, **5**(1), 89 – 117, (2014).

Annotate-Sample-Average (ASA): A New Distant Supervision Approach for Twitter Sentiment Analysis

Felipe Bravo-Marquez and Eibe Frank and Bernhard Pfahringer¹

Abstract. The classification of tweets into polarity classes is a popular task in sentiment analysis. State-of-the-art solutions to this problem are based on supervised machine learning models trained from manually annotated examples. A drawback of these approaches is the high cost involved in data annotation. Two freely available resources that can be exploited to solve the problem are: 1) large amounts of unlabelled tweets obtained from the Twitter API and 2) prior lexical knowledge in the form of opinion lexicons. In this paper, we propose Annotate-Sample-Average (ASA), a distant supervision method that uses these two resources to generate synthetic training data for Twitter polarity classification. Positive and negative training instances are generated by sampling and averaging unlabelled tweets containing words with the corresponding polarity. Polarity of words is determined from a given polarity lexicon. Our experimental results show that the training data generated by ASA (after tuning its parameters) produces a classifier that performs significantly better than a classifier trained from tweets annotated with emoticons and a classifier trained, without any sampling and averaging, from tweets annotated according to the polarity of their words.

1 Introduction

Twitter² is a service in which users can post messages or tweets limited to 140 characters and subscribe to tweets posted by other users. It has become the most popular microblogging platform, with hundreds of millions of users who produce millions of posts on a daily basis. The great volume of publicly available social data that is published in Twitter has made it a rich resource for sentiment analysis [9].

A popular approach for classifying tweets (posts in Twitter) into polarity classes is to represent tweets from a corpus of hand-annotated tweets by feature vectors and train supervised models on them [18]. However, considering that annotation of tweets into sentiment classes is a time-consuming and labour-intensive task, supervised models can be impractical in the absence of labelled tweets.

Distant supervision models are heuristic labelling functions [16] for creating training data from unlabelled corpora. These models have been widely adopted for Twitter sentiment analysis because large amounts of unlabelled tweets can be easily obtained through the use of the Twitter API. A well-known distant supervision approach for Twitter polarity classification is the emoticon-annotation approach (EAA), in which tweets with positive :) or negative :(emoticons are labelled according to the polarity indicated by the

emoticon after removing the emoticon from the content [22]. This method is affected by two main limitations:

1. The removal of all tweets without emoticons may cause a loss of valuable information.
2. There are many domains such as politics, in which emoticons are not frequently used to express positive and negative opinions.

Opinion lexicons are another type of resource that has been used for supporting the sentiment analysis of tweets. An opinion lexicon is a list of terms or *opinion words* annotated according to sentiment categories such as positive and negative. Examples of positive opinion words are **love** and **happy**, and examples of negative opinion words are **disgusting** and **horrible**. There are several opinion lexicons freely available on the web, e.g., *SentiWordNet*³, *MPQA Subjectivity Lexicon*⁴, and *AFINN*⁵. Opinion lexicons can be used as prior lexical knowledge for calculating the sentiment of tweets [27], and to extract message-level features for training classifiers [3, 9, 18].

In this paper we propose a distant supervision method called **Annotate-Sample-Average** (ASA) for training polarity classifiers in Twitter in the absence of labelled data. ASA takes a collection of unlabelled tweets and a polarity lexicon composed of positive and negative words and creates synthetic labelled instances for Twitter polarity classification. Each labelled instance is created by sampling with replacement a number of tweets containing at least one word from the lexicon with the desired polarity, and averaging the feature vectors of the sampled tweets. This allows the usage of any kind of features for representing the tweets, e.g., unigrams and part-of-speech tags (POS) tags.

Polarity lexicons are normally formed by thousands of opinion words, so there is a high probability that a tweet contains at least one word from the lexicon, which means that ASA can potentially exploit more unlabelled data than EAA because the latter is based on a small number of positive and negative emoticons.

The intuition behind ASA is that a tweet containing a word with a certain known positive or negative polarity has a certain likelihood of expressing the same polarity in the whole message. Of course, the opposite polarity may also be expressed due to the presence of negation, sarcasm, or other opinion words with the opposite polarity. We propose a hypothesis, which we refer to as the “lexical polarity hypothesis”, stating that the first scenario is more likely than the second one. Based on that, when sampling and averaging multiple tweets exhibiting at least one word with the desired positive or negative polarity, we increase the confidence of obtaining a vector located

¹ University of Waikato, New Zealand, email: fjb11@students.waikato.ac.nz, {eibe,bernhard}@cs.waikato.ac.nz

² <http://www.twitter.com>

³ <http://sentiwordnet.isti.cnr.it/>

⁴ http://mpqa.cs.pitt.edu/opinionfinder/opinionfinder_2/

⁵ <http://neuro.imm.dtu.dk/wiki/AFINN>

in the region of the desired polarity.

Most sentiment analysis datasets are imbalanced in favor of positive examples [12]. This is presumably because users are more likely to report positive than negative opinions. The shortcoming of training sentiment classifiers from imbalanced datasets is that many classification algorithms tend to predict test samples as the majority class [7] when trained from this type of data. A popular way to address this problem is to rebalance the data by under-sampling the majority class or by over-sampling the minority class. A noteworthy property of ASA is that it incorporates a rebalancing mechanism in which balanced training data can be generated.

We compare classifiers trained with ASA against other distant supervision methods on three collections of hand-annotated polarity tweets. The baselines we consider are EAA and a lexicon-based annotation approach (LAA) that annotates tweets according to the polarity of their words. The experimental results show that ASA, with appropriate choice of the number of tweets averaged for each generated instance, outperforms the other methods in all cases.

This article is organised as follows. In Section 2, we provide a review of related work. In Section 3, we describe the proposed ASA method. The lexical polarity hypothesis is empirically studied in Section 4. The evaluation of the method is presented in Section 5. The conclusions are discussed in Section 6.

2 Related Work

State-of-the-art solutions for Twitter polarity classification are based on supervised techniques such as logistic regression and support vector machines trained from hand-annotated polarity corpora. Some of the features used for describing the tweets are: n-grams, POS tags, Brown clusters [4], and features derived from polarity lexicons [18].

In the absence of training data most previous distant supervision approaches for Twitter sentiment analysis rely on strong sentiment signals such as emoticons or hashtags e.g., #joy, #sadness, for labelling tweets into positive and negative polarity classes, after dropping these signals from the content [6].

Emoticon-annotated tweets have been used for a variety of sentiment analysis tasks: training of polarity classifiers [6, 20], training incremental classifiers from Twitter streams [2], fitting sentiment-oriented language models [14], inducing polarity lexicons [18], and initialising the parameters of deep neural networks [23, 26].

Other types of knowledge are lexical knowledge provided by opinion lexicons and contextual knowledge provided by unlabelled corpora. There is some work exploiting these sources of knowledge for training document-level sentiment classifiers from small collections of labelled documents. In [24], words and documents are jointly represented by a bipartite graph of labelled and unlabelled nodes. The sentiment labels of words and documents are propagated to the unlabelled nodes using regularised least squares. In [13], the term-document matrix associated with a corpus of documents is factorised into three matrices specifying cluster labels for words and documents using a constrained non-negative tri-factorisation technique. Sentiment-annotated words and documents are introduced into the model as optimisation constraints. A generative naive Bayes model based on a polarity lexicon, which is then refined using sentiment-annotated documents, is proposed in [15]. Zhang et al. [29] proposed a lexicon-based approach for annotating unlabelled tweets into polarity classes regarding a given entity by aggregating the polarities of words from a lexicon with positive and negative words using a scoring function. The automatically labelled tweets are then used for training a classifier.

Another approach based on distant supervision and lexical prior knowledge is proposed in [25]. The authors build a graph that has users, tweets, words, hashtags, and emoticons as its nodes. A subset of these nodes is labelled by prior sentiment knowledge provided by a polarity lexicon, the known polarity of emoticons, and a message-level classifier trained with emoticons. These sentiment labels are propagated throughout the graph using random walks.

A semi-supervised model for imbalanced sentiment classification is proposed in [12]. The model exploits both labelled and unlabelled documents by iteratively performing under-sampling of the majority class in a co-training framework using random subspaces of features.

The ASA method proposed in this paper exploits prior lexical knowledge and unlabelled data for creating synthetic polarity data by sampling and averaging multiple tweets without requiring any labelled tweets. ASA works on the whole message rather than being entity oriented as the method in [29]. Moreover, ASA can be used for creating training data with any size and distribution of labels and hence may be useful for dealing with the class imbalance problem reported in [12]. To the best of our knowledge, this is the first distant supervision method for Twitter sentiment analysis with these characteristics.

3 Annotate-Sample-Average Method

In this section, we describe the Annotate-Sample-Average (ASA) algorithm for generating training data for Twitter polarity classification. The method receives two data inputs: 1) the source corpus, and 2) the opinion lexicon.

The source corpus is a collection of unlabelled tweets \mathcal{C} on which the generated instances are based. The corpus can be built using the public Twitter API⁶, which allows the retrieval of public tweets. The tweets must be written in the same language as the opinion lexicon, and the type of tweets included in the collection should depend on the type of sentiment classifier intended to be built. For instance, in order to build a domain-specific sentiment classifier (e.g., for a political election), the collection should be restricted to tweets associated with the target domain. This can be done using the Twitter API by specifying key words, users, or geographical areas. In this work, we focus on domain-independent polarity classification. Thus, we consider a general purpose collection of English tweets.

The opinion lexicon \mathcal{L} is a list of words labelled by sentiment. In this work, we consider positive and negative sentiment categories. The positive and negative subsets of the lexicon are denoted by symbols \mathcal{L}_+ and \mathcal{L}_- respectively. Several existing opinion lexicons can be used here. There are basically two families of lexicons that can be considered:

1. Manually annotated lexicons, in which the sentiment of the words is annotated according to human judgements. Crowdsourcing tools such as Amazon Mechanical Turk can be used to support the annotation [17].
2. Automatically-annotated lexicons that are created by automatically expanding a small set of opinion words using relations provided by semantic networks, e.g., synonyms, and antonyms [8], or using statistical associations calculated from document corpora, e.g., point-wise mutual information [28].

Manually-annotated lexicons tend to be smaller than the automatically made ones. Conversely, automatically-annotated lexicons are likely to be noisy and may include several neutral words that are not

⁶ <https://dev.twitter.com/overview/api>

very useful for polarity classification [3]. In this work, we use the AFINN lexicon [1], which is a manually annotated lexicon formed by 1176 positive words and 2204 negative words. The lexicon includes informal words commonly found in Twitter such as slang, obscene words, acronyms and Web jargon. It is important to mention that AFINN does not include any emoticons.

The other parameters of ASA are: a , which determines the number of tweets to be averaged for each generated instance, p , which corresponds to the number of positive instances to be generated, n , corresponding to the number of negative instances, and m , which is a flag specifying how to handle tweets with both positive and negative words.

The tweets from \mathcal{C} are preprocessed according to the procedure proposed in [6]. All tweets are lowercased and tokenised. The words are simplified by replacing sequences of letters occurring more than two times with two occurrences of the letter (e.g., huuungry is reduced to huungry, loooove to loove) and replacing user mentions and URLs with the generic tokens “USER” and “URL”, respectively.

The first step of the algorithm is the **annotation** phase, in which the tweets from \mathcal{C} are annotated according to the prior sentiment knowledge provided by the lexicon. Every time a positive word from \mathcal{L}_+ is found in a message, the whole tweet is added to a set called **posT**; analogously, if a negative word is found in \mathcal{L}_- , the tweet is added to a set called **negT**. Tweets with both positive and negative words will be discarded if the flag m is set, and will be simultaneously added to both **posT** and **negT** otherwise.

The tweets contained in **posT** and **negT** are candidates for building the synthetic labelled instances. The assumption here is that tweets in each set, positive and negative, are more probable to express the corresponding polarity in the whole message than the opposite polarity. This can be explained by the the short length of tweets. As tweets are short straight-to-the-point messages, the presence of a polarity word has a strong correlation with the overall polarity expressed in the message. For example, the tweet: “*Hey guess what? I think you’re awesome*” contains the word *awesome* and is clearly expressing a positive sentiment. Conversely, there also tweets with opinion words than can express the opposite polarity, e.g., “*Not happy where I’m at in life*”. This can occur due to several factors such as the presence of other words with the opposite polarity, negations, or sarcasm. However, we hypothesise that the first situation is more likely than the the second one. We refer to this hypothesis as the “lexical polarity hypothesis” and we study it empirically in Section 4.

We represent all the candidate tweets by vectors of features. We consider three type of features, which are concatenated for building the feature space. These features have been proven to be useful for analysing the sentiment of tweets [18]:

1. Word unigrams (UNI): a vector space model based on counting the frequency of unigrams.
2. Brown clusters (BWN): a vector space model based on counting the frequency of word clusters trained with the Brown clustering algorithm [4]. This algorithm produces hierarchical clusters of words by maximising the mutual information of bigrams.
3. Part-of-speech tags (POS): a vector space model based on counting the frequency of each POS tag in the message.

The second step of ASA is the **sampling** step. ASA randomly samples with replacement a tweets from either **posT** or **negT** for each generated instance. Next, in the **averaging** step the feature vectors of the sampled tweets are averaged and labelled according to the polarity of the set from which they were sampled. The rationale behind

this step is that, assuming that the “lexical polarity hypothesis” holds, averaging multiple tweets sampled from the same set increases the confidence of generating instances located in the region of the desired polarity.

We define the random variable D as the event of sampling a tweet with the desired positive or negative polarity from either **posT** or **negT**. We assume that D is distributed with a Bernoulli distribution of parameter p_d . According to the lexical polarity hypothesis, $p_d > 0.5$. We define another random variable M as the event that the majority of the a randomly sampled tweets from **posT** or **posN** have the desired polarity. This is equivalent to saying that at least $\lfloor \frac{a}{2} \rfloor + 1$ tweets from the sample have the desired positive or negative polarity. If we assume that the tweets in **posT** and **negT** are independent and identically distributed (IID), the probability of M can be calculated by adding the values of the Binomial probability mass function from $\lfloor \frac{a}{2} \rfloor + 1$ to a . This corresponds to adding all the cases in which more than the half of the sampled tweets (the majority) have the desired polarity. This probability is calculated as follows:

$$P(M) = \sum_{i=\lfloor \frac{a}{2} \rfloor + 1}^a \binom{a}{i} p_d^i (1 - p_d)^{a-i}$$

Note that this value is equivalent to 1 minus the cumulative distribution function of the Binomial distribution evaluated at $\lfloor \frac{a}{2} \rfloor$. The probabilities of M for different values of a ($a \geq 3$) and p_d ($p_d > 0.5$) are shown in Table 1.

	$p_d = 0.6$	$p_d = 0.7$	$p_d = 0.8$	$p_d = 0.9$
$a = 3$	0.648	0.784	0.896	0.972
$a = 5$	0.683	0.837	0.942	0.991
$a = 10$	0.633	0.850	0.967	0.998
$a = 50$	0.902	0.998	1	1
$a = 100$	0.973	1	1	1
$a = 500$	1	1	1	1
$a = 1000$	1	1	1	1

Table 1. Probabilities of sampling a majority of tweets with the desired polarity.

From the table, we observe that all the calculated probabilities are greater than p_d and generally increase when increasing p_d or a (exceptions occur when switching from an odd to an even number of votes). Thus, assuming the lexical polarity hypothesis is true and $p_d > 0.5$ for **posT** and **negT**, we can say that the majority of the tweets sampled by ASA have the desired polarity with a probability greater than p_d . We can expect that the instances produced by ASA will behave similarly to the majority of the instances they are obtained from. Thus, compared to sampling individual tweets, we can have greater confidence that ASA instances will be in the desired polarity region.

The ideas discussed above are inspired by Condorcet’s Jury Theorem, which is used in the context of decision making. The theorem states that if a random individual votes for the correct decision with probability $p_d > 0.5$, the probability of the majority being correct tends to one when increasing the number of independent voters. This is a consequence of the law of great numbers, and as was shown in [10], the same conclusions can be obtained after relaxing the independence assumption.

In our problem, each tweet sampled from **posT** or **negT** can be interpreted as a vote for the polarity of the averaged instance. We expect a trade-off in the value of a . While a small value of a will decrease the confidence of generating an instance with the target

```

Algorithm  $ASA(C, \mathcal{L}, a, p, n, m)$ 
  foreach  $tweet \in C$  do
    if  $m$  and  $(hasWord(tweet, \mathcal{L}_+) \text{ and } hasWord(tweet, \mathcal{L}_-))$ 
      then
        continue
    if  $hasWord(tweet, \mathcal{L}_+)$  then
       $tweetVec \leftarrow extractFeatures(tweet)$ 
       $posT.put(tweetVec)$ 
    if  $hasWord(tweet, \mathcal{L}_-)$  then
       $tweetVec \leftarrow extractFeatures(tweet)$ 
       $posN.put(tweetVec)$ 
  end
   $i \leftarrow 0$ 
  while  $i \leq p$  do
     $pInst \leftarrow sampleAndAverage(posT, a)$ 
     $pInst.label \leftarrow pos$ 
     $\mathcal{O}.put(pInst)$ 
     $i \leftarrow i + 1$ 
  end
   $i \leftarrow 0$ 
  while  $i \leq n$  do
     $nInst \leftarrow sampleAndAverage(negT, a)$ 
     $nInst.label \leftarrow neg$ 
     $\mathcal{O}.put(nInst)$ 
     $i \leftarrow i + 1$ 
  end
  return  $\mathcal{O}$ ;
Procedure  $sampleAndAverage(T, a)$ 
   $i \leftarrow 0$ 
   $inst \leftarrow newZeroVector$ 
  while  $i \leq a$  do
     $x \leftarrow randomSample(T)$ 
     $inst \leftarrow inst + (x/a)$ 
     $i \leftarrow i + 1$ 
  end
  return  $inst$ ;

```

Algorithm 1: ASA ALGORITHM

polarity, a very large value will generate instances that, despite being likely to have the right label, will be very similar to each other. This could affect the generalisation ability of a classifier trained from those instances.

The resulting training dataset \mathcal{O} is created by repeating the sample and average steps p times for the positive class and n times for the negative one. The pseudo-code of ASA is given in Algorithm 1.

Setting the flag m in the algorithm will generate polarity instances from tweets in which words from the opposite polarity are never observed. Considering that positive and negative tweets are likely to contain words with the opposite polarity, we expect that unsetting the flag will produce instances with better generalisation properties. Both setups are compared in Section 5.

We use ASA for creating balanced training data by setting p and n to the same value. This is done to address the sentiment imbalance problem discussed in [12]: classifiers trained from imbalanced datasets may have difficulties recognising the minority class. The balancing properties of ASA are inspired by a well-known resampling technique used for training classifiers from imbalanced datasets called Synthetic Minority Over-sampling Technique (SMOTE) [5]. SMOTE oversamples the minority class by generating synthetic examples for the minority class. Each new instance is calculated as a

random weighted average between an existing example of the minority class and one of its nearest neighbours. The similarity between ASA and SMOTE is that both methods generate new instances by averaging existing ones. The difference is that in ASA the average is unweighted and can involve more than two examples. Furthermore, ASA does not require calculating the distance between the examples being averaged. This is a convenient aspect of ASA considering that tweets are represented by high-dimensional vectors. Another important difference relates to the type of data used for generating the instances. SMOTE combines labelled instances; ASA combines unlabelled instances annotated using an opinion lexicon.

4 The Lexical Polarity Hypothesis

In this section, we study the lexical polarity hypothesis on which ASA is based. It encapsulates the idea that a single opinion word in a tweet is a very strong indicator of the polarity of the message. The hypothesis is expressed in the following two statements:

1. A tweet containing at least one positive word is more likely to be positive than negative.
2. A tweet containing at least one negative word is more likely to be negative than positive.

We study this hypothesis empirically by estimating the probabilities of events corresponding to these statements using the *SemEval*⁷ corpus of hand-annotated positive and negative tweets and the AFINN lexicon. The *SemEval* [19] corpus contains 5232 positive tweets and 2067 negative tweets, annotated by human evaluators using the crowdsourcing platform Amazon Mechanical Turk⁸. Each tweet is annotated by five Mechanical Turk workers and the final label is determined based on the majority of the labels. We take a balanced sample of 2000 positive and 2000 negative tweets from this corpus to avoid bias caused by unevenly distributed tweets and focus the analysis on how the polarity of tweets is affected by the polarity of their words. Hence, we calculate the sets **posT** and **negT** from this corpus and study the polarity distribution of their messages.

We first study the distribution of **posT** and **negT** by unsetting the m flag. Hence, we include tweets with mixed positive and negative words in both sets. The set **posT** has 2419 tweets, which corresponds to 60% of the tweets, and has a distribution of 826 negative and 1593 positive tweets. Thus, the estimated probability of a tweet from **posT** of having a positive polarity is 0.66. The set **negT** contains 1774 tweets, corresponding to 44% of the tweets, and has a distribution of 1354 negative and 420 positive tweets. This gives an estimated probability of 0.76 that a tweet from **negT** is negative. These results suggest that negative words are stronger indicators than positive words for determining the polarity of a tweet.

We also study the distribution of **posT** and **negT** after discarding tweets with mixed positive and negative words (m turned on). In this case, the size of **posT** is reduced to 1552 (39% of the total) tweets with a distribution of 284 negative and 1268 positive tweets. This gives an estimated probability of 0.817 that a tweet from **posT** is positive. The size of **negT** is reduced to 907 tweets (23% of the total) with a distribution of 812 negative and 95 positive tweets. This gives an estimated probability of 0.9 that a tweet from **negT** is negative.

The polarity distribution of these sets is presented as bar charts in Figure 1. The figure shows how the distributions become more skewed when removing tweets with mixed positive and negative opinion words.

⁷ <http://www.cs.york.ac.uk/semEval-2013/task2/>

⁸ <http://www.mturk.com>

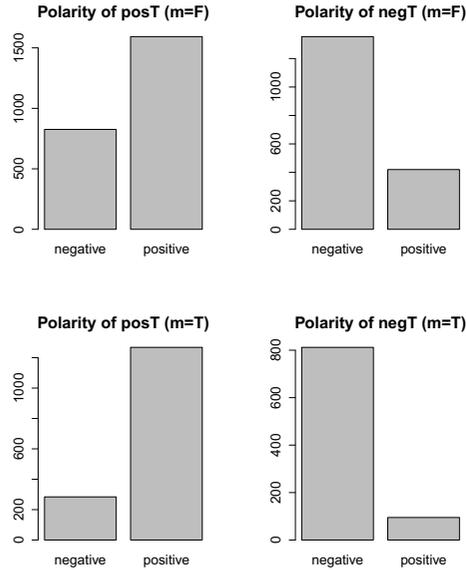


Figure 1. Polarity distributions of **posT** and **negT**.

We also study the distribution of tweets with mixed positive and negative words. We found 857 such tweets (21% of the total) with a distribution of 542 negative and 325 positive tweets. These numbers also indicate that negative opinion words have a greater effect than positive words on the polarity of the tweets in which they occur. However, negative words are also less frequent than positive ones.

The results obtained in this section support the lexical polarity hypothesis on which ASA is based. We can conclude that opinion words are indeed strong indicators of the polarity of tweets. We observed that discarding tweets with mixed opinion words produces a stronger effect. However, it is important to bear in mind that discarding these tweets may also cause loss of valuable information. The further effects of averaging multiple tweets containing opinion words with the same polarity are investigated in the following section.

5 Classification Experiments

In this section, we conduct an experimental evaluation of the proposed ASA algorithm. We evaluate the performance of classifiers trained with instances generated by ASA on different datasets of hand-annotated tweets. We compare these classifiers with classifiers trained on data obtained from two other distant supervision baselines:

1. Emoticon-annotation approach (EAA): labels tweets with positive or negative emoticons according to the emoticon’s polarity after removing the emoticon from the message. Tweets containing both positive and negative emoticons are discarded.
2. Lexicon-annotation approach (LAA): annotates tweets using the AFINN lexicon. Tweets with at least one positive word and no negative words are labelled positive, and analogously, tweets with at least one negative word and no positive words are labelled negative. The positive and negative instances from LAA are equivalent to the sets **posT** and **negT** from ASA when m is turned on.

The goal of comparing ASA against EAA and LAA is to determine which distant supervision model generates better training data for polarity classification from a source corpus of unlabelled tweets. Considering that positive signals such as positive emoticons or positive opinion words occur more frequently in tweets than their negative counterparts, we also study balanced versions of EAA and LAA. The balanced baselines are referred to as EAA.B and LAA.B, and are obtained by undersampling the majority class in each case.

The unlabelled tweets used as source corpora in all methods are taken from the Edinburgh corpus (ED) [21], which is a general purpose collection of 97 million unlabelled tweets in multiple languages collected with the Twitter streaming API between November 11th 2009 and February 1st 2010. Tweets written in languages different from English are discarded, resulting in a corpus of around 50 million English tweets.

The positive and negative emoticons used for labelling tweets with EAA are: “:)”, “:D”, “=D”, “=)”, “:]”, “=]”, “:-)”, “:-D”, “:-]”, “;:)”, “;D”, “;]”, “;-)”, “;-D”, and “;-]” for positive tweets and “:(”, “=(”, “;(”, “:[”, “=[”, “:-(”, “:-[”, “;:(”, “;[”, and “D:” for negative tweets.

The features used for representing the tweets in both approaches are: unigrams, POS tags, and Brown word clusters. The tokenisation and the POS tagging of the tweets is conducted using the **TweetNLP** library⁹. The word clusters are also taken from the **TweetNLP** project and correspond to 1000 different clusters trained from a collection of around 56 million tweets using the Brown-clustering algorithm.

With the aim of analysing the effect of averaging multiple tweets for building training instances, we study different values of the a parameter of ASA. We also study the effect of including or excluding tweets with mixed positive and negative words by comparing the performance of ASA with the flag m turned on and off, respectively. We create balanced training datasets with size equal to 1% of the size of the source corpus by setting parameters p and n to 0.5% of the source corpus size.

The evaluation of ASA, EAA, and LAA is carried out by comparing the performance obtained by an L_2 -regularised logistic regression trained with data generated from those two models, applied on three test collections of tweets that were manually assigned to positive and negative classes. These collections are: *6HumanCoded*¹⁰, *Sanders*¹¹, and *SemEval*¹², which was also used in Section 4. The *6HumanCoded* dataset is a collection of tweets scored according to positive and negative numerical scores by six human evaluators. The ratings are averaged and we use the difference of these scores to create polarity classes. We discard messages where this difference is zero. The *Sanders* dataset consists of 570 positive and 654 negative tweets evaluated by a single human annotator. The number of positive and negative tweets per dataset is given in Table 2.

	Positive	Negative	Total
6HumanCoded	1340	949	2289
Sanders	570	654	1224
SemEval	5232	2067	7299

Table 2. Manually-annotated collections of tweets.

⁹ <http://www.ark.cs.cmu.edu/TweetNLP/>

¹⁰ <http://sentistrength.wlv.ac.uk/documentation/6humanCodedDataSets.zip>

¹¹ <http://www.sananalytics.com/lab/twitter-sentiment/>

¹² <http://www.cs.york.ac.uk/semEval-2013/task2/>

The tweets from the target collections are mapped into the same feature-space as the tweets generated by the distant supervision models. The logistic regression model is taken from LIBLINEAR¹³, with the regularisation parameter C set to 1.0. Each distant supervision model is trained ten times on data generated from ten independent partitions of 2 million tweets from the source corpus. The average performance of each classifier trained with ASA is compared with the average performance of classifiers trained with each of the four distant supervision baselines 1) EAA, 2) EAA.B, 3) LAA, 4) LAA.B, using a paired Wilcoxon signed-rank test with the significance value set to 0.05.

Different distant supervision models produce different numbers of labelled instances from the same corpus of unlabelled tweets. The average number of positive and negative instances generated by each distant supervision model from the ten collections of 2 million unlabelled tweets is shown in Table 3.

	Avg. Positive	(%)	Avg. Negative	(%)	Avg. Total	(%)
EAA	130,641	(6.5%)	21,537	(1.1%)	152,179	(7.6%)
EAA.B	21,537	(1.1%)	21,537	(1.1%)	43,074	(2.2%)
LAA	681,531	(34.1%)	294,177	(14.7%)	975,708	(48.8%)
LAA.B	294,177	(14.7%)	294,177	(14.7%)	588,354	(29.4%)
ASA	10,000	(0.5%)	10,000	(0.5%)	20,000	(1%)

Table 3. Average number of positive and negative instances generated by different distant supervision models from 10 collections of 2 million tweets.

We use the macro-averaged F1 score and the weighted area under the ROC curves (AUCs) as evaluation measures for comparing classifiers. Macro-averaged F1 was used in the SemEval sentiment analysis in Twitter task¹⁴, and AUC is a useful metric for comparing the performance of classifiers because it is independent of any specific value for the decision threshold.

The comparisons are done for each target collection of tweets and the results for the macro-averaged F1 score and AUC are given in Table 4. The statistical significance tests of each configuration of ASA with respect to each of the four baselines are indicated by a sequence of four symbols. Improvements are denoted by a plus (+), degradations by a minus (-), and cases where no statistical significant difference is observed by an equals (=). The baselines are also compared amongst each other.

We observe that EAA performs substantially worse than the other baselines in F1 score. EAA.B performs substantially better than EAA. From Table 3 we observe that EAA is the model that produces the most uneven distribution of positive and negative instance. This suggest that the macro-average F1 score is very sensitive to classifiers trained from heavily imbalanced data. In contrast, we can note that balancing EAA does not cause any improvement in AUC. AUC is a more robust measure for classifiers trained from imbalanced datasets.

Regarding the LAA baseline, we observe a degradation in F1 after balancing the data (LAA.B). On the other hand, LAA.B performs almost identically to LAA in AUC. We believe that the reason why balancing is not causing a positive impact in the lexicon-based approach is that LAA produces a less skewed distribution of positive and negative instances than EAA. The benefits of resampling are more substantial for F1 for very skewed distributions such as those produced by EAA. There is no clear consensus about which baseline is the best. The baselines based on lexicons perform better than the ones based on emoticons in SemEval for both F1 and AUC. In Sanders, the

lexicon and the balanced emoticons behave similarly in F1, but the emoticons perform better for AUC. In 6HumanCoded EAA.B performs better than LAA and LAA.B in F1, but in AUC they produce almost identical results. It is worth mentioning that the emoticon-based approach can achieve competitive results to the lexicon-based one even though it generates substantially less training data (Table 3).

Regarding ASA, we observe that the performance achieved by our proposed method depends on the parameter setting. When the tweets with mixed positive and negative tweets are discarded ($m=T$) we observe that the best results are achieved when very few tweets are averaged. There is a strong decline in the performance of ASA ($m=T$) when the value of a is increased. We believe that this is because instances become too similar when formed by averaging too many tweets. ASA ($m=T$) with $a=1$ is essentially a subsampled version of LAA.B, and indeed produces very similar results. ASA ($m=T$) is not capable of producing statistically significant improvements over the four baselines for either AUC and F1 score for any dataset, even with its optimum value of a . This suggests that there is no clear contribution in the sample and average steps of ASA when tweets with mixed positive and negative tweets are discarded.

On the other hand, when tweets with mixed positive and negative words are simultaneously added to both sets ($m=F$), ASA produces statistically significant improvements over all the baselines in all target collections for both F1 and AUC, for appropriate values of a . The best value of a is ten in the three target collections, for both performance metrics. These results indicate that ASA, with calibrated parameters, outperforms existing distant supervision models for Twitter polarity classification. The fact that turning m off is better than discarding tweets with mixed positive and negative words suggests that mixed tweets contribute to better generalisation. This is because real positive and negative tweets are likely to contain words with both polarities.

We clearly observe that setting a to one in ASA ($m=F$) produces results that are far from the optimum. This validates the idea that averaging multiple tweets with at least one word with the same polarity increases the chance of producing an instance of the desired polarity. We observe again a decline in performance when the value of a is further increased.

Based on the numbers in Table 3 we are using 7.6 and 48.8 times more training data with EAA and LAA than with ASA respectively. It is noteworthy that ASA classifiers outperform the classifiers trained with EAA and LAA even though they are trained with less data. This essentially shows that ASA can produce a more compact and efficient training dataset than previous distant supervision models.

Examples of tweets from the SemEval corpus classified using ASA, with $a = 10$ and $m = F$, are given in Table 5. The positive and negative words from the AFINN lexicon are marked with blue and red colours respectively. The classification outputs reveal some insights about the strengths and shortcomings of our method. The correctly classified examples suggest that ASA is capable of learning sentiment expressions that go beyond the lexicon used in the annotation phase. This is observed in the second and third negative examples, and the last positive one, which are all correctly classified even though they do not contain AFINN words with the same polarity than the corresponding tweet. ASA learns opinion words co-occurring with the words from the lexicon, because all words from a tweet are considered in the feature space. This is an indirect form of polarity lexicon expansion. Regarding the misclassified examples, we observe that the current implementation of ASA is not capable of accurately handling complex sentiment patterns involving negations

¹³ <http://www.csie.ntu.edu.tw/~cjlin/liblinear/>

¹⁴ <http://alt.qcri.org/semeval2016/task4/>

Macro-averaged F1						
	6HumanCoded		Sanders		SemEval	
EAA.U	0.576 ± 0.007	= - - -	0.506 ± 0.018	= - - -	0.591 ± 0.018	= - - -
EAA.B	0.735 ± 0.008	+ = + +	0.709 ± 0.018	+ = = =	0.711 ± 0.006	+ = - =
LAA.U	0.729 ± 0.004	+ - = +	0.711 ± 0.003	+ = = +	0.725 ± 0.002	+ + = +
LAA.B	0.719 ± 0.002	+ - - =	0.703 ± 0.004	+ = - =	0.712 ± 0.002	+ = - =
ASA ($a = 1, m = T$)	0.734 ± 0.005	+ = + +	0.721 ± 0.010	+ + + +	0.724 ± 0.004	+ + = +
ASA ($a = 5, m = T$)	0.745 ± 0.005	+ + + +	0.723 ± 0.010	+ + + +	0.722 ± 0.006	+ + = +
ASA ($a = 10, m = T$)	0.737 ± 0.003	+ = + +	0.703 ± 0.011	+ = - =	0.708 ± 0.007	+ - - =
ASA ($a = 50, m = T$)	0.693 ± 0.003	+ - - -	0.643 ± 0.004	+ - - -	0.639 ± 0.006	+ - - -
ASA ($a = 100, m = T$)	0.672 ± 0.004	+ - - -	0.620 ± 0.005	+ - - -	0.607 ± 0.006	+ - - -
ASA ($a = 500, m = T$)	0.638 ± 0.004	+ - - -	0.599 ± 0.008	+ - - -	0.563 ± 0.005	- - - -
ASA ($a = 1000, m = T$)	0.635 ± 0.004	+ - - -	0.594 ± 0.010	+ - - -	0.554 ± 0.003	- - - -
ASA ($a = 1, m = F$)	0.717 ± 0.007	+ - - =	0.691 ± 0.013	+ - - -	0.699 ± 0.008	+ - - -
ASA ($a = 5, m = F$)	0.755 ± 0.004	+ + + +	0.730 ± 0.008	+ + + +	0.735 ± 0.005	+ + + +
ASA ($a = 10, m = F$)	0.761 ± 0.003	+ + + +	0.735 ± 0.015	+ + + +	0.742 ± 0.006	+ + + +
ASA ($a = 50, m = F$)	0.749 ± 0.004	+ + + +	0.673 ± 0.005	+ - - -	0.699 ± 0.009	+ - - -
ASA ($a = 100, m = F$)	0.717 ± 0.003	+ - - -	0.645 ± 0.006	+ - - -	0.664 ± 0.005	+ - - -
ASA ($a = 500, m = F$)	0.665 ± 0.002	+ - - -	0.621 ± 0.007	+ - - -	0.621 ± 0.004	+ - - -
ASA ($a = 1000, m = F$)	0.653 ± 0.003	+ - - -	0.619 ± 0.007	+ - - -	0.613 ± 0.002	+ - - -
AUC						
	6HumanCoded		Sanders		SemEval	
EAA.U	0.805 ± 0.005	= - - -	0.800 ± 0.017	= = + +	0.802 ± 0.006	= + - -
EAA.B	0.809 ± 0.001	= = = =	0.795 ± 0.016	= = + +	0.798 ± 0.007	- - - -
LAA.U	0.809 ± 0.001	+ = = =	0.778 ± 0.002	- - = =	0.814 ± 0.000	+ + = =
LAA.B	0.809 ± 0.001	+ = = =	0.778 ± 0.003	- - = =	0.813 ± 0.001	+ + = =
ASA ($a = 1, m = T$)	0.806 ± 0.003	= = - -	0.786 ± 0.007	- - + +	0.808 ± 0.002	+ + - -
ASA ($a = 5, m = T$)	0.809 ± 0.002	= = = =	0.787 ± 0.005	- = + +	0.810 ± 0.003	+ + - -
ASA ($a = 10, m = T$)	0.804 ± 0.001	- - - -	0.776 ± 0.008	- - = =	0.806 ± 0.003	+ + - -
ASA ($a = 50, m = T$)	0.756 ± 0.003	- - - -	0.697 ± 0.005	- - - -	0.763 ± 0.002	- - - -
ASA ($a = 100, m = T$)	0.729 ± 0.002	- - - -	0.672 ± 0.006	- - - -	0.739 ± 0.002	- - - -
ASA ($a = 500, m = T$)	0.696 ± 0.003	- - - -	0.642 ± 0.008	- - - -	0.707 ± 0.005	- - - -
ASA ($a = 1000, m = T$)	0.690 ± 0.004	- - - -	0.637 ± 0.008	- - - -	0.701 ± 0.006	- - - -
ASA ($a = 1, m = F$)	0.793 ± 0.005	- - - -	0.762 ± 0.016	- - - -	0.787 ± 0.007	- - - -
ASA ($a = 5, m = F$)	0.837 ± 0.004	+ + + +	0.807 ± 0.010	= = + +	0.833 ± 0.003	+ + + +
ASA ($a = 10, m = F$)	0.845 ± 0.001	+ + + +	0.812 ± 0.015	+ + + +	0.840 ± 0.003	+ + + +
ASA ($a = 50, m = F$)	0.815 ± 0.003	+ + + +	0.759 ± 0.006	- - - -	0.810 ± 0.004	+ + + +
ASA ($a = 100, m = F$)	0.781 ± 0.003	- - - -	0.720 ± 0.007	- - - -	0.779 ± 0.004	- - - -
ASA ($a = 500, m = F$)	0.723 ± 0.002	- - - -	0.670 ± 0.008	- - - -	0.729 ± 0.005	- - - -
ASA ($a = 1000, m = F$)	0.712 ± 0.002	- - - -	0.665 ± 0.007	- - - -	0.721 ± 0.005	- - - -

Table 4. Macro-averaged F1 and AUC measures for different distant supervision models. Best results per column for each measure are given in bold.

and but clauses. We attribute these problems to two factors: 1) the annotation phase is solely based on unigrams, and 2) the current feature space omits the order in which words occur. The first factor could be addressed by using a lexicon of sentiment annotated phrases, and the second one by using more sophisticated feature representations such as n-grams or paragraph vector-embeddings [11].

We also study the effect of increasing the source corpus size in all different distant supervision methods: EAA, EAA.B, LAA, LAA.B, and ASA. It is important to remark that the number of generated instances in the four distant supervision baselines increases when increasing the size of the source corpus. The increments are proportional to the percentages shown in Table 3.

We trained classifiers using partitions of the source corpus ranging from ten thousand to ten million tweets. For the ASA model we set a to 10 and m to false, which were the best parameters according to the previous experiments (Table 4), and kept p and n with values set to $0.005 \times |\mathcal{C}|$, for generating balanced datasets with size equal to 1% of the size of the source corpus. Thus, the number of generated instances in ASA is also increased when using a larger source corpus.

The learning curves produced by logistic regressions applied to the SemEval dataset, trained with data generated using ASA and the four baselines from source corpora of different sizes, are shown in Figure 2. The performance metrics are again the macro-averaged F1 measure and AUC.

The figure indicates that most methods increase their performance when increasing the corpus size, and that these improvements tend to plateau when using more than 2 million tweets as input. We observe again that EAA exhibits poor performance in F1 and that balancing this method (EAA.B) produces substantial improvements for this measure. Surprisingly, the lexicon-based baselines LAA and LAA.B exhibit a slight decrease in F1 when increasing the source corpus size after the million tweet mark.

We observe in the initial part of the curves that LAA and LAA.B are the best distant supervision methods for source corpora smaller than 1 million tweets. This suggests that the prior knowledge from the lexicon can be very useful with small collections of data. It is important to consider that the setup of ASA for this experiment generates very few examples when the source corpus is small. This can be easily changed by generating more training data when the source

	Negative Tweets	Positive Tweets
f(x)=neg	Can we just haw class cancelled tomorrow? Cause I really don't want to go to BCA 101. I'd rather eat worms....	Never start working on your dreams and goals tomorrow... tomorrow never comes... if it means anything to U, ACT NOW! #getafterit
	I never had a good time, I sat by my bedside. With papers and poetry about Estella	Just did Spartacus 2.0 and sauna imma be sore tomorrow but so worth it
	I got tickets to the NC State game saturday and nobody to go with..	@patrishuhx7 I have English tomorrow but it honestly doesn't bother me for some reason. Rella always makes my day. Don't ask
f(x)=pos	Wish me lucky on the Cahsee tomorrow I'm pretty nervous	Happy Valentine's Day!!! @MAziing: Everyday is the 14th!
	I haven't talked to you since July 19 th and all you can say is So do you like	Ground hog day is such a good film, Sunday is for food and films #sunday
	Beyonce's new cd GTFO	
	Being in Amsterdam this early on a friday morning is not my ideal, I just want to get home!	Going to see Kendrick Lamar with @Pea_Starks in jan :D

Table 5. Examples of tweets classified with ASA. Positive and negative words from AFINN are marked with blue and red colours respectively. The leftmost column indicates the classifier's output.

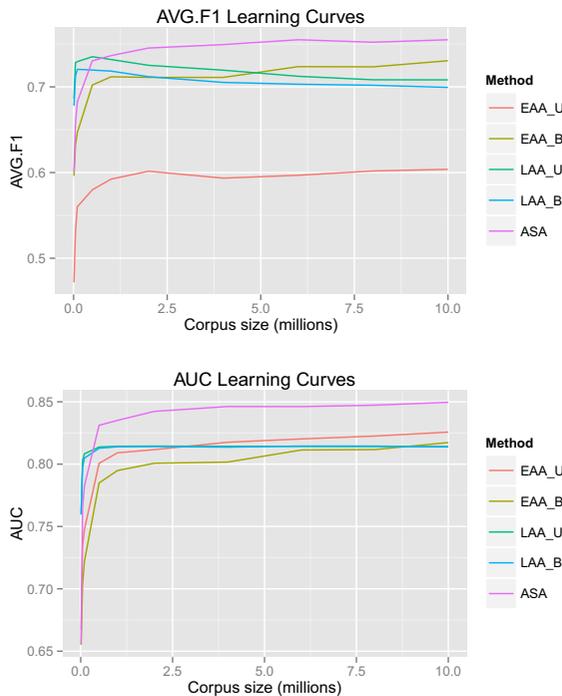


Figure 2. Learning curves over the SemEval dataset.

corpus is too small.

We also observe that after passing the million tweet mark, the emoticon-based models are better than LAA and LAA_B, and that ASA outperforms all the other models. These results indicate that ASA is a powerful distant supervision model that can be used for training accurate message-level polarity classifiers without relying on very large collections of unlabelled data.

6 Conclusions

We propose a new model called ASA to generate synthetic training data for Twitter sentiment analysis from unlabelled corpora using the prior knowledge provided by an opinion lexicon¹⁵. The method annotates tweets according to the polarity of their words, using a given polarity lexicon, and generates balanced training data by sampling

and averaging tweets containing words with the same polarity. ASA is based on the lexical polarity hypothesis: because tweets are short messages, opinion words are strong indicators of the sentiment of the tweets in which they occur, and therefore tweets with at least one word with a certain known prior polarity are more likely to express the same polarity on the message level than the opposite one. The sample and average steps of ASA exploit this hypothesis by increasing the confidence of generating an instance located in the desired polarity region. ASA also incorporates a novel way for incorporating the knowledge provided by tweets with mixed positive and negative words.

The experimental results show that ASA produces better classifiers than the widely-adopted approach of using emoticons for labelling tweets into polarity classes and also better results than labelling tweets based on the polarity of their words, without sampling and averaging. Moreover, classifiers trained with data generated by ASA achieve better results than the other distant supervision models using substantially less training data. This shows that ASA can generate compact and efficient dataset for learning polarity concepts.

The proposed model can be used for training Twitter polarity classifiers in scenarios without labelled training data and for creating domain-specific sentiment classifiers by collecting data from the target domain. Considering that opinion lexicons are usually easier to obtain than corpora of polarity-annotated tweets, ASA can save significant labelling efforts for learning polarity classifiers in Twitter.

ASA opens several directions for further research. In essence, ASA allows the transfer of sentiment labels from the word-level to the message-level. Therefore, it could potentially be used for classifying tweets according to other sentiment labels associated with words, such as subjectivity labels, numerical scores indicating sentiment strength, and multi-label emotions.

Considering that ASA can generate large amounts of training data from large source corpora, it could also be suitable for training deep neural networks that learn more sophisticated representations of tweets for sentiment classification.

Another important aspect of ASA is its flexibility: it can be used with any kind of features for representing the tweets. For example, paragraph vector-embeddings [11], which have shown to be powerful representations for sentences, could be trained from large corpora of unlabelled tweets and included in the feature space.

Finally, ASA could also be adapted for training incremental polarity classifiers in an on-line fashion from a stream of time-evolving tweets. This approach could be used for online opinion mining from social media streams [2], and potentially be useful for tracking public opinion regarding high-impact events on Twitter, such as political campaigns, movie releases and natural disasters.

¹⁵ The source code of the model is available for download at <http://www.cs.waikato.ac.nz/ml/sa/ds.html#asa>.

REFERENCES

- [1] Finn Årup Nielsen, 'A new ANEW: Evaluation of a word list for sentiment analysis in microblogs', in *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, #MSM2011, pp. 93–98, (2011).
- [2] Albert Bifet and Eibe Frank, 'Sentiment knowledge discovery in twitter streaming data', in *Proceedings of the 13th International Conference on Discovery science*, DS'10, pp. 1–15, Berlin, Heidelberg, (2010). Springer-Verlag.
- [3] Felipe Bravo-Marquez, Marcelo Mendoza, and Barbara Poblete, 'Meta-level sentiment models for big social data analysis', *Knowledge-Based Systems*, **69**(0), 86–99, (2014).
- [4] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai, 'Class-based n-gram models of natural language', *Computational Linguistics*, **18**(4), 467–479, (1992).
- [5] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer, 'Smote: Synthetic minority over-sampling technique', *J. Artif. Int. Res.*, **16**(1), 321–357, (June 2002).
- [6] Alec Go, Richa Bhayani, and Lei Huang, 'Twitter sentiment classification using distant supervision', *CS224N Project Report*, Stanford, (2009).
- [7] Nathalie Japkowicz and Shaju Stephen, 'The class imbalance problem: A systematic study', *Intell. Data Anal.*, **6**(5), 429–449, (October 2002).
- [8] Soo-Min Kim and Eduard Hovy, 'Determining the sentiment of opinions', in *Proceedings of the 20th International Conference on Computational Linguistics*, pp. 1367–1373, Stroudsburg, PA, USA, (2004). Association for Computational Linguistics.
- [9] Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore, 'Twitter sentiment analysis: The good the bad and the omg!', *ICWSM*, **11**, 538–541, (2011).
- [10] Krishna K. Ladha, 'Condorcet's jury theorem in light of de Finetti's theorem', *Social Choice and Welfare*, **10**(1), 69–85, (1993).
- [11] Quoc V Le and Tomas Mikolov, 'Distributed representations of sentences and documents', in *Proceedings of the 31th International Conference on Machine Learning*, pp. 1188–1196, (2014).
- [12] Shoushan Li, Zhongqing Wang, Guodong Zhou, and Sophia Yat Mei Lee, 'Semi-supervised learning for imbalanced sentiment classification', in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, IJCAI'11, pp. 1826–1831. AAAI Press, (2011).
- [13] Tao Li, Yi Zhang, and Vikas Sindhwani, 'A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge', in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL '09, pp. 244–252, Stroudsburg, PA, USA, (2009). Association for Computational Linguistics.
- [14] Kun-Lin Liu, Wu-Jun Li, and Minyi Guo, 'Emoticon smoothed language models for twitter sentiment analysis', in *Proceedings of the National Conference on Artificial Intelligence*, pp. 1678–1684, (2012).
- [15] Prem Melville, Wojciech Gryc, and Richard D. Lawrence, 'Sentiment analysis of blogs by combining lexical knowledge with text classification', in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1275–1284, New York, NY, USA, (2009). ACM.
- [16] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky, 'Distant supervision for relation extraction without labeled data', in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL '09, pp. 1003–1011, Stroudsburg, PA, USA, (2009). Association for Computational Linguistics.
- [17] Saif Mohammad and Peter D. Turney, 'Crowdsourcing a word-emotion association lexicon.', *Computational Intelligence*, **29**(3), 436–465, (2013).
- [18] Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu, 'NRC-canada: Building the state-of-the-art in sentiment analysis of tweets', in *Proceedings of the Seventh International Workshop on Semantic Evaluation Exercises*, SemEval'13, pp. 321–327, (2013).
- [19] Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson, 'Semeval-2013 task 2: Sentiment analysis in twitter', in *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Volume 2: *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 312–320, Atlanta, Georgia, USA, (June 2013). Association for Computational Linguistics.
- [20] Alexander Pak and Patrick Paroubek, 'Twitter as a corpus for sentiment analysis and opinion mining', in *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pp. 1320–1326, Valletta, Malta, (2010).
- [21] Saša Petrović, Miles Osborne, and Victor Lavrenko, 'The edinburgh twitter corpus', in *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, WSA '10, pp. 25–26, Stroudsburg, PA, USA, (2010). Association for Computational Linguistics.
- [22] Jonathon Read, 'Using emoticons to reduce dependency in machine learning techniques for sentiment classification', in *Proceedings of the ACL Student Research Workshop*, ACLstudent '05, pp. 43–48, Stroudsburg, PA, USA, (2005). Association for Computational Linguistics.
- [23] Aliaksei Severyn and Alessandro Moschitti, 'Twitter sentiment analysis with deep convolutional neural networks', in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 959–962, New York, NY, USA, (2015). ACM.
- [24] Vikas Sindhwani and Prem Melville, 'Document-word co-regularization for semi-supervised sentiment analysis', in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pp. 1025–1030, Washington, DC, USA, (2008). IEEE Computer Society.
- [25] Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge, 'Twitter polarity classification with label propagation over lexical links and the follower graph', in *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pp. 53–63, Stroudsburg, PA, USA, (2011). Association for Computational Linguistics.
- [26] Duyu Tang, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou, 'Coooolll: A deep learning system for twitter sentiment classification', in *Proceedings of the 8th International Workshop on Semantic Evaluation*, pp. 208–212, Dublin, Ireland, (August 2014). Association for Computational Linguistics and Dublin City University.
- [27] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou, 'Sentiment strength detection for the social web.', *JASIST*, **63**(1), 163–173, (2012).
- [28] Peter D. Turney, 'Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews', in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 417–424, Stroudsburg, PA, USA, (2002). Association for Computational Linguistics.
- [29] Lei Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu, 'Combining lexicon-based and learning-based methods for twitter sentiment analysis', Technical report, Hewlett-Packard Development Company, L.P., (2011).

Semi-Supervised Group Sparse Representation: Model, Algorithm and Applications

Longwen Gao¹ and Yeqing Li² and Junzhou Huang³ and Shuigeng Zhou⁴

Abstract. Group sparse representation (GSR) exploits group structure in data and works well on many problems. However, the group structure must be manually given in advance. In many practical scenarios such as classification, samples are grouped according to their labels. Constructing a consistent group structure in such cases is not easy. The reasons are: 1) samples may be incorrectly labeled; and 2) label assigning in big data is time-consuming and expensive. In this paper, we propose and formulate a new problem, *semi-supervised group sparse representation* (SS-GSR) to support group sparse representation among both labeled and unlabeled data, while learning a more robust group structure, which can be further exploited to more effectively represent other unlabeled data. We develop a model to tackle the SS-GSR problem, based on the manifold assumption in subspace segmentation that samples in the same group lie close in feature space and span the same subspace. We also propose an alternating algorithm to solve the model. Finally, we validate the model via extensive experiments.

1 INTRODUCTION

Sparse representation (SR) [18] and group sparse representation (GSR) [24] have been successfully applied to many regression problems [24] and machine learning tasks, such as the classification tasks of images [15, 22], texts [20, 9] and biological data [13, 23]. GSR considers the group structure of data as prior knowledge and benefits from it when the data is consistent with such structure. For example, in most classification tasks, samples can be seen naturally with a group structure, because samples in the same class tend to be grouped together. For such cases, GSR usually outperforms SR [15] because group sparsity works better when the underlying samples are strongly group-sparse [10]. However, GSR requires that the group structure is explicitly given in advance, which is implied in the class relationship of labeled samples. In real applications, accurate label information may not be easy to acquire. On the one hand, the samples may be incorrectly labeled. On the other hand, it requires a lot human effort to assign the labels, which is prohibitively expensive for big data. Consequently, a large fraction of data in reality are unlabeled although we know that they should have certain labels. In parallel to

semi-supervised learning, if we can exploit the large amounts of unlabeled data in the GSR process, better data representation should be achieved.

With this in mind, in this paper we propose a new problem, *semi-supervised group sparse representation* (SS-GSR) to conduct GSR over a dataset consisting of both labeled and unlabeled samples. The aim is two-fold: 1) representing each sample with respect to the other samples while the representation coefficients are consistent with the underlying group structure of the whole dataset; 2) learning a more robust group structure underlying the dataset via exploiting also the unlabeled samples. SS-GSR is not only a nontrivial advancement but also a significant complement to the traditional GSR that represents unlabeled samples with a dictionary of labeled samples by imposing the group sparsity constraint. SS-GSR performs GSR among labeled and unlabeled data, meanwhile refines the group structure explicitly given in the labeled data by additionally utilizing unlabeled data.

To reveal the underlying group structure of the dataset, we believe that the coefficient matrix should be in a specific form. Manifold assumption and block-diagonal constraint are introduced in subspace segmentation [21] to cluster samples into groups. Samples (labeled and unlabeled) are assumed to be grouped according to their underlying subspace and the distance in the feature space. This assumption allows the block-diagonal constraint on the affinity matrix to find clustering structure among samples [8]. In this paper, we employ the same assumption to the SS-GSR problem and formulate our model with block-diagonal constraint, thus the underlying group structure can be discovered with the block structure in the coefficient matrix. Furthermore, to exploit the group structure of unlabeled data in sparse representation, we construct the affinity matrix using the coefficient matrix and try to maintain the local consistency of group structure among samples according to the affinity matrix as in [27].

Contributions of this paper are as follows:

- We propose the problem of SS-GSR to extend GSR so that unlabeled data can be also exploited in the representation process.
- We formulate our model to automatically learn the underlying group structure by utilizing the manifold structure of data, and develop an efficient algorithm to solve the model.
- We validate our model by extensive experiments of two typical applications. Experimental results show that our model outperforms the existing GSR model and three semi-supervised learning methods (including one proposed recently).

The rest of this paper is organized as follows: Section 2 reviews the traditional group sparse representation (GSR) model, which is the starting point and the most related work to our model proposed in this paper. Section 3.2 presents the new model that is called semi-supervised group sparse representation (SS-GSR). Section 4 intro-

¹ Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, Shanghai 200433, China. Email: lw-gao@fudan.edu.cn

² University of Texas at Arlington, Arlington, Texas, USA, 76019. Email: yeqing.li@mavs.uta.edu

³ University of Texas at Arlington, Arlington, Texas, USA, 76019. Email: jzhuang@uta.edu

⁴ Correspondence author. Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, Shanghai 200433, China. Email: sgzhou@fudan.edu.cn

duces the algorithm to solve the proposed model. Section 5 gives the validation of the proposed model on two applications. Section 6 concludes this paper.

2 GROUP SPARSE REPRESENTATION (GSR)

GSR explores group structure information during representation by requiring the coefficients corresponding to different groups to be sparse. The training samples used to represent other samples together constitute a dictionary $\mathbf{X} \in \mathcal{R}^{d \times n}$. Let \mathcal{G}_g be the group of indices of training samples with group id $g \in \{1..c\}$, given another test sample $\mathbf{y} \in \mathcal{R}^d$, GSR can be formulated as:

$$\min_{\mathbf{z} \in \mathcal{R}^n} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\mathbf{z}\|_2^2 + \lambda \sum_{g=1}^c \|z_{\mathcal{G}_g}\|_2, \quad (1)$$

with tradeoff parameter $\lambda > 0$. Here, the non-zero elements of vector $z_{\mathcal{G}_g}$ are the same as those of vector \mathbf{z} indexed in \mathcal{G}_g . The first term is the regression error, and the second term can be seen as an ℓ_{21} -norm: the ℓ_2 -norm is for the elements of the coefficient vector \mathbf{z} inside each group and ℓ_1 -norm measures the sparsity among groups.

Given the group structure, GSR favors the selection of multiple correlated samples in the dictionary to represent the test sample, and thus promotes the representation of the test sample in terms of all the training samples from the correct group [15]. Even though, it requires that the group structure of the whole dictionary should be given in advance and the structure should be correct according to the prior knowledge. However, in practice the given structure might not be fully consistent with data due to the complexity of data and noise in data collection. Assigning structures to all samples in the dictionary via human labor might be prohibitively expensive if not impossible when large amount of data are collected. In this paper, we will try to exploit the group structure information of some samples properly and learn the group structure of all samples automatically.

3 SS-GSR MODEL

3.1 Problem statement

Suppose we have a dataset $\mathbf{X} \in \mathcal{R}^{d \times n}$ whose column vector \mathbf{X}^i ($i = 1, \dots, n$) corresponds to each of the n samples. These samples can be grouped into c non-overlapping groups. However only part of these samples are given with group labels in $\mathcal{C} = \{1, \dots, c\}$, and for the rest of them, the group labels are unknown. We simply assume that the first m samples $\mathbf{X}^{1..m}$ are given with group labels, and these group labels form a group label vector $\mathbf{G} \in \mathcal{C}^m$. Our problem is to decide a coefficient matrix $\mathbf{Z} \in \mathcal{R}^{n \times n}$, whose columns are representation coefficients \mathbf{Z}^i that represent sample \mathbf{X}^i using the others, and the non-zero elements in \mathbf{Z}^i should correspond to samples in the same underlying group with sample \mathbf{X}^i . That is, the group sparsity on \mathbf{Z}^i should be the underlying group sparsity. Since we do not want samples to represent themselves, we fix the diagonal elements in \mathbf{Z} to be 0 to avoid such trivial representations. Accordingly, we have the following equation:

$$\mathbf{X} = \mathbf{X}\mathbf{Z}, \text{ s.t. } \mathbf{Z}_i^i = 0, \forall i \in \{1, \dots, n\}. \quad (2)$$

If we rearrange the samples to an order that the samples in the same underlying group are put together, the desired coefficient matrix \mathbf{Z} would be a block-diagonal matrix with each block corresponding to a group structure. This gives us the inspiration that we might be possible to find the underlying group structure by finding the block structure in the coefficient matrix \mathbf{Z} .

However, when we assume the given group structure of $\mathbf{X}^{1..m}$ to be unreliable, we need some other assumptions on data that can help find the block structure in \mathbf{Z} . Interestingly, the works of subspace segmentation [21, 7, 14, 8] follow similar idea to build a block-diagonal affinity matrix $\mathbf{W} \in \mathcal{R}_+^{n \times n}$. Subspace segmentation is to segment the samples according to the manifold assumption. The work of [8] explicitly imposes a fixed rank constraint on the graph Laplacian, which constrains the number of connected components in the affinity matrix \mathbf{W} as:

$$\text{rank}(\mathbf{L}\mathbf{W}) = n - c, \quad (3)$$

where $\mathbf{L}\mathbf{W}$ is the Laplacian matrix for \mathbf{W} and c is the number of connected components (a connected component corresponds to a group of samples). Thus the optimal affinity matrix is constrained to be a c -block-diagonal matrix.

Here, we employ the manifold assumption and the block constraint into GSR, with which the underlying group structure can be obtained by finding the block structure of the coefficient matrix. Note that though we used a similar block constraint form to that in [8], our work is different from subspace segmentation [8] at least in two aspects: a) The task is different. Their work aims at solving an unsupervised learning problem, while ours aims at extending the traditional group sparse representation, and applied it to both supervised and semi-supervised learning problems. b) The solution is different. Their work follows a two-step scheme: first they compute an affinity matrix \mathbf{W} from data, and then perform regular clustering on the affinity matrix. And the first step is independent from the second step. However in our work, the second step (classification) also affects the first step (computing the affinity matrix). Thus, we propose a combined scheme that solves the affinity matrix and the label assignment jointly, in order to simultaneously obtain a better affinity matrix and a more accurate label assignment.

3.2 Model formulation

We introduce a confidence matrix $\mathbf{F} \in \mathcal{R}_+^{n \times c}$ whose elements indicate the probability that a sample belongs to a certain group. So we have the following equation and inequation:

$$\begin{aligned} \sum_{j=1}^c F_i^j &= 1, \forall i \in \{1, \dots, n\}, \\ 0 &\leq F_i^j \leq 1, \forall i \in \{1, \dots, n\}, j \in \{1, \dots, c\}. \end{aligned}$$

As the first m samples' labels are already known, thus:

$$\begin{aligned} F_i^{G_i} &= 1, \forall i \in \{1, \dots, m\}; \\ F_i^j &= 0, \forall j \neq G_i, j \in \{1, \dots, c\}, i \in \{1, \dots, m\}. \end{aligned}$$

The first m rows of \mathbf{F} are fixed and we write them as \mathbf{F}_L . The rest part of \mathbf{F} is denoted by \mathbf{F}_U . Thus $\mathbf{F} = [\mathbf{F}_L^T \mathbf{F}_U^T]^T$.

We take two steps to formulate our model according to the two problems in GSR: first we try to detect the underlying group structure in samples whose group structure is given, then we present the procedure of finding the hidden group structure of the whole set of samples by taking also the unlabeled samples into consideration.

3.2.1 Detecting the underlying structure of labeled data

First, we focus on detecting the underlying structure in labeled data. To avoid the influence of unlabeled data, we fix the coefficients corresponding to those samples as 0, namely $\mathbf{Z}_{m+1, \dots, n}^{m+1, \dots, n} = \mathbf{0}$, and denote

the Laplacian from labeled data to labeled data as $\widehat{\mathbf{L}}_{\mathbf{W}} = \mathbf{L}_{\mathbf{W}_{1,\dots,m}^{1,\dots,m}}$, where $\mathbf{L}_{\mathbf{W}}$ is the Laplacian matrix of the affinity matrix \mathbf{W} .

Based on the requirement in Equation (2) and our assumption in Section 3.1, as well as the group sparsity regularization term, we formulate our model as:

$$\min_{\mathbf{Z}, \mathbf{F}} \frac{1}{2} \|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_{\mathbf{F}}^2 + \lambda \sum_{i=1}^m \|\mathbf{Z}^i\|_{G(\mathbf{F})} + \gamma \text{tr}(\mathbf{F}_L^{\top} \widehat{\mathbf{L}}_{\mathbf{W}} \mathbf{F}_L)$$

$$s.t. \text{rank}(\widehat{\mathbf{L}}_{\mathbf{W}}) = m - c, \mathbf{W} = (|\mathbf{Z}| + |\mathbf{Z}|^{\top})/2. \quad (4)$$

Above, the norm $\|\cdot\|_{G(\mathbf{F})}$ is a group sparse norm in which the group structure is given by the column number of the maximum element of each row in \mathbf{F} , namely, samples with the same column number are considered to be in the same group.

The first term in the objective function is to ensure the representation to be of small residual error to meet the requirement in Equation (2). The second term uses the confidence matrix to help decide the group structure of the coefficient matrix, and the third term uses the Laplacian matrix constructed by the coefficient matrix as the affinity matrix and perform label propagation on Laplacian graph. The rank constraint is equivalent to a block-diagonal constraint on \mathbf{W} [8], which encourages the samples to be clustered into groups according to the manifold assumption.

3.2.2 Finding the group structure for the entire dataset

Next, we go to find the hidden group structure of the whole set of samples and use the learned structure information for group sparse representation. Since we only have the group structure of part of the samples, we have to propagate the group structure with respect to the Laplacian $\mathbf{L}_{\mathbf{W}}$ as in the works of graph-based semi-supervised learning (SSL) [4, 27, 26]. Since in our first step, we have already learned a coefficient matrix and then an affinity matrix can be constructed as $\mathbf{W} = (|\mathbf{Z}| + |\mathbf{Z}|^{\top})/2$, the same propagation process as in graph-based SSL can be directly applied to our first model (4). Therefore, we have:

$$\min_{\mathbf{Z}, \mathbf{F}} \frac{1}{2} \|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_{\mathbf{F}}^2 + \lambda \sum_{i=1}^n \|\mathbf{Z}^i\|_{G(\mathbf{F})} + \gamma \text{tr}(\mathbf{F}^{\top} \mathbf{L}_{\mathbf{W}} \mathbf{F})$$

$$s.t. \text{rank}(\mathbf{L}_{\mathbf{W}}) = n - c, \mathbf{W} = (|\mathbf{Z}| + |\mathbf{Z}|^{\top})/2. \quad (5)$$

In the above formulation, the coefficient matrix \mathbf{Z} is learned for all samples, and the rank constraint is performed on the whole Laplacian graph.

Although we have formulated the model, solving the optimization problem is not easy because it is a non-smooth and non-convex problem, and the rank constraint is generally NP-hard. In Section 4, we will present an algorithm to efficiently solve the problem.

3.3 The Advantages of SS-GSR

In our model formulation (5), the affinity matrix \mathbf{W} can be divided into four parts:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{LL} & \mathbf{W}_{LU} \\ \mathbf{W}_{UL} & \mathbf{W}_{UU} \end{bmatrix}, \quad (6)$$

where $\mathbf{W}_{UL} = \mathbf{W}_{LU}^{\top}$ indicates the relationship between labeled samples and unlabeled samples, and \mathbf{W}_{LL} and \mathbf{W}_{UU} indicate respectively the relationships among labeled samples and unlabeled samples. In the best case, all the four matrices are block-diagonal matrices as shown in Figure 1.

Algorithm 1 Iteration between \mathbf{Z} and \mathbf{F}

Input: dictionary \mathbf{X} , initial labels \mathbf{F}_L
Initialize \mathbf{W}
repeat
 Solve \mathbf{F}_U using Equation (9)
 Solve \mathbf{Z} using Algorithm 2
 Update $\mathbf{W} = (|\mathbf{Z}| + |\mathbf{Z}|^{\top})/2$
until \mathbf{Z} and \mathbf{F} converge

Algorithm 2 Projected subgradient descent

Input: dictionary \mathbf{X} , labels \mathbf{F} , initial \mathbf{Z}_{init}
Initialize step size η
repeat
 Calculate subgradient \mathbf{g} of the objective in Equation (7)
 Subgradient descent $\mathbf{Z} = \mathbf{Z} - \eta\mathbf{g}$
 Project $\mathbf{Z} = \Pi_{\mathcal{K}}(\mathbf{Z})$ as in Algorithm 3
until \mathbf{Z} converges

We discuss the advantages of our model from two aspects: 1) learning the underlying group structure that is consistent with the labeled samples; 2) finding the underlying group structure by exploiting both labeled and unlabeled samples, and using the structure to represent all samples.

For the first aspect, how a sample is consistent with its group structure can be measured by the group sparsity of the corresponding column of \mathbf{W}_{LL} . Since our model learns the underlying group structure automatically, the properly learned \mathbf{W}_{LL} will show us how samples are consistent with their group structures and therefore improves the group sparse representation. We take the supervised classification task as an example, where unlabeled samples are classified one by one. For our model, only one sample is unlabeled (the one to be classified), and the matrix \mathbf{W}_{UU} becomes a single real number which is set to 0 since it is also the diagonal element, that is, we solve the model (4). In this case, though \mathbf{W}_{UU} will not help in classifying the unlabeled samples, our model can still outperforms GSR via learning \mathbf{W}_{LL} . Obviously, GSR can be seen as a special case of our model when the matrix \mathbf{W}_{LL} is fixed as a zero matrix.

For the second aspect, we compare our model with graph-based *semi-supervised learning* (SSL) methods because they all use the group information of unlabeled samples. The major difference is that, for the graph-based SSL methods, the affinity matrix \mathbf{W} must be constructed in advance and is fixed during the learning process. However, in our model the affinity matrix is constructed by coefficient matrix \mathbf{Z} , which is learned during the model optimization. Furthermore, \mathbf{W} in our model contains the underlying structure of data, while a pre-given \mathbf{W} in graph-based SSL methods may not be consistent with the structure of data [25]. In Section 5, our experiments over five real datasets clearly show that our model outperforms the graph-based SSL methods even when similar initialized \mathbf{W} is used.

4 SS-GSR ALGORITHM

In this section, we first design an alternating algorithm to solve the proposed model, and then briefly discuss the convergence of the algorithm.

4.1 Alternatively solving \mathbf{Z} and \mathbf{F}

Note that the rank constraint is all about the coefficient matrix \mathbf{Z} , so we first alternate between solving \mathbf{Z} and solving \mathbf{F} as outlined in

Algorithm 3 Projecting \mathbf{Z} into \mathcal{K}

Input: \mathbf{Z}_0 and c
Initialize $\mathbf{Z} = \mathbf{Z}_0$; $\rho = 1.1$; $\beta = 1 \times 10^{-4}$
repeat
Solve the first quadratic problem in (13) for \mathbf{Z}
Calculate $\tilde{\mathbf{L}}$ via Equation (14)
Update $\mathbf{J} = \mathbf{J} + \beta(\tilde{\mathbf{L}} - \mathbf{L}_W)$
Update $\beta = \rho\beta$
until \mathbf{Z} and $\tilde{\mathbf{L}}$ converge

Algorithm 1.

When \mathbf{F} is fixed, the optimization problem becomes:

$$\min_{\mathbf{Z}} \frac{1}{2} \|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_F^2 + \lambda \sum_{i=1}^n \|\mathbf{Z}^i\|_{G(\mathbf{F})} \quad (7)$$

s.t. $\text{rank}(\mathbf{L}_W) = n - c.$

We will further show how to solve this in Algorithm 2.

When \mathbf{Z} is fixed, the optimization problem becomes:

$$\min_{\mathbf{F}} \text{tr}(\mathbf{F}^\top \mathbf{L}_W \mathbf{F}), \quad (8)$$

and this unconstrained problem has a closed form solution [27]:

$$\mathbf{F}_U = (\mathbf{D}_{UU} - \mathbf{W}_U \mathbf{U})^{-1} \mathbf{W}_{UL} \mathbf{F}_L, \quad (9)$$

where \mathbf{D} is a diagonal matrix whose diagonal elements are the sums of every row of \mathbf{W} and \mathbf{D}_{UU} is that of \mathbf{W}_{UU} .

4.2 Sub-gradient descent for \mathbf{Z}

The optimization problem (7) is a non-smooth and constrained problem, which can be solved by the projected subgradient descent method. Let \mathcal{K} be the set of c -block-diagonal matrix as:

$$\mathcal{K} = \{\mathbf{Z} | \text{rank}(\mathbf{L}_W) = n - c\}. \quad (10)$$

Thus the rank constraint can be rewritten as $\mathbf{Z} \in \mathcal{K}$. For each iteration, we perform a subgradient descent on \mathbf{Z} and then project \mathbf{Z} back into the feasible set \mathcal{K} . This process is shown in Algorithm 2.

4.3 Projecting \mathbf{Z} into \mathcal{K}

To distinguish between input variable and output variable, we assume that the variable to be projected is \mathbf{Z}_0 . The projection step is to find a matrix in the set \mathcal{K} , which is closest to \mathbf{Z}_0 as follows:

$$\min_{\mathbf{Z}} \frac{1}{2} \|\mathbf{Z} - \mathbf{Z}_0\|_F^2, \quad \text{s.t. } \mathbf{Z} \in \mathcal{K}. \quad (11)$$

We introduce an auxiliary variable $\tilde{\mathbf{L}}$ to replace the Laplacian matrix \mathbf{L}_W and rewrite the projection (11) via Augmented Lagrangian Multiplier as in [1]:

$$\min_{\mathbf{Z}, \tilde{\mathbf{L}}} \frac{1}{2} \|\mathbf{Z} - \mathbf{Z}_0\|_F^2 + \langle \mathbf{J}, \tilde{\mathbf{L}} - \mathbf{L}_W \rangle + \frac{\beta}{2} \|\tilde{\mathbf{L}} - \mathbf{L}_W\|_F^2, \quad (12)$$

s.t. $\text{rank}(\tilde{\mathbf{L}}) = n - c,$

where \mathbf{J} is the Lagrangian multiplier and β is an increasing weight parameter. This problem can be solved by alternatively updating \mathbf{Z} ,

$\tilde{\mathbf{L}}$ and \mathbf{J} as follows:

$$\begin{aligned} \mathbf{Z} &= \arg\min_{\mathbf{Z}} \frac{1}{2} \|\mathbf{Z} - \mathbf{Z}_0\|_F^2 - \langle \mathbf{J}, \mathbf{L}_W \rangle + \frac{\beta}{2} \|\tilde{\mathbf{L}} - \mathbf{L}_W\|_F^2; \\ \tilde{\mathbf{L}} &= \arg\min_{\tilde{\mathbf{L}}} \langle \mathbf{J}, \tilde{\mathbf{L}} \rangle + \frac{\beta}{2} \|\tilde{\mathbf{L}} - \mathbf{L}_W\| \quad \text{s.t. } \text{rank}(\tilde{\mathbf{L}}) = n - c; \\ \mathbf{J} &= \mathbf{J} + \beta(\tilde{\mathbf{L}} - \mathbf{L}_W). \end{aligned} \quad (13)$$

The first problem above can be solved via quadratic programming because except for the first term, all the other terms contain only $|\mathbf{Z}|$. Thus the sign of all elements in the optimal \mathbf{Z} are the same as those of elements in \mathbf{Z}_0 . The second problem has a closed-form solution via SVD [6]:

$$\tilde{\mathbf{L}} = \mathbf{U}^{1:(n-c)} \boldsymbol{\Sigma}_{1:(n-c)}^{1:(n-c)} (\mathbf{V}^{1:(n-c)})^\top, \quad (14)$$

where $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top = \mathbf{L}_W - \frac{1}{\beta}\mathbf{J}$. The projection process is outlined in Algorithm 3.

Our algorithm has a relatively higher computational complexity, because we try to solve a much more challenging problem than the existing algorithms. The main challenge is the noise in labeled data and the unknown group structure of a large fraction of unlabeled data, which have not been considered in the existing works. Furthermore, our algorithm is suitable for a branch of accelerating strategies. For example, with stochastic sub-gradient descent, our algorithm can be implemented in a distributed way.

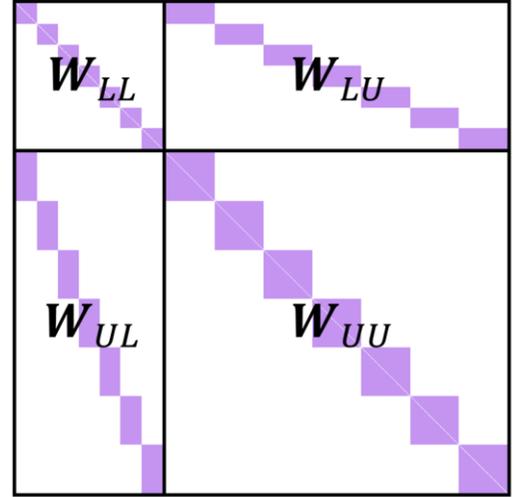


Figure 1. illustration of \mathbf{W} learned via SS-GSR.

4.4 Convergence of the algorithm

The optimization problem in Model (5) is strongly non-convex and we solve it using an EM-like algorithm (Algorithm 1). The motivation of using such an algorithm is based on the fact that minimizing the objective with respect to \mathbf{F} is obviously a convex problem and minimizing the objective with respect to \mathbf{Z} has been approximated using its convex relaxation as in [1]. Therefore, by convex relaxation, the optimization problem actually solved is a bi-convex problem with respect to \mathbf{F} and \mathbf{Z} .

Also, the gradient descent with projection used in solving Z is guaranteed to converge to the global optimum because the Scalable Restricted Isometry Property holds [1, 17].

It is worthy of noting that in our experiments, the algorithm usually gets converged within 10 outer iterations, and achieves satisfactory result.

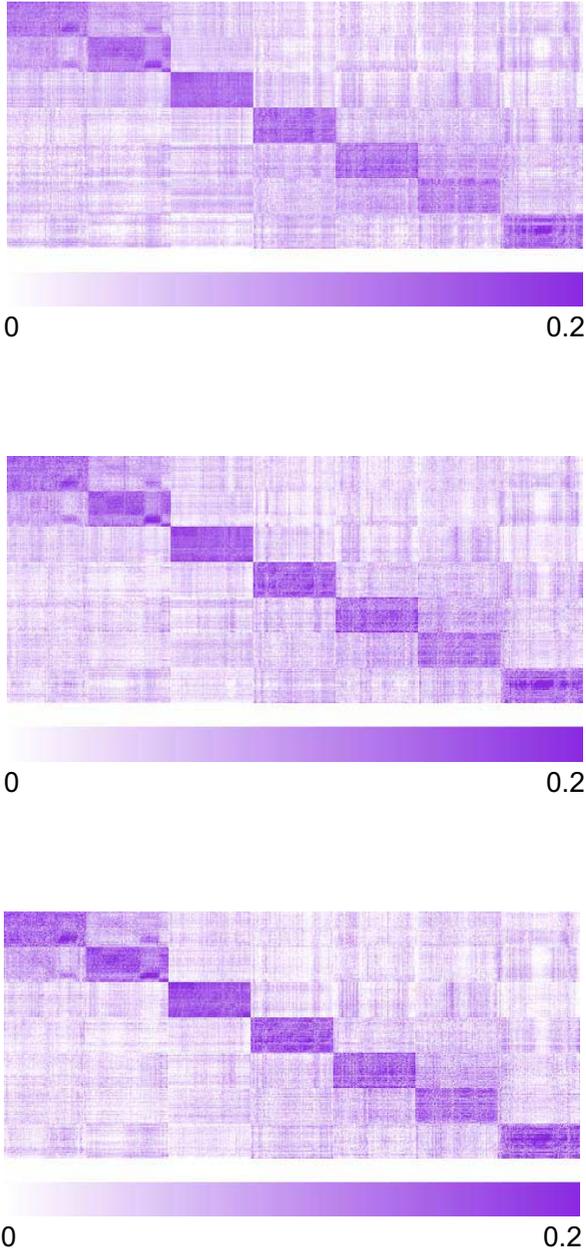


Figure 2. Pictures of Z on Caltech7. From top to bottom are: (a) Z generated by GSR, (b) Z_{LU} generated by SS-GSR-1 and (c) Z_{LU} generated by SS-GSR-2. SS-GSR-1 learns the group structure using only labeled data, while SS-GSR-2 learns the group structure using both labeled and unlabeled data.

Table 1. Group sparsity results of different representation methods on dataset Caltech7. SS-GSR-1 learns the group structure using only labeled data, while SS-GSR-2 learns the group structure using both labeled and unlabeled data.

Representation Method	Group Sparsity
GSR	4455.0
SS-GSR-1	3455.2
SS-GSR-2	3144.8

Table 2. Details of datasets used in the experiments.

Dataset	Data type	Num. of samples	Num. of classes
Caltech7	images	1471	7
PENDIGITS	images	5620	10
OPTDIGITS	images	5620	10
Reuters	texts	7424	6
WEBKB4	texts	4196	4

5 Performance Evaluation and Applications

To validate the effectiveness of our method, here we apply it to both supervised and semi-supervised classification tasks. Concretely, we first evaluate the improvement on representing samples with the help of SS-GSR, we then test the performance of SS-GSR on both a supervised classification task and a semi-supervised classification task, and compare it with some major existing methods. We evaluate those methods with performance metrics *Accuracy*, *Precision* and *Recall*, which are first evaluated on each class and then averaged over the classes.

5.1 Model validation: SS-GSR vs. GSR

We compare the representation abilities of GSR and SS-GSR on dataset Caltech7 [12] in terms of group sparsity. To calculate the group sparsity, we generate the coefficient matrices that use labeled data to represent the other data with the same parameter. For GSR, it is the whole coefficient matrix Z ; for SS-GSR, it is the matrix Z_{LU} . We generate two results for SS-GSR: SS-GSR-1 represents the test data one by one so that the matrix Z_{UU} is fixed as 0, namely it learns only the group structure of the labeled data as in model (4); SS-GSR-2 represents the test data with both labeled and test data, namely it learns the group structure of all data (labeled and unlabeled) as in model (5).

Figures 2 (a)-(c) show the normalized coefficient matrices. As we have already sorted the samples according to their labels, the expected coefficient matrix should be a block-diagonal matrix. In our above figures, those with fewer non-zero elements outside the diagonal blocks are better representations. We can see that the color of the

Table 3. Supervised classification results. Both algorithms classify test samples one by one.

Dataset	Method	Accuracy	Precision	Recall
Caltech7	GSR	96.33%	87.63%	87.14%
	SS-GSR	96.73%	89.05%	88.57%
PENDIGITS	GSR	99.63%	98.33%	98.13%
	SS-GSR	99.63%	98.35%	98.13%
OPTDIGITS	GSR	99.40%	97.32%	97.00%
	SS-GSR	99.42%	97.38%	97.10%
Reuters	GSR	92.67%	65.13%	63.71%
	SS-GSR	93.71%	68.18%	66.86%
WEBKB4	GSR	84.00%	68.00%	68.00%
	SS-GSR	85.13%	71.81%	70.25%

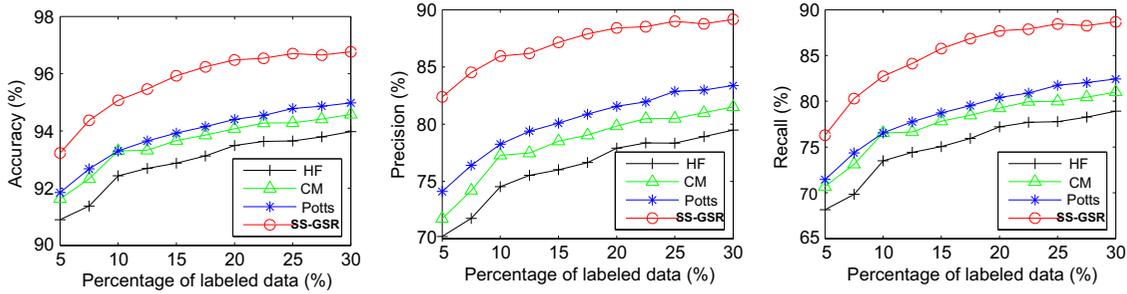


Figure 3. Performance on Caltech7 with 5%-30% labeled samples.

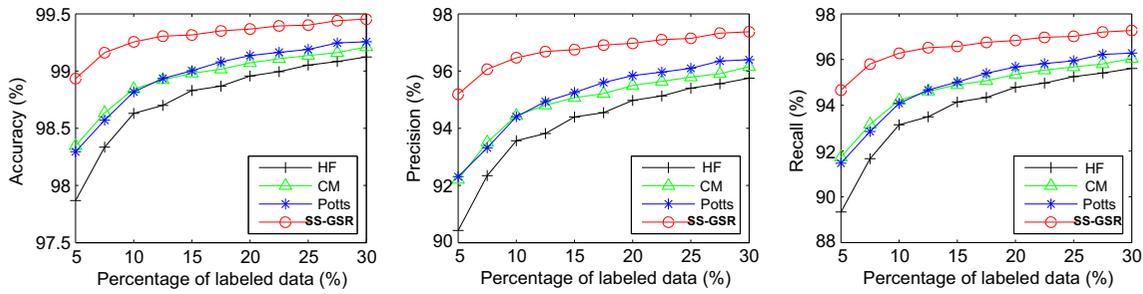


Figure 4. Performance on PENDIGITS with 5%-30% labeled samples.

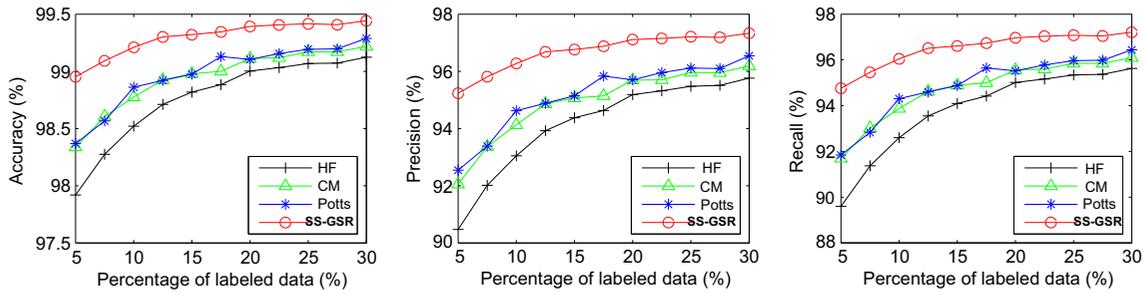


Figure 5. Performance on OPTDIGITS with 5%-30% labeled samples.

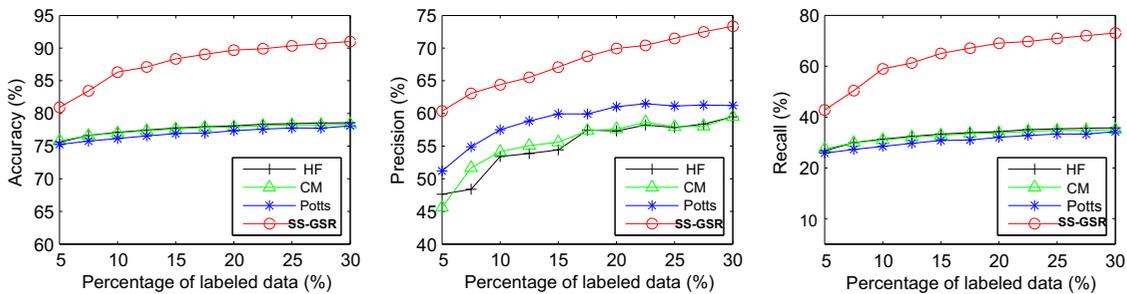


Figure 6. Performance on Reuters with 5%-30% labeled samples.

area outside the diagonal blocks of Figure 2 (a) is obviously deeper than the color of the area outside the diagonal blocks of Figure 2 (b), and the color of the area outside the diagonal blocks of Figure 2 (b) is

slightly deeper than the color of the area outside the diagonal blocks of Figure 2 (c). Thus, Figure 2 (c) is better than Figure 2 (b) and Figure 2 (b) is better than Figure 2 (a). We further calculate the group

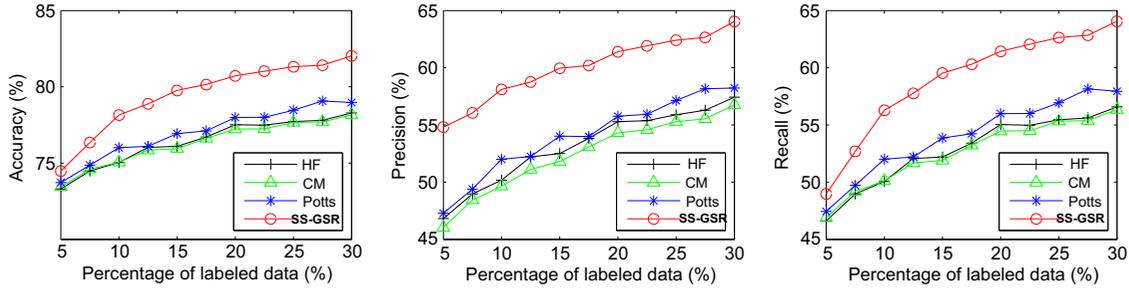


Figure 7. Performance on WEBKB4 with 5%-30% labeled samples.

sparse norm of normalized Z in GSR and normalized Z_{LU} in GSL, and the results are listed in Table 1. By comparing Figure 2 (a), Figure 2 (b) and Figure 2 (c) and checking the results in Table 1, we can conclude that: 1) the coefficient matrices generated by SS-GSR are sparser than the one generated by GSR, which indicates that SS-GSR can more effectively mine and exploit the relationship between the labeled data and the unlabeled data than GSR; 2) As far as sparsity is concerned, learning the structure from the whole data set is better than learning the structure from only the labeled data. The results meet the expectation of our model: exploiting both labeled and unlabeled data in a semi-supervised way can do better group sparse representation.

5.2 Performance comparison in two applications

We apply the new model to two applications: supervised classification and semi-supervised classification, and compare its performance with that of some major existing methods. Five datasets, including Caltech7 [12], PENDIGITS [2], OPTDIGITS [2], Reuters [11] and WEBKB4 [5] are used. The details of these datasets are shown in Table 2. Three performance metrics *Accuracy*, *precision* and *recall* are employed for performance comparison.

5.2.1 Supervised classification task

The first application is text classification, a popular supervised learning task. In this task, our aim is to compare the performance of the traditional GSR model and our new model SS-GSR (when no unlabeled data are used). For each dataset in Table 2, we perform 10-fold cross-validation to compare the classification results of GSR and SS-GSR: give labels to 9 subsets of samples and then use GSR and SS-GSR to classify the rest samples one by one. This process is repeated 10 times and the output results are averaged. The results are shown in Table 3. We can see that SS-GSR works better than GSR in supervised classification. This is because samples in the dictionary are not fully consistent with their labels. For those real datasets, noisy feature vectors can not be given simple labels and outliers cause mislabeling. Nevertheless, SS-GSR can learn a more consistent group structure from the labeled data and selects more precise groups of data according to their underlying group structure.

5.2.2 Semi-supervised classification task

This second application is semi-supervised text classification, a semi-supervised learning task. In this classification task, we compare the capability of our model in group structure learning with that of

three typical (including one proposed recently) graph-based semi-supervised methods:

- **Harmonic function (HF)** [27]: it assumes that the harmonic property of label function should be preserved with respect to the graph with given affinity matrix (weight matrix). Equivalently, this method minimizes the quadratic energy function which results in a harmonic solution.
- **Consistency method (CM)** [26]: it proposes a regularization framework which contains two terms: the smoothness term and the fitting term. The former penalizes on the changes between nearby points and the latter penalizes on the change from the given labels. By trading-off between these two terms, the method finds a smooth solution with respect to the intrinsic structure of data points.
- **Mumford-Shah-Potts model (Potts)** [3]: it extends the Mumford-Shah method [16] and Potts method [19] to transductive learning problem using ℓ_1 relaxation.

For each dataset in Table 2, we give labels only to 5% – 30% random samples (uniformly selected from each class), so the remaining samples are not labeled. We then use the three SSL methods and SS-GSR to decide the labels of the unlabeled samples. This process is repeated 100 times and the output results are averaged. The results of *recall*, *precision* and *accuracy* are presented in Figures 3-7.

From those figures, we can see that SS-GSR clearly outperforms the SSL methods even though they are initialized with the same affinity matrices. There reasons are: on the one hand, manually setting affinity matrix in SSL methods has the consistency problem with real data. On the other hand, the affinity matrix learned by SS-GSR contains the underlying group structure information of all samples and thus is more accurate.

6 CONCLUSION

In this paper, we propose and formulate semi-supervised GSR (SS-GSR) to conduct group sparse representation on datasets containing both labeled and unlabeled data. It can overcome the two drawbacks of the traditional GSR: 1) the pre-defined group structure in GSR may not be fully consistent with that in data; and 2) the underlying group structure of unlabeled data is not exploited in GSR. Compared with GSR, SS-GSR is able to utilize the prior group structure of labeled data properly and take advantage of the group structure information of the unlabeled data. In comparison with SSL methods, SS-GSR can automatically learn the structured affinity matrix from the data instead of using a fixed one. We apply SS-GSR to both supervised and semi-supervised classification tasks, which validate the effectiveness and advantages of SS-GSR.

Acknowledgement

This work was partially supported by the Key Projects of Fundamental Research Program of Shanghai Municipal Commission of Science and Technology under grant No. 14JC1400300.

REFERENCES

- [1] Amir Beck and Marc Teboulle, 'A linearly convergent algorithm for solving a class of nonconvex/affine feasibility problems', in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, 33–48, Springer, (2011).
- [2] Catherine Blake and Christopher J Merz, '{UCI} repository of machine learning databases', (1998).
- [3] Xavier Bresson, Xue-Cheng Tai, Tony F Chan, and Arthur Szlam, 'Multi-class transductive learning based on ℓ_1 relaxations of cheeger cut and mumford-shah-potts model', *Journal of Mathematical Imaging and Vision*, **49**(1), 191–201, (2014).
- [4] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al., *Semi-supervised learning*, volume 2, MIT press Cambridge, 2006.
- [5] Mark Craven, Andrew McCallum, Dan PiPasquo, Tom Mitchell, and Dayne Freitag, 'Learning to extract symbolic knowledge from the world wide web', Technical report, DTIC Document, (1998).
- [6] Carl Eckart and Gale Young, 'The approximation of one matrix by another of lower rank', *Psychometrika*, **1**(3), 211–218, (1936).
- [7] Ehsan Elhamifar and Rene Vidal, 'Sparse subspace clustering: Algorithm, theory, and applications', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(11), 2765–2781, (2013).
- [8] Jiashi Feng, Zhouchen Lin, Huan Xu, and Shuicheng Yan, 'Robust subspace segmentation with block-diagonal prior', in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14)*, pp. 3818–3825. IEEE, (2014).
- [9] Longwen Gao, Shuigeng Zhou, and Jihong Guan, 'Effectively classifying short texts by structured sparse representation with dictionary filtering', *Information Sciences*, **323**, 130–142, (2015).
- [10] Junzhou Huang and Tong Zhang, 'The benefit of group sparsity', *The Annals of Statistics*, **38**(4), 1978–2004, (2010).
- [11] David D Lewis, 'Reuters-21578 text categorization test collection, distribution 1.0', <http://www.research.att.com/~lewis/reuters21578.html>, (1997).
- [12] Fei-Fei Li, Rob Fergus, and Pietro Perona, 'Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories', *Computer Vision and Image Understanding*, **106**(1), 59–70, (2007).
- [13] Yifeng Li and Alioune Ngom, 'Fast sparse representation approaches for the classification of high-dimensional biological data', in *2012 IEEE International Conference on Bioinformatics and Biomedicine (BIBM'12)*, pp. 1–6. IEEE, (2012).
- [14] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma, 'Robust recovery of subspace structures by low-rank representation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(1), 171–184, (2013).
- [15] Angshul Majumdar and Rabab K Ward, 'Classification via group sparsity promoting regularization', in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09)*, pp. 861–864, (Apr. 2009).
- [16] David Mumford and Jayant Shah, 'Optimal approximations by piecewise smooth functions and associated variational problems', *Communications on pure and applied mathematics*, **42**(5), 577–685, (1989).
- [17] Feiping Nie, Xiaoqian Wang, and Heng Huang, 'Clustering and projected clustering with adaptive neighbors', in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 977–986. ACM, (2014).
- [18] Bruno A Olshausen and David J Field, 'Sparse coding with an overcomplete basis set: A strategy employed by v1?', *Vision research*, **37**(23), 3311–3325, (1997).
- [19] Renfrey Burnard Potts, 'Some generalized order-disorder transformations', **48**(01), 106–109, (1952).
- [20] Tara N Sainath, Sameer R Maskey, Bhuvana Ramabhadran, and Dimitri Kanevsky, 'Sparse Representations for Text Categorization', in *INTER-SPEECH'10*, pp. 2266–2269, (2010).
- [21] René Vidal, 'A tutorial on subspace clustering', *IEEE Signal Processing Magazine*, **28**(2), 52–68, (2010).
- [22] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma, 'Robust face recognition via sparse representation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**(2), 210–227, (Feb. 2009).
- [23] Lei Yuan, Alexander Woodard, Shuiwang Ji, Yuan Jiang, Zhi-Hua Zhou, Sudhir Kumar, and Jieping Ye, 'Learning sparse representations for fruit-fly gene expression pattern image annotation and retrieval', *BMC Bioinformatics*, **13**(1), 107, (2012).
- [24] Ming Yuan and Yi Lin, 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68**(1), 49–67, (2006).
- [25] Richard S Zemel and Miguel Á Carreira-Perpiñán, 'Proximity graphs for clustering and manifold learning', in *Advances in Neural Information Processing Systems*, pp. 225–232, (2004).
- [26] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf, 'Learning with local and global consistency', *Advances in Neural Information Processing Systems*, **16**, 321–328, (2004).
- [27] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al., 'Semi-supervised learning using gaussian fields and harmonic functions', in *Proceedings of the 20th Annual International Conference on Machine Learning (ICML'03)*, volume 3, pp. 912–919, (2003).

Modeling Bounded Rationality for Sponsored Search Auctions

Jiang Rong¹ and Tao Qin² and Bo An³ and Tie-Yan Liu²

Abstract. Sponsored search auctions (SSAs) have attracted a lot of research attention in recent years and different equilibrium concepts have been studied in order to understand advertisers' bidding strategies. However, the assumption that advertisers are perfectly rational in these studies is unrealistic in the real world. In this work, we apply the quantal response equilibrium (QRE), which is powerful in modeling bounded rationality, to SSAs. Due to high computational complexity, existing methods for QRE computation have very poor scalability for SSAs. Through exploiting the structures of QRE for SSAs, this paper presents an efficient homotopy-based algorithm to compute the QRE for large-size SSAs, which features the following two novelties: 1) we represent the SSAs as an Action Graph Game (AGG) which can compute the expected utilities in polynomial time; 2) we further significantly reduce redundant calculations by leveraging the underlying relations between advertisers' utilities. We also develop an estimator to infer parameters of SSAs and fit the QRE model into a dataset from a commercial search engine. Our experimental results indicate that the algorithm can significantly improve the scalability of QRE computation for SSAs and the QRE model can describe the real-world bidding behaviors in a very accurate manner.

1 Introduction

Sponsored search has become a major monetization means for commercial search engines (e.g., Google, Yahoo! and Bing) and has shown great success [23, 29, 38]. When a user issues a query to a search engine, in addition to several relevant webpages, a set of selected advertisements will also be displayed on the search result page. To show his/her ad on the search result page, an advertiser (or bidder) is required to submit a bid for the query. Most of the time, there are many more advertisers bidding for the query than the number of available ad slots and the search engines need a mechanism to decide which ads should be shown on the result page, how to allocate the slots to the shown ads, and how to charge an advertiser if his/her ad is clicked by users.

Generalized Second Price (GSP) is the most popular mechanism used in sponsored search auctions (SSAs) and has attracted much research attention recently [16, 35, 38, 40]. Among those studies, equilibrium analysis is a hot topic to understand advertisers' behaviors. Varian [45] studied the concept of symmetric Nash equilibrium for GSP auctions and proved its existence. Edelman et al. [15] defined a subset of Nash equilibria called locally envy-free equilibria which

are equivalent to the symmetric Nash equilibria. Borgers et al. [5] further proved the existence of multiple Nash equilibria in GSP auctions. The forward looking Nash equilibrium was studied in [7, 8].

A critical limitation of existing studies on equilibrium analysis is that they assume the full rationality of advertisers. That is, advertisers are very smart; they can find their optimal strategies and take optimal actions. In practice, an advertiser may fail in estimating his/her competitors' bidding strategies and therefore cannot take the "best-response" action [21, 46]. As a result, it is necessary to study the equilibrium for SSAs based on bounded rationality, which is the focus of this work.

In this paper, we introduce the *quantal response equilibrium* (QRE) [17, 20, 33, 34] into SSAs, which can deal with bounded rationality and has demonstrated very good performance in general normal form games (NFGs). Specifically, because of the limited information about the market and opponents in the real world, an advertiser cannot calculate his/her accurate utility, where the error is assumed to follow some distribution (e.g., the extreme value distribution [33, 34]) with a *precision parameter* (i.e., a measurement of an advertiser's rationality). Due to the disturbance of the errors, advertisers can only maximize their inaccurate utilities in each round of the auctions, which makes their outcome policies to form a QRE — a mixed-strategy equilibrium in which strategies with higher utilities are more likely to be chosen than those with lower utilities, but the best one is not chosen with certainty. A higher precision parameter implies that the advertiser is more rational and hence can choose the better strategies with higher probabilities.

We focus on designing an algorithm for the search engine (or the auctioneer) to compute the QRE⁴ in SSAs, which can be used to capture advertisers' bidding behaviors. We show that the calculation of a QRE is equivalent to computing the fixed point of Browder's functions [6, 28, 42], the complexity of which is at least PPAD-complete [10, 11, 37]. We further formulate the problem as finding a solution of a continuous non-linear function. Basic Newton-type algorithms are usually locally convergent and work well only when we could provide a good starting point, which, however, is difficult to find in SSAs. To address this problem, we leverage the *homotopy* principle [2, 3, 30], which has been used for equilibrium computation [18, 22, 39, 44]. Advantages of homotopy-based methods include their numerical stability and potential to be globally convergent. We noticed that Gambit [32] used a similar method to compute the QRE for NFGs, which, however, is very time-consuming and cannot be directly applied to large-size SSAs. To tackle this challenge, we leverage some nice properties of the SSAs (as compared to general N-

¹ The Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences; University of Chinese Academy of Sciences. Email: rongjiang13@mails.ucas.ac.cn

² Microsoft Research. Email: {taoqin, tie-yan.liu}@microsoft.com

³ School of Computer Science and Engineering, Nanyang Technological University. Email: boan@ntu.edu.sg

⁴ There might be multiple QREs and the one computed with our algorithm is called the principal equilibrium, which is mostly studied in the literature [18, 22, 44]

FGs), including the context-specific independence structure and the underlying relations between advertisers' utilities, to refine the computational procedure and significantly speed up the algorithm. The experimental results show that the improved homotopy algorithm can efficiently compute a QRE for SSAs in large sizes. We also investigate how to use the QRE model to infer the parameters, including the values and precisions of bidders and the click-through-rates (CTRs). We develop an estimating algorithm based on the commonly used Maximum Likelihood Estimation (MLE) principle [1, 26, 36]. Our experiments show that the QRE model fits the real data well.

To sum up, this paper makes two major contributions. First, we design an efficient homotopy-based algorithm to compute the QRE for large-size SSAs by utilizing the nice properties of SSAs. Second, we fit the QRE model into the real data and do extensive experiments to show that, comparing with Nash equilibria, the QRE is more practical since it can handle bidders' bounded rationality and model the real world in a very accurate manner.

The rest of this paper is organized as follows. In Section 2, we introduce the model of the GSP mechanism and then define the QRE for SSAs. The homotopy-based algorithm is proposed in Section 3, including the methods for significantly improving the efficiency of the algorithm. Section 4 gives the parameter estimation algorithm. Then we conduct extensive experiments in Section 5 to evaluate our algorithms. Conclusions are given in the last section.

2 Quantal Response Equilibrium

In this section, we first demonstrate our motivation of investigating the QRE for SSAs, following which we give some notations and assumptions and then define the QRE for SSAs.

2.1 Motivation

Given that GSP is not a dominant-strategy mechanism, Nash equilibrium solutions become an important means to understand how bidders behave in SSAs. While there exist quite a few Nash equilibrium concepts proposed and studied for SSAs, symmetric Nash equilibrium [45] and locally envy-free Nash equilibrium [15] are the most famous two: the former captures the notion that there should be no incentive for any pair of bidders to swap their slots, and the latter captures the notion that there should be no incentive for any bidder to exchange bids with the bidder ranked one position above him/her.

While those equilibrium concepts have many nice properties, a common limitation of them is that they assume the perfect rationality of bidders, i.e., bidders have perfect knowledge about their utilities and take optimal actions to maximize their utilities. However, the perfect rationality assumption is too good to be true in real-world SSAs. Therefore, a natural question arises: how can we weaken the perfect-rationality assumption and still obtain some meaningful solution concept for SSAs?

Observing that in real-world SSAs, an advertiser usually has uncertainty about his/her utility and is more likely to choose strategies with higher utilities instead of always choosing the best one. We introduce the quantal response equilibrium to model the bounded rationality of bidders in the following two subsections.

2.2 Notations and Assumptions

We focus on the GSP mechanism. Generally there are N bidders competing for K ad slots ($N \geq K$). We use the symbol $[N]$ to represent the set $\{1, 2, \dots, N\}$. Let v_i denote the private value of bidder

i , which expresses the maximum per-click price he/she is willing to pay, and the vector $v = (v_1, v_2, \dots, v_N)$ represents the value profile of all bidders. We use b_i to represent the bid submitted by i to participate in the auction. θ_{ik} is the CTR of i 's ad when placed at slot k , which is usually assumed to be the product of the ad CTR α_i and the slot CTR β_k [38]. We use $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)$ and $\beta = (\beta_1, \beta_2, \dots, \beta_K)$ to denote the profiles of ad and slot CTRs respectively. Following the common practice [38], without loss of generality, we assume

$$\beta_1 \geq \beta_2 \geq \dots \geq \beta_K.$$

In the GSP mechanism, bidders are ranked in the descending order of their ranking scores which is defined as

$$s_i = \alpha_i b_i.$$

The top $k \leq K$ bidders whose scores are not less than the reserve price r are allocated at the first k slots. If an ad is clicked, the payment of the corresponding bidder is the minimum amount that maintains his/her current rank position.

2.3 Definition of QRE

Let B_i denote advertiser i 's bid space and b_{ij} be the j -th minimal price in B_i . We define the score space of bidder i as

$$S_i = \{s_{ij} | s_{ij} = \alpha_i b_{ij}, j \in [|B_i|]\},$$

where $|B_i|$ represents the size of B_i , and define the joint score space of all advertisers except i as

$$S_{-i} = \times_{l \in [N] \setminus \{i\}} S_l.$$

Then we have that

$$|S_i| = |B_i|, \forall i \in [N].$$

Let $q_{ij}(s_{-i})$ and $p_{ij}(s_{-i})$ denote the slot allocated to and the payment of bidder i , given $s_{-i} \in S_{-i}$ and $s_{ij} \in S_i$. Then the utility of advertiser i is

$$u_{ij}(s_{-i}) = \begin{cases} 0, & s_{ij} < r; \\ (v_i - p_{ij}(s_{-i}))\alpha_i\beta_{q_{ij}(s_{-i})}, & s_{ij} \geq r. \end{cases} \quad (1)$$

Let σ_i be i 's mixed strategy over B_i and σ_{ij} denote the probability on b_{ij} . Similarly, we define $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_N)$ and $\sigma_{-i} = (\sigma_1, \dots, \sigma_{i-1}, \sigma_{i+1}, \dots, \sigma_N)$. The expected utility of i with s_{ij} , given σ_{-i} , is

$$\bar{u}_{ij}(\sigma_{-i}) = \sum_{s_{-i} \in S_{-i}} P(s_{-i} | \sigma_{-i}) u_{ij}(s_{-i}), \quad (2)$$

where $P(s_{-i} | \sigma_{-i})$ is the probability that the score profile of other bidders except i is s_{-i} given σ_{-i} . The quantal response π_{ij} of bidder i to others' mixed strategy profile σ_{-i} is defined as⁵

$$\pi_{ij}(\sigma_{-i}) = \frac{1}{\lambda_i} \cdot \frac{1}{|B_i|} + \left(1 - \frac{1}{\lambda_i}\right) \frac{e^{\bar{u}_{ij}(\sigma_{-i})\lambda_i}}{\sum_{k \in [|B_i|]} e^{\bar{u}_{ik}(\sigma_{-i})\lambda_i}}$$

⁵ In the commonly-used logit form of quantal response [33, 44], multiplying α_i and dividing λ_i for all $i \in [N]$ by the same constant will not change the QRE outcome, which means that they are undistinguishable. To address this problem, we use a slightly different definition which satisfies the principle of QRE, i.e., better strategies are more likely to be chosen.

$$= \frac{1}{\lambda_i |B_i|} + \left(1 - \frac{1}{\lambda_i}\right) \frac{1}{\sum_{k \in [|B_i|]} e^{(\bar{u}_{ik}(\sigma_{-i}) - \bar{u}_{ij}(\sigma_{-i}))\lambda_i}}, \quad (3)$$

where $\lambda_i \in [1, +\infty)$ is the *precision parameter* of bidder i . We can easily verify that Eq. (3) is consistent with our expectation that better strategies are more likely to be chosen than worse ones. When $\lambda_i = 1$, we have $\pi_{ij}(\sigma_{-i}) = \frac{1}{|B_i|}$, which means that i uniformly chooses any strategy in B_i ; and when $\lambda_i \mapsto +\infty$, the choice probability of the bid strategy with the highest expected utility approaches 1. To sum up, λ_i is a parameter to measure i 's "rationality". That is, a larger λ_i suggests that i will choose the best strategy with a higher probability. We use the vector $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_N)$ to denote the precision profile. A QRE [17, 33] is defined as follows.

Definition 1. A quantal response equilibrium with λ is a mixed strategy profile σ such that for all $i \in [N]$ and $j \in [|B_i|]$, $\sigma_{ij} = \pi_{ij}(\sigma_{-i})$.

Definition 1 implies that a QRE strategy profile is a fixed point of Browder's functions [6]. Besides, the Nash equilibrium is a special case of the QRE with $\lambda_i \rightarrow +\infty$ for all $i \in [N]$.

3 Equilibrium Computation

In this section, we design an efficient homotopy-based algorithm to compute the QRE for SSAs. We first describe the algorithm and then show how to significantly speed it up by exploiting peculiarities of SSAs.

3.1 The Homotopy-based Algorithm

In this subsection, we discuss how to compute a QRE given v , λ , α and β . We define

$$U = \sum_{i \in [N]} |B_i|$$

and obtain a continuous function $F : [0, 1]^U \mapsto [0, 1]^U$ from Definition 1 as below:

$$F_{ij}(\sigma) = \pi_{ij}(\sigma_{-i}) - \sigma_{ij}, \forall i \in [N], j \in [|B_i|].$$

Now we can see that computing a QRE of SSAs is equivalent to finding a zero point of the nonlinear function $F(\sigma)$. If a good starting point is available, we can directly apply Newton-style iteration methods. However, we have little information about such a good initial point. As pointed out by Allgower and Georg [2, 3], Newton-style iteration methods often fail because poor starting points are very likely to be chosen.

The basic idea of the homotopy is composed of two steps: given a problem we want to solve, first, define a problem with a unique easy-to-compute solution and then build a continuous transformation from the artificial problem into the original problem we want to solve; second, begin with the solution of the easy-to-solve problem and trace solutions of the associated problems of the transformation until finally the solution of the original problem is found.

To design a homotopy-based algorithm, our first step is to propose a degenerate problem which is easy solve. In particular, we can find a degenerated form of F (denoted as G) by letting $\bar{u}_{ij}(\sigma_{-i})$, $\forall i \in [N]$ and $j \in [|B_i|]$, be zero:

$$G_{ij}(\sigma) = \frac{1}{|B_i|} - \sigma_{ij}, i \in [N], j \in [|B_i|].$$

Obviously, $G(\sigma)$ has a unique zero point: $\sigma_{ij} = 1/|B_i|, \forall i \in [N], j \in [|B_i|]$. Then we define a homotopy function $H : [0, 1]^U \times [0, 1] \mapsto [0, 1]^U$ between $F(\sigma)$ and $G(\sigma)$ as

$$H_{ij}(\sigma, t) = \frac{1}{\lambda_i |B_i|} + \left(1 - \frac{1}{\lambda_i}\right) R(\sigma_{-i}, t)^{-1} - \sigma_{ij},$$

$$R(\sigma_{-i}, t) = \sum_{k \in [|B_i|]} e^{(\bar{u}_{ik}(\sigma_{-i}) - \bar{u}_{ij}(\sigma_{-i}))\lambda_i t},$$

which is a continuous transformation from $H(\sigma, 0) = G(\sigma)$ to $H(\sigma, 1) = F(\sigma)$ as t grows continuously from 0 to 1.

We further define the solution set of $H(\sigma, t) = 0$ as

$$H^{-1}(0) = \{(\sigma, t) | H(\sigma, t) = 0\}.$$

It follows from Browder's fixed point theorem [6] that, for a given $t \in [0, 1]$, there is a $\sigma(t)$ such that $H(\sigma(t), t) = 0$. From the definition of H we know that $\sigma(0)$ and $\sigma(1)$ correspond to the zero point of $G(\sigma)$ and $F(\sigma)$ respectively. The remaining problem is to trace out a path consisting of $(\sigma(t), t) \in H^{-1}(0)$, which starts at $(\sigma(0), 0)$ and ends at $(\sigma(1), 1)$. Considering the possibility of the existence of turning points [27], increasing t monotonically when tracing the path may lead to points far away from the path. A common practice to avoid the disturbance of turning points is to view the σ and t as functions of an implicit parameter a simultaneously and to compute a parametric path

$$c(a) = (\sigma(t(a)), t(a)),$$

which satisfies

$$H(c(a)) = 0. \quad (4)$$

The method we use to trace the path is called predictor-corrector (PC) [2, 3], the basic idea of which is to numerically trace the path $c(a)$ by generating a sequence of points $c_i = (\sigma, t)_i, i = 1, 2, \dots$ along the path satisfying $\|H(c_i)\| \leq \varepsilon$ for some $\varepsilon > 0$. In particular, given that we have found a point c_i on the path $c(a)$, an Euler predictor step is used to predict the next point c_{i+1} on $c(a)$:

$$c_{i+1} = c_i + \Delta \cdot \frac{c'(a)|_{c(a)=c_i}}{\|c'(a)|_{c(a)=c_i}\|}, \quad (5)$$

where $c'(a)$ is the derivative of $c(a)$ with respect to a and $\Delta > 0$ is the step length. Then a corrector phase is necessary to refine the accuracy of c_{i+1} . We make use of the Gauss-Newton method as presented below:

$$\hat{c}_{i+1} = c_{i+1} - H'(c_{i+1})^+ H(c_{i+1}), \quad (6)$$

where $H'(c_{i+1})^+$ is the Moore-Penrose inverse⁶ of the Jacobian matrix $H'(c_{i+1})$ of $H(\cdot)$ at point c_{i+1} , and \hat{c}_{i+1} is the refined point of c_{i+1} . If $\|H(\hat{c}_{i+1})\| > \varepsilon$, we will substitute \hat{c}_{i+1} into the right side of Eq. (6) to further refine it. The corrector procedure may be performed several times until we find the satisfactory point which will be used in the predictor phase to infer the next point. The PC method, starting with $(\sigma(0), 0)$, is applied step by step until $(\sigma(1), 1)$ is reached.

Now we discuss how to compute the derivative $c'(a)$ in Eq. (5) and the Jacobian matrix $H'(c_{i+1})$ in Eq. (6). We first consider the calculation of $c'(a)$. By differentiating Eq. (4) we get the following equation:

$$H'(c(a))c'(a) = 0. \quad (7)$$

⁶ The Moore-Penrose inverse is defined by $A^+ = A^T(AA^T)^{-1}$.

The solution of Eq. (7) is

$$c'_d(a) = \mu \cdot (-1)^d \cdot \det(H'_{-d}(c(a))) \quad (8)$$

where $c'_d(a)$, $d=1, \dots, U+1$, denotes the d -th⁷ component of $c'(a)$ and $H'_{-d}(c(a))$ is $H'(c(a))$ with the d -th column removed; $\det(\cdot)$ is the determinant operation; $\mu = \pm 1$ is the sign of $c'(a)$ to be chosen. We know from Eq. (8) that once $H'(c(a))$ is given, $c'(a)$ can be obtained directly. Eq. (6) also involves computing $H'(\cdot)$. So next we will concentrate on how to compute $H'(\cdot)$. We use \tilde{u}_{ij} to represent $\bar{u}_{ij}(\sigma_{-i})\lambda_i$ for simplicity in the remaining part of this paper. Since the function H is a mapping from $[0, 1]^U \times [0, 1]$ to $[0, 1]^U$, its Jacobian matrix contains $U \cdot (U + 1)$ partial derivatives that can be divided into four cases:

Case 1. $i \in [N]$ and $j \in [|B_i|]$:

$$\frac{\partial H_{ij}}{\partial \sigma_{ij}} = -1;$$

Case 2. $i \in [N]$, j and $k \in [|B_i|]$, $j \neq k$:

$$\frac{\partial H_{ij}}{\partial \sigma_{ik}} = 0;$$

Case 3. i and $l \in [N]$, $j \in [|B_i|]$, $m \in [|B_l|]$, $i \neq l$:

$$\begin{aligned} \frac{\partial H_{ij}}{\partial \sigma_{lm}} &= -(1 - \frac{1}{\lambda_i})R(\sigma_{-i}, t)^{-2} \frac{\partial R(\sigma_{-i}, t)}{\partial \sigma_{lm}}, \\ \frac{\partial R(\sigma_{-i}, t)}{\partial \sigma_{lm}} &= te^{-\tilde{u}_{ij}t} \sum_{k \in [|B_i|]} e^{\tilde{u}_{ik}t} \left(\frac{\partial \tilde{u}_{ik}}{\partial \sigma_{lm}} - \frac{\partial \tilde{u}_{ij}}{\partial \sigma_{lm}} \right); \end{aligned}$$

Case 4. $i \in [N]$, $j \in [|B_i|]$:

$$\begin{aligned} \frac{\partial H_{ij}}{\partial t} &= -(1 - \frac{1}{\lambda_i})R(\sigma_{-i}, t)^{-2} \frac{\partial R(\sigma_{-i}, t)}{\partial t}, \\ \frac{\partial R(\sigma_{-i}, t)}{\partial t} &= e^{-\tilde{u}_{ij}t} \sum_{k \in [|B_i|]} e^{\tilde{u}_{ik}t} (\tilde{u}_{ik} - \tilde{u}_{ij}). \end{aligned}$$

We choose μ to ensure that the derivative of t with respect to a , which corresponds to the $(U + 1)$ -th component in $c'(a)$, is positive at $(\sigma(0), 0)$, i.e.,

$$\mu \cdot (-1)^{U+1} \det(H'_{-(U+1)}(\sigma(0), 0)) > 0.$$

Substituting $t = 0$ into this inequality and combining with cases 1-3, we get

$$\mu \cdot (-1)^{U+1} \cdot (-1)^U = (-1)^{2U+1} \mu > 0,$$

which indicates that $\mu = -1$ at $(\sigma(0), 0)$.

By now we have discussed how to compute Eqs. (5) and (6). Then we can use the PC method to find the point $(\sigma(1), 1) \in H^{-1}(0)$ step by step, the convergence property of which is analyzed in [2, 3].

The complete process of our proposed method is summarized in Algorithm 1. In line 1 we assign t with 0 and the starting point c_1 with $(\sigma(0), 0)$. Line 2 initializes Δ and ϵ . Lines 3-8 use the PC method to generate a set of points c_i , $i = 2, 3, \dots$ along the path until eventually the point $(\sigma(1), 1)$ is found. Line 4 is the Euler predictor step which computes the next point c_{i+1} given c_i according to Eq. (5). The Gauss-Newton corrector step is performed repeatedly in lines 5-6 to improve the accuracy of the point predicted in line 4. The value of t is updated in line 7 and the step length is adjusted in line 8 based on the Asymptotic Expansion method proposed in [2, 3]. The result is returned in line 10.

⁷ σ_{ij} 's are assumed to be assigned to $c(a)$ in lexicographic order of their subscripts. The last component of the vector corresponds to t .

Algorithm 1: Computing a QRE

```

1  $t \leftarrow 0, c \leftarrow (\sigma(0), 0)$ ;
2 initialize  $\Delta$  and  $\epsilon$ ;
3 while  $t \neq 1$  do
4    $c \leftarrow c + \Delta \cdot \frac{c'(a)|_{c(a)=c}}{\|(c'(a)|_{c(a)=c})\|}$ ;
5   while  $\|H(c)\| > \epsilon$  do
6      $c \leftarrow c - H'(c)^+ H(c)$ ;
7    $t \leftarrow$  the last component of  $c$ ;
8   Adjust the step length  $\Delta$ ;
9  $(\sigma, t) \leftarrow c$ ;
10 return  $\sigma$ ;
```

3.2 Efficient Computation for SSAs

Algorithm 1 indicates that we need to compute the Jacobian matrix $H'(\cdot)$ at each predictor and corrector step when tracing the path. Thus the efficiency of calculating $H'(\cdot)$ will significantly affect the scalability of the algorithm. In this subsection, we discuss how to reduce the complexity of Algorithm 1 through efficient calculation of $H'(\cdot)$ by leveraging the properties of SSAs. First, we represent the SSAs as an Action Graph Game (AGG) [24, 25, 43] which can compute the components of $H'(\cdot)$ in polynomial time, while general NFGs cost exponential time to calculate them. Second, we further significantly reduce redundant calculations based on the analyses of the relations between advertisers' utilities.

3.2.1 Representing SSAs as AGG

The elements in $H'(\cdot)$ are classified into four cases as shown in Section 3.1, the last two cases of which involve computing \tilde{u}_{ij} and $\frac{\partial \tilde{u}_{ij}}{\partial \sigma_{lm}}$. We can rewrite Eq. (2) as

$$\begin{aligned} \bar{u}_{ij}(\sigma_{-i}) &= \sum_{m \in [|S_l|]} \sigma_{lm} \sum_{s_{-il} \in S_{-il}} P(s_{-il} | \sigma_{-il}) u_{ij}(s_{-il}, s_{lm}) \\ &= \sum_{m \in [|S_l|]} \sigma_{lm} \frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}}, \forall l \neq i, \end{aligned} \quad (9)$$

where $S_{-il} = \times_{i' \in [N] \setminus \{i, l\}} S_{i'}$ and s_{-il} is an element of S_{-il} . So the main effort on calculating $H'(\cdot)$ is to compute a set of partial derivatives, i.e.,

$$D = \left\{ \frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}} \mid i, l \in [N]; l \neq i; j \in [|B_i|]; m \in [|B_l|] \right\}.$$

We need to traverse the $\prod_{i' \in [N] \setminus \{i, l\}} |B_{i'}|$ realizations in S_{-il} to compute $\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}}$ if we view SSAs as general NFGs. Clearly, this traversal method (TM for short) is exponential, i.e., $O(M^N)$, where

$$M = \max\{|B_{i'}| \mid i' \in [N]\}.$$

Fortunately, expected utilities in SSAs with GSP mechanism have many special properties that could be utilized to reduce the computational complexity. Here we take bidder i with s_{ij} and bidder l with s_{lm} as an example ($s_{ij} \geq r$) and define

$$I_G = \{n \mid s_n > s_{ij}, n \neq i\}.$$

Similarly, we further define

$$I_E = \{n \mid s_n = s_{ij}, n \neq i\}$$

and

$$I_L = \{n | s_n < s_{ij}, n \neq i\}.$$

We assume the tie is broken randomly. It thus follows that:

1. When $|I_G| \geq K$, i 's utility is zero.
2. When $|I_G| < K$ and $|I_G| + |I_E| \geq K$, bidder i has a probability $\frac{1}{|I_E|+1}$ to be allocated at a slot ranging from $|I_G| + 1$ to K and his/her payment is $p_i = b_{ij}$. According to our assumption on the tie, i ' utility in such case is

$$\frac{1}{|I_E|+1} (v_i - b_{ij}) \sum_{k=|I_G|+1}^K \alpha_i \beta_k.$$

3. When $|I_G| + |I_E| < K$, i 's location will be any one from slot $|I_G|+1$ to slot $|I_G|+|I_E|+1$ with an identical probability $\frac{1}{|I_E|+1}$. His/her payment is $p_i = b_{ij}$ if ranked at the slot between $|I_G| + 1$ and $|I_G| + |I_E|$. On the other hand, his/her payment is p_{max}/α_i if allocated at slot $|I_G| + |I_E| + 1$, where

$$p_{max} = \max\{r, s_n | n \in I_L\}.$$

Bidder i 's utility in this case is

$$\frac{1}{|I_E|+1} \left((v_i - b_{ij}) \sum_{k=|I_G|+1}^{|I_G|+|I_E|} \alpha_i \beta_k + (v_i - p_{max}) \alpha_i \beta_{|I_G|+|I_E|+1} \right).$$

The above properties indicate that given bidder i 's score $s_i = s_{ij}$, his/her utility only depends on $|I_G|$, $|I_E|$ and p_{max} , but not on who are in I_G and I_E or exactly what their bids are, and not on whose ranking score is p_{max} . That is SSAs have considerable context-specific independence structure and can be represented compactly by an AGG. An action graph (AG) is a trie⁸, each leaf of which corresponds to a tuple $(|I_G|, |I_E|, p_{max})$. Specifically, when computing $\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}}$, because 1) we just consider the cases where $|I_G| < K$, 2) $|I_E| \leq N$, and 3) p_{max} has at most NM different values, the AG has $O(KN^2M)$ leaves, which can be built in time $O(KN^2MN) = O(KN^3M)$ by a dynamic program [24]. Compared with that of TM ($O(M^N)$), it is a significant improvement.

3.2.2 Reducing Redundancy

The computation of the set D with AGG involves two steps: 1) building the AGs and 2) calculating the partial derivatives based on these AGs. Intuitively, we need to apply the two procedures to each of the $N(N-1)M^2$ elements of D . Actually, we can significantly reduce the calculations by utilizing the properties of SSAs.

We first focus on the process of AG construction (i.e., step 1). The AG for $\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}}$ is built by a dynamic program which traverses the union set

$$\{s_{nk} | n \in [N] \setminus \{i, l\}, k \in [|B_n|]\} \cup \{s_{lm}\},$$

the first part of which is the same for all $m \in [|B_l|]$. Thus we can just build one trie with $\{s_{nk} | n \in [N] \setminus \{i, l\}, k \in [|B_n|]\}$, from which the AGs for $\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}}, \forall m \in [|B_l|]$, can be directly derived by further taking s_{lm} into consideration. This observation implies that we can compute D by building at most $N(N-1)M$ AGs. Next we show how to further refine step 1 based on the following propositions.

⁸ Trie is an ordered tree data structure. More details can be found at <https://en.wikipedia.org/wiki/Trie> and [4].

Proposition 1. For all $l \neq i$, $\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}} = 0$ if $s_{ij} < r$.

This proposition is straightforward since Eq. (1) indicates that $u_{ij}(s_{-il}, s_{lm}) = 0$ if $s_{ij} < r$.

Proposition 2. For all $l \neq i$,

$$\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}} = \frac{\partial \bar{u}_{ij+1}(\sigma_{-i})}{\partial \sigma_{lm}} = \dots = \frac{\partial \bar{u}_{ij+k}(\sigma_{-i})}{\partial \sigma_{lm}},$$

if there is no $s_{l'm'}$, $l' \neq i$ and $m' \in [|B_{l'}|]$, satisfying $s_{ij} \leq s_{l'm'} \leq s_{ij+k}$.

Proof. I_E is empty under the assumption. Given b_{-il} , bidders except i are either in I_G or in I_L and will not change their positions when i 's score changes from s_{ij} to s_{ij+n} , $n \in [k]$. Then we have that, for all $s_{-il} \in S_{-il}$, $u_{ij}(s_{-il}, s_{lm}) = u_{ij+n}(s_{-il}, s_{lm}), \forall n \in [k]$, because they have the same I_G , I_E and I_L . Taking expectation over S_{-il} completes the proof according to Eq. (9). \square

Proposition 3. For all $l \neq i$, the AGs for $\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}}$ and $\frac{\partial \bar{u}_{lj}(\sigma_{-i})}{\partial \sigma_{im}}$ are the same if $S_i = S_l$.

Proof. Since the AG is indeed a trie, we only need to show that they have the same leaves. Given $s_{-il} \in S_{-il}$, because $s_{ij} = s_{lj}$ and $s_{im} = s_{lm}$, the profiles $s_{-i} = (s_{-il}, s_{lm})$ for s_{ij} and $s_{-l} = (s_{-il}, s_{im})$ for s_{lj} can be mapped to the same tuple $(|I_G|, |I_E|, p_{max})$. By traversing S_{-il} and considering that there is a one-to-one correspondence between the tuples and leaves, we prove the proposition. \square

Proposition 2 shows that if a subset $\{s_{ij}, s_{ij+1}, \dots, s_{ij+k}\}$ of S_i satisfies the constraint, then $\frac{\partial \bar{u}_{ij+1}(\sigma_{-i})}{\partial \sigma_{lm}}, \dots, \frac{\partial \bar{u}_{ij+k}(\sigma_{-i})}{\partial \sigma_{lm}}$ can be obtained directly through $\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}}$ without building the AGs. Proposition 2 cannot be applied to the case where bidders have the same score space. By contrast, Proposition 3 is particularly useful for this case, which can reduce the number of AGs to $\frac{N(N-1)M}{2}$ in step 1. The next proposition is used to speed up step 2.

Proposition 4. For all $l \neq i$,

$$\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm_1}} = \frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm_2}},$$

if 1) $s_{lm_1}, s_{lm_2} \leq r$ or 2) $s_{lm_1}, s_{lm_2} > s_{ij}$.

Proof. Similar to the proof of Proposition 2, we just need to prove that for all $s_{-il} \in S_{-il}$, $u_{ij}(s_{-il}, s_{lm_1}) = u_{ij}(s_{-il}, s_{lm_2})$, which is true due to the fact that (s_{-il}, s_{lm_1}) and (s_{-il}, s_{lm_2}) correspond to the same tuple $(|I_G|, |I_E|, p_{max})$ for s_{ij} on each of the two conditions. \square

We use an example where bidders have the same score space to analyze the effects of Propositions 3 and 4 in step 2. We assume $r = 0$ for ease of analysis. Given i and l , we only need to compute $\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}}$ for $j < M$ and $m \leq j+1$, and $\frac{\partial \bar{u}_{lM}(\sigma_{-i})}{\partial \sigma_{lm}}$ for $m \leq M$, because those for $j < M$ and $m > j+1$ are equal to $\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lj+1}}$ according to Proposition 4. Then the amount of components calculated with AGs is

$$(2 + 3 + \dots + M + M) = \frac{M^2 + 3M - 2}{2}.$$

On the other hand, given j and m , Proposition 3 implies that $\frac{\partial \bar{u}_{ij}(\sigma_{-i})}{\partial \sigma_{lm}}$ and $\frac{\partial \bar{u}_{lj}(\sigma_{-i})}{\partial \sigma_{im}}$ as a pair can be computed simultaneously

with one AG. There are $\frac{N(N-1)}{2}$ such pairs. As a result, the number of calculations reduced in total is

$$\begin{aligned} & N(N-1)M^2 - \frac{N(N-1)}{2} \frac{M^2 + 3M - 2}{2} \\ & > \frac{3}{4}N(N-1)M(M-1). \end{aligned}$$

Since D has $N(N-1)M^2$ elements, the efficiency for step 2 is improved by about 75%.

4 Parameter Estimation

In this subsection, we propose an algorithm based on the principle of MLE and QRE to estimate v , α and β of SSAs from real data. It is pointed out in [19] that the auctioneer itself can not accurately estimate the CTRs and it is not rare to get a 50% error. Our model provides an alternative way for the search engine to infer these parameters when they are unknown.

Given a QRE strategy σ , the logarithmic likelihood of the unknown parameters v , λ , α and β is

$$L(v, \lambda, \alpha, \beta | \sigma) = \log \left(\prod_{i \in [N]} \prod_{j \in [B_i]} \pi_{ij} (\sigma_{-i})^{\sigma_{ij}} \right) \quad (10)$$

Then the parameters can be estimated by maximizing the likelihood as shown in the following optimization problem:

$$\max_{v, \lambda, \alpha, \beta} L(v, \lambda, \alpha, \beta | \sigma) \quad (11)$$

$$s.t. \begin{cases} v_i \geq 0, \forall i \in [N]; \\ \lambda_i \geq 1, \forall i \in [N]; \\ 1 > \beta_s \geq \beta_{s+1} > 0, \forall s \in [K-1]; \\ 0 < \alpha_i < 1, \forall i \in [N]. \end{cases} \quad (12)$$

However, $q_{ij}(s_{-i})$ is not a continuous function of α_i , $\forall i \in [N]$, nor are the utilities of bidders. As a result, the likelihood defined in Eq. (10) is not continuous with respect to α .

To address the discontinuity of the likelihood function, we split the unknown parameters into two groups and sequentially optimize them: we treat v, λ, β as a group and α as the other group; in each iteration, we first optimize v, λ, β and then α .

The function $L(v, \lambda, \alpha, \beta | \sigma)$ in Eq. (11) is continuous with respect to the parameters in the first group. We can learn a better set of v, λ, β by solving the following sub optimization problem:

$$\max_{v, \lambda, \beta} L(v, \lambda, \alpha, \beta | \sigma) \quad (13)$$

$$s.t. \begin{cases} v_i \geq 0, \forall i \in [N]; \\ \lambda_i \geq 1, \forall i \in [N]; \\ 1 > \beta_s \geq \beta_{s+1} > 0, \forall s \in [K-1]. \end{cases} \quad (14)$$

Since the above optimization problem is non-convex, it is difficult to find the global maximum. We turn to find a set of local maxima with different starting points and then choose the best one to improve the possibility of reaching the global maximum of the sub problem.

As aforementioned, the likelihood function is not continuous with respect to α . Here we do not optimize bidders' ad CTRs simultaneously. Instead, we deal with them one by one. Let us take the ad CTR α_i of bidder i as an example and keep $\alpha_j, \forall j \neq i$ fixed. Given that all

the other parameters are fixed, it is easy to know that the likelihood has the following discontinuous points:

$$\left\{ \frac{\alpha_j b_j}{b_i} \mid 0 < \frac{\alpha_j b_j}{b_i} < 1, j \in [N] \setminus \{i\}, b_i \in B_i, b_j \in B_j \right\}.$$

Then we can partition the feasible domain of α_i into several intervals where the likelihood function L is continuous with respect to α_i , and then by solving the optimization problem defined in Eq. (15) in each interval we can find a better α_i given all the other parameters:

$$\max_{\alpha_i} L(v, \lambda, \alpha, \beta | \sigma) \quad (15)$$

s.t. α_i in the continuous interval.

Similarly, the above optimization problem is not convex. To avoid being tracked into a bad local maximum, we can also find a set of local maxima with different starting points and choose the best one.

Algorithm 2: Parameter estimation

```

1  $L^* \leftarrow -\infty$ ;
2 Randomly generate an ad CTR profile  $\alpha$ ;
3 while True do
4   Fix  $\alpha$  and update  $v, \lambda, \beta$  by solving the problem shown in
   Eqs. (13) and (14);
5   for  $i \leftarrow 1, 2, \dots, N$  do
6     Fix  $\alpha_j, \forall j \neq i, v, \lambda, \beta$  and update  $\alpha_i$  by solving the sub
     problem as shown in Eq. (15) in each continuous
     interval;
7    $\hat{L} \leftarrow L(v, \lambda, \alpha, \beta | \sigma)$ ;
8   if  $\hat{L} > L^*$  then
9      $L^* \leftarrow \hat{L}$ ;
10  else return the learned parameters  $v, \lambda, \alpha, \beta$ ;
```

The complete procedure is presented in Algorithm 2. In line 1 we initialize the likelihood of the original optimization problem with negative infinity. Line 2 sets an initial α . Lines 3-9 iteratively optimize the two groups of parameters. Line 4 fixes α and updates (v, λ, β) . Lines 5-6 fix (v, λ, β) and update α . Lines 7-10 control the optimization process: if we make progress in this iteration, we continue the optimization; otherwise, we return the latest parameters. Again, to avoid a bad local maximum, we run the algorithm for multiple times with different initial α 's in Line 2 and choose the best parameters as the final output in our experiments.

5 Experimental Evaluation

We conduct extensive experiments to evaluate the algorithms for QRE computation and parameter estimation.

5.1 Effectiveness of the Homotopy Algorithm

We first evaluate the runtime of the three different approaches for computing D : TM, AGG, and AGG combined with our speed-up methods (AGGSU for short). The experiments are divided into two groups based on whether bidders have the same score space. In each group, we test the three methods with different game sizes where $N = 5, 6, \dots, 20$ and $M = 10, 15, 20$. In all the experiments, α, β and $B_i, \forall i \in [N]$, are sampled from uniform distributions with supports $(0.1, 1)^N, (0.1, 1)^K$ and $(0, M)^M$ respectively. We let $K = \lfloor N/2 \rfloor$ and $v_i = \max_{b_i \in B_i} b_i, \forall i \in [N]$. The runtime for each method in each setting is averaged over 100 experiments. The

results are depicted in Figure 1 with logarithmic y-axis, where data greater than 10^4 seconds are not displayed and the symbol “-S” (“-D”) denotes the same (different) score space group. Since the runtimes for TM in the two groups are almost the same, we just plot the average of them.

We see from Figure 1 that the runtime of TM increases exponentially as N grows. The 8×10 (i.e., $N = 8, M = 10$) game cannot be solved by TM within 1 hour. As a comparison, both AGG and AGGSU are much more efficient than TM. We observe that AGGSU-D (AGGSU-S) is about ten times faster than AGG-D (AGG-S), which confirms the efficiency of our speed-up methods proposed in Section 3.2.2. We further notice that AGGSU-S (AGG-S) always runs slower than AGGSU-D (AGG-D). That is because I_E is almost an empty set for AGGSU-D (AGG-D) and thus its AGs have $O(KNM)$ leaves, while the AGs for AGGSU-S (AGG-S) contain $O(KN^2M)$ leaves. Overall, our AGGSU method performs the best in all the settings.

Next we evaluate the performance of Algorithm 1. The parameters of the experiments are generated as above and λ is uniformly sampled from the support $(0, 10)^N$. We do not assume the identical score space and the set D is computed with AGGSU-D. The experiments are based on three different settings (game sizes). We use dynamic (Dy) and various fixed step lengths to test Algorithm 1. The results are depicted in Table 1, which shows the runtime of the algorithm in seconds (time), the number of Euler steps (#E), and the averaged amount of Gauss-Newton steps (#G) in one Euler step.

Table 1. Performance of Algorithm 1

Δ	$N=10, M=5$			$N=10, M=10$			$N=15, M=10$		
	time	#E	#G	time	#E	#G	time	#E	#G
Dy	0.55	5.10	2.40	5.66	8.00	2.44	19.5	5.42	2.38
0.1	1.92	26.0	0.73	18.6	47.0	0.76	82.1	29.1	0.68
0.3	0.85	10.9	1.03	8.58	18.1	1.05	37.8	11.2	1.03
0.5	0.82	7.45	1.71	7.67	12.3	1.83	33.0	8.03	1.37
0.7	0.81	6.76	2.40	7.68	10.7	2.32	26.8	5.45	2.44
0.9	0.82	5.44	2.60	7.92	8.12	3.56	30.1	4.96	2.75
1.1	0.85	4.34	3.25	10.8	10.2	3.91	37.7	5.02	3.03

We learn from Table 1 that larger step lengths usually lead to fewer Euler procedures, but more Gauss-Newton processes are needed to correct the zero point predicted in the Euler phase. The Gauss-Newton corrector often fails to converge when $\Delta \geq 1.3$, which implies that Newton-style methods cannot be directly used to compute the QRE. When Δ is small, the predicted zero points are often accurate enough and the corrector step is not needed, hence the averaged number of Gauss-Newton steps may be less than 1. Another interesting finding is that the numbers of steps for the predictor and corrector phases do not increase with the game size. We can verify that, given the numbers of bidders (N) and strategies (M), the runtime of Algorithm 1 is positively correlated to the total amount of steps and the dynamic step length strategy outperforms those strategies with fixed step lengths. The results indicate that our algorithm can efficiently compute the QRE for large-size SSAs.

5.2 Evaluate the Estimation Algorithm

We first used Algorithm 2 to infer the parameters of the QREs computed in Section 5.1 and found that the estimated parameters are always equal to the generated ones, which verifies the effectiveness of Algorithm 2 in parameter estimation. Next we conduct experiments based on Yahoo’s public data on advertising and market [14, 41, 47], which contain the information about advertisers’ bids and ranks over

4 months. More than 89% of the queries⁹ in the dataset have less than 5 bidders. We find that bidders’ information in many queries are very incomplete, i.e., there are only several records about a bidder over the 4 months. As in some related work like [13], we pick out 70 queries with almost complete information in the log (which do not include the queries containing just one bidder), and further remove the bidders who give a very high or very low bid and never make a change since these bidders will create singularity issues for the estimation and provide little information about bidders’ behaviors.

We fit the QRE model into the processed dataset, in which the distribution of the number of bidders is shown in Table 2. Specifically, for each query, we first compute bidders’ real mixed strategy profile σ with the log file, then we use Algorithm 2 to infer parameters with the QRE model, next by substituting the estimated parameters into Eq. (3), we compute bidders’ quantal responses π_i for all $i \in [N]$. We first evaluate whether π_i is equal to $\sigma_i, \forall i \in [N]$, for each query, i.e., whether advertisers’ bidding strategies (σ) form a QRE. To do this, we calculate the error $\frac{1}{\sum_{i \in [N]} |B_i|} \sum_{i \in [N], j \in |B_i|} |\pi_{ij} - \sigma_{ij}|$ for each query, based on which we compute the maximum, minimum and average of the errors over each of the four scenarios. The results are depicted in Table 2.

Table 2. Fitting Accuracy Evaluation

scenario	No. of bidders	distribution	maximum	minimum	average
1	3	77.14%	0.0732	0.0002	0.0373
2	4	15.71%	0.0813	0.0002	0.0388
3	5	4.29%	0.0922	0.0011	0.0432
4	≥ 6	2.86%	0.1078	0.0106	0.0592

We see from Table 2 that for some queries, the minimal errors are at the magnitude of 10^{-4} , and on average the errors are less than 0.06, which indicates that the QRE can model advertisers’ behaviors in the real world well. In most cases, the worst-case errors are around 0.08. Thus it is reasonable to assume that bidders were playing the QRE. Next, following the practice in [45], we use two specific queries to show details about the parameters estimated with QRE model and then make a comparison with the mixed strategy Nash equilibrium (MSNE) model¹⁰.

The parameters estimated with the QRE are depicted in Table 3, by substituting which into Eq. (3) we compute bidders’ quantal responses (π_i) and expected utilities (\bar{u}_i), as shown in Tables 4 and 5. We see that better strategies are chosen with higher probabilities, which is consistent with the principle of QRE. We find that the quantal responses (π_i) are very close to the real mixed strategies (σ_i) in both tables, which implies that the QRE model can accurately describe bidders’ bidding behaviors in the real world.

Table 3. Estimated parameters with QRE

query	i	v_i	λ_i	α_i	β	L^*
1	1	66.5	85	.04	.07	-3.2804
	2	43.0	1078	.03	.06	
	3	61.4	1602	.03	.05	
2	1	6.00	6251	.13	.15	-2.6098
	2	1.65	222	.65	.10	
	3	5.67	4700	.18	.05	

⁹ For simplicity, we only consider exact match between a query and keywords. For broad match, please refer to [9, 12, 31].

¹⁰ We do not compare the QRE with the symmetric Nash equilibrium [45] because the latter is a pure-strategy equilibrium which cannot explain advertisers’ mixed-strategy behaviors.

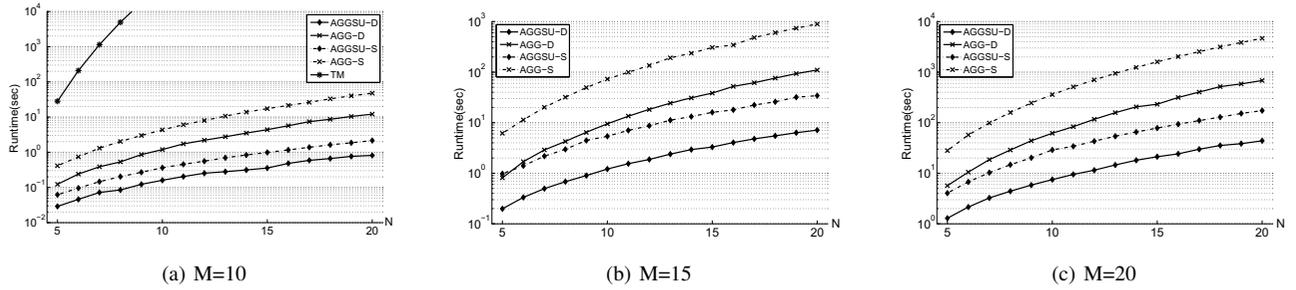


Figure 1. Runtime of TM, AGG and AGGSU

Table 4. Strategies comparison for query 1

σ_1	π_1	\bar{u}_1	σ_2	π_2	\bar{u}_2	σ_3	π_3	\bar{u}_3
.1261	.1239	.1202	.2975	.2979	.0479	.2609	.2607	.0906
.3025	.3563	.1328	.5950	.5948	.0485	.3565	.3697	.0908
.4202	.3714	.1333	.1074	.1073	.0469	.3826	.3697	.0908
.1513	.1484	.1224						

Table 5. Strategies comparison for query 2

σ_1	π_1	\bar{u}_1	σ_2	π_2	\bar{u}_2	σ_3	π_3	\bar{u}_3
.2927	.2682	.0339	.6897	.6881	.0633	.0522	.0524	.0581
.2439	.2682	.0339	.1466	.1040	.0548	.1304	.1304	.0583
.4634	.4635	.0340	.1207	.1040	.0548	.8174	.8172	.0587
			.0431	.1040	.0548			

We learn from Table 3 that bidders' precision parameters differ from each other significantly. Note that Eq. (3) implies that besides λ_i , the magnitude of the difference between bidder i 's expected utilities has a strong impact on his/her quantal response. To see this impact, we take bidder 1 in query 2 for an example. We know from Table 3 that $\lambda_1 = 6251$ for query 2, which seems to indicate that bidder 1 should be very rational because λ_1 is large. However, the quantal response $\pi_1 = (.2682, .2682, .4635)$ for query 2 in Table 5 indicates that bidder 1 is not very rational since he/she does not choose the optimal strategy with very high probability. That is because the difference between the components of \bar{u}_1 is at the magnitude of 10^{-4} , which reduces the effect of $\sum_{k \in [B_1]} e^{(\bar{u}_{1k}(\sigma_{-1}) - \bar{u}_{1j}(\sigma_{-1}))\lambda_1}$ in Eq. (3) even though λ_1 is at a magnitude of thousands.

Next we fit the MSNE into the dataset, where each player's expected utilities by choosing different pure strategies are the same. Hence we solve the following optimization problem with Algorithm 2:

$$\tilde{L}(v, \alpha, \beta | \sigma) = \log \left(\prod_{i \in [N]} \prod_{j \in [B_i]} \left(\frac{\bar{u}_{ij}(\sigma_{-i})}{\sum_{k \in [B_i]} \bar{u}_{ik}(\sigma_{-i})} \right)^{\frac{1}{|B_i|}} \right) \quad (16)$$

$$s.t. \begin{cases} v_i \geq 0, \forall i \in [N]; \\ 1 > \beta_s \geq \beta_{s+1} > 0, \forall s \in [K-1]; \\ 0 < \alpha_i < 1, \forall i \in [N]. \end{cases} \quad (17)$$

which get its maximal value when

$$\frac{\bar{u}_{ij}(\sigma_{-i})}{\sum_{k \in [B_i]} \bar{u}_{ik}(\sigma_{-i})} = \frac{1}{|B_i|}, \forall i \in [N], j \in [B_i],$$

or equivalently, when

$$\bar{u}_{ij}(\sigma_{-i}) = \bar{u}_{ik}(\sigma_{-i}), \forall i \in [N], j \in [B_i], k \in [B_i].$$

The estimated results are in Table 6. It shows that the maximal likelihoods of the queries with MSNE (\tilde{L}^*) are less than those with

QRE (L^*), which implies that QRE is more accurate than MSNE for modeling advertisers' pricing policies. We learn from the log that $B_1 = \{10, 15, 20, 30\}$, $B_2 = \{10, 20, 25\}$ and $B_3 = \{15, 25, 35\}$ for query 1. The values estimated by QRE are all larger than bids, which is consistent with the experience that bidders usually do not overbid [21]. As a comparison, the values predicted by MSNE are not very reasonable. Besides, the estimated CTRs (θ_{ik}) of MSNE for query 1 are overly large, e.g., $\alpha_1\beta_1 = 0.42$ and $\alpha_2\beta_1 = 0.76$, while the CTRs in the real world are generally lower than 10%. Furthermore, the log shows that the maximal bid in query 2 is not greater than 10, while the estimated values are at the magnitude of 10^6 , which does not make sense. The slot CTRs (β) estimated by MSNE for query 2 also seem strange since in the real world CTRs usually decrease from the top position to the bottom one. As a comparison, those estimated with QRE match the real world well. Overall, QRE can fit the real data much better than MSNE.

Table 6. Estimated parameters with MSNE

query	i	v_i	α_i	β	\tilde{L}^*
1	1	18.1	.51	.83	-3.5835
	2	12.8	.91	.02	
	3	112.2	.16	.02	
2	1	8.4×10^6	.02	.07	-4.7594
	2	4.8×10^6	.05	.07	
	3	3.08×10^6	.09	.07	

6 Conclusion

In this paper, we introduced the solution concept of QRE into S-SAs to model the bounded rationality of advertisers' bidding behaviors. Along this line, we made two key technical contributions. First, we designed an efficient homotopy-based algorithm to compute the QRE for SSAs. By further utilizing the special structure of advertisers' expected utilities, we significantly improved the efficiency of our algorithm which can be applied to large-size SSAs. Second, we developed an estimation algorithm and fitted the QRE model into real data to infer values, precisions and CTRs of the SSAs. In addition, we conducted extensive experiments to evaluate the performance of our algorithms, which show that the proposed homotopy algorithm for computing QRE is very efficient and the QRE model can fit the real data much better than previous models.

Acknowledgement

This paper is supported by 2015 Microsoft Research Asia Collaborative Research Program.

REFERENCES

- [1] Hirotugu Akaike, 'Information theory and an extension of the maximum likelihood principle', in *Selected Papers of Hirotugu Akaike*, 199–213, (1998).
- [2] Eugene L Allgower and Kurt Georg, *Introduction to numerical continuation methods*, volume 45, SIAM, 2003.
- [3] Eugene L Allgower and Kurt Georg, *Numerical continuation methods: an introduction*, volume 13, Springer Science & Business Media, 2012.
- [4] Paul E Black, *Dictionary of algorithms and data structures*, National Institute of Standards and Technology, 2004.
- [5] Tilman Börgers, Ingemar Cox, Martin Pesendorfer, and Vaclav Petricek, 'Equilibrium bids in sponsored search auctions: Theory and evidence', *American economic Journal: microeconomics*, **5**(4), 163–187, (2013).
- [6] Felix E Browder, 'On continuity of fixed points under deformations of continuous mappings', *Summa Brasiliensis Mathematicae*, **4**, 183–191, (1960).
- [7] Tian-Ming Bu, Xiaotie Deng, and Qi Qi, 'Forward looking nash equilibrium for keyword auction', *Information Processing Letters*, **105**(2), 41–46, (2008).
- [8] Tian-Ming Bu, Li Liang, and Qi Qi, 'On robustness of forward-looking in sponsored search auction', *Algorithmica*, **58**(4), 970–989, (2010).
- [9] Wei Chen, Di He, Tie-Yan Liu, Tao Qin, Yixin Tao, and Liwei Wang, 'Generalized second price auction with probabilistic broad match', in *Proceedings of the 15th ACM Conference on Economics and Computation*, pp. 39–56, (2014).
- [10] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou, 'The complexity of computing a nash equilibrium', *SIAM Journal on Computing*, **39**(1), 195–259, (2009).
- [11] Constantinos Daskalakis, Aranyak Mehta, and Christos Papadimitriou, 'Progress in approximate nash equilibria', in *Proceedings of the 8th ACM conference on Electronic commerce*, pp. 355–358, (2007).
- [12] Peerapong Dhangwatnotai, *Auction Design with Robust Guarantees*, Ph.D. dissertation, Stanford University, 2012.
- [13] Quang Duong and Sébastien Lahaie, 'Discrete choice models of bidder behavior in sponsored search', in *Internet and Network Economics*, 134–145, (2011).
- [14] Benjamin Edelman and Michael Ostrovsky, 'Strategic bidder behavior in sponsored search auctions', *Decision support systems*, **43**(1), 192–198, (2007).
- [15] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz, 'Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords', Technical report, National Bureau of Economic Research, (2005).
- [16] Nicola Gatti, Alessandro Lazaric, and Francesco Trovò, 'A truthful learning mechanism for contextual multi-slot sponsored search auctions with externalities', in *Proceedings of the 13th ACM Conference on Electronic Commerce*, pp. 605–622, (2012).
- [17] Jacob K Goeree, Charles A Holt, and Thomas R Palfrey, 'Regular quantal response equilibrium', *Experimental Economics*, **8**(4), 347–367, (2005).
- [18] Paul W Goldberg, Christos H Papadimitriou, and Rahul Savani, 'The complexity of the homotopy method, equilibrium selection, and lemke-howson solutions', *ACM Transactions on Economics and Computation*, **1**(2), 9, (2013).
- [19] Thore Graepel, Joaquin Q Candela, Thomas Borchert, and Ralf Herbrich, 'Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine', in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 13–20, (2010).
- [20] Philip A Haile, Ali Hortaçsu, and Grigory Kosenok, 'On the empirical content of quantal response equilibrium', *The American Economic Review*, **98**(1), 180–200, (2008).
- [21] Di He, Wei Chen, Liwei Wang, and Tie-Yan Liu, 'A game-theoretic machine learning approach for revenue maximization in sponsored search', in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pp. 206–212, (2013).
- [22] P Jean-Jacques Herings and Ronald Peeters, 'Homotopy methods to compute equilibria in game theory', *Economic Theory*, **42**(1), 119–156, (2010).
- [23] Bernard J Jansen and Tracy Mullen, 'Sponsored search: an overview of the concept, history, and technology', *International Journal of Electronic Business*, **6**(2), 114–131, (2008).
- [24] Albert Xin Jiang and Kevin Leyton-Brown, 'A polynomial-time algorithm for action-graph games', in *Proceedings of the 21st AAAI Conference on Artificial Intelligence*, pp. 679–684, (2006).
- [25] Albert Xin Jiang, Kevin Leyton-Brown, and Navin AR Bhat, 'Action-graph games', *Games and Economic Behavior*, **71**(1), 141–173, (2011).
- [26] Soren Johansen and Katarina Juselius, 'Maximum likelihood estimation and inference on cointegration with applications to the demand for money', *Oxford Bulletin of Economics and statistics*, **52**(2), 169–210, (1990).
- [27] Kenneth L Judd, *Numerical methods in economics*, MIT press, 1998.
- [28] R Bruce Kellogg, Tien-Yien Li, and James Yorke, 'A constructive proof of the brouwer fixed-point theorem and computational results', *SIAM Journal on Numerical Analysis*, **13**(4), 473–483, (1976).
- [29] Sébastien Lahaie, David M Pennock, Amin Saberi, and Rakesh V Vohra, 'Sponsored search auctions', *Algorithmic game theory*, 699–716, (2007).
- [30] Shijun Liao, 'On the homotopy analysis method for nonlinear problems', *Applied Mathematics and Computation*, **147**(2), 499–513, (2004).
- [31] Mohammad Mahdian and Grant Wang, 'Clustering-based bidding languages for sponsored search', in *Algorithms-ESA 2009*, 167–178, Springer, (2009).
- [32] Richard D McKelvey, Andrew M McLennan, and Theodore L Turocy, 'Gambit: Software tools for game theory, version 14.1.0. <http://www.gambit-project.org/>', (2014).
- [33] Richard D McKelvey and Thomas R Palfrey, 'Quantal response equilibria for normal form games', *Games and economic behavior*, **10**(1), 6–38, (1995).
- [34] Richard D McKelvey and Thomas R Palfrey, 'Quantal response equilibria for extensive form games', *Experimental economics*, **1**(1), 9–41, (1998).
- [35] Paul Milgrom, 'Simplified mechanisms with an application to sponsored-search auctions', *Games and Economic Behavior*, **70**(1), 62–70, (2010).
- [36] Garib N Murshudov, Alexei A Vagin, and Eleanor J Dodson, 'Refinement of macromolecular structures by the maximum-likelihood method', *Acta Crystallographica Section D: Biological Crystallography*, **53**(3), 240–255, (1997).
- [37] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani, *Algorithmic game theory*, volume 1, Cambridge University Press Cambridge, 2007.
- [38] Tao Qin, Wei Chen, and Tie-Yan Liu, 'Sponsored search auctions: Recent advances and future directions', *ACM Transactions on Intelligent Systems and Technology (TIST)*, **5**(4), 60, (2015).
- [39] Jiang Rong, Tao Qin, and Bo An, 'Computing quantal response equilibrium for sponsored search auctions', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1803–1804, (2015).
- [40] Jiang Rong, Tao Qin, Bo An, and Tie-Yan Liu, 'Optimal sample size for adword auctions', in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp. 1459–1460, (2016).
- [41] Savannah Wei Shi and Xiaojing Dong, 'The effects of bid pulsing on keyword performance in search engines', *International Journal of Electronic Commerce*, **19**(2), 3–38, (2015).
- [42] David Roger Smart, *Fixed point theorems*, volume 66, CUP Archive, 1980.
- [43] David Robert Martin Thompson and Kevin Leyton-Brown, 'Computational analysis of perfect-information position auctions', in *Proceedings of the 10th ACM conference on Electronic commerce*, pp. 51–60, (2009).
- [44] Theodore L Turocy, 'A dynamic homotopy interpretation of the logistic quantal response equilibrium correspondence', *Games and Economic Behavior*, **51**(2), 243–263, (2005).
- [45] Hal R Varian, 'Position auctions', *International Journal of Industrial Organization*, **25**(6), 1163–1178, (2007).
- [46] Haifeng Xu, Bin Gao, Diyi Yang, and Tie-Yan Liu, 'Predicting advertiser bidding behaviors in sponsored search by rationality modeling', in *Proceedings of the 22nd international conference on World Wide Web*, pp. 1433–1444, (2013).
- [47] Yahoo, 'Webscope for advertising and market data. A3: ydata-ysm-keyphrase-bid-imp-click-v1.0. <http://webscope.sandbox.yahoo.com/>', (2013).

An Improved State Filter Algorithm for SIR Epidemic Forecasting

Weipeng Huang¹ and Gregory Provan¹

Abstract. In epidemic modeling, state filtering is an excellent tool for enhancing the performance of traditional epidemic models. We introduce a novel state filter algorithm to further improve the performance of state-of-the-art approaches based on Susceptible-Infected-Recovered (SIR) models. The proposed algorithm merges two techniques, which are typically used separately: linear correction, as seen in the Ensemble Kalman Filter (EnKF), and resampling, as used in the Particle Filter (PF). We compare the inferential accuracy of our approach against the EnKF and the Ensemble Adjustment Kalman Filter (EAKF), using algorithms employing both an uncentered covariance matrix (UCM) and the standard column-centered covariance matrix (CCM). Our algorithm requires $\mathcal{O}(DN)$ more time than EnKF does, where D is the ensemble dimension and N denotes the ensemble size. We demonstrate empirically that our algorithm with UCM achieves the lowest root-mean-square-error (RMSE) and the highest correlation coefficient (CORR) amongst the selected methods, in 11 out of 14 major real-world scenarios. We show that the EnKF with UCM outperforms the EnKF with CCM, while the EAKF gains better accuracy with CCM in most scenarios.

1 Introduction

Epidemic prediction has a long history, and an early model SIR model [27] has proved essential for accurate forecasting [4, 28]. The SIR model divides the population into three sub-populations: susceptible (S), infected (I) and recovered (R). During the outbreak of an infectious disease, some susceptible individuals will become infected by contact with the infected individuals, and some infected individuals will recover within a certain period of time.

Recently, Shaman and Karspeck, and Yang et al. [34, 44] showed that state filtering methods significantly improve the inferential accuracy of the SIRS-humidity model (which is a variant of SIR) [4]. Yang et al. also empirically demonstrated that EnKF, EAKF and PF have the lowest RMSE [44]. Although the difference in the performance of these three filters is small, EAKF comes out on top, while PF is in bottom place. They examine the model performance concerning 115 cities in the United States (U.S.), using only Google Flu Trends (GFT) data [16].

The EnKF is a Monte Carlo approximation of the Kalman filter, which represents the distribution of the system state using a collection of state vectors, called an ensemble, and replace the KF covariance matrix by the sample covariance computed from the ensemble. The EnKF assumes Gaussian-distributed models, while the PF does not impose that restriction. However, Kalman-type filters re-

quire fewer ensemble members (or particles) than the PF to guarantee good performance [36]. Moreover, the EnKF applies linear correction updates to the states to satisfy the Maximum Likelihood; the PF updates its ensemble members from the existing particles by sampling from their weight distributions, where the weights are assigned by the ensemble members' importance (a.k.a importance weighting) [12, 10, 30]. Our algorithm integrates all these techniques, correcting the states with Maximum Likelihood, and updating the ensemble by sampling from the best-performing particles. The algorithm imposes only slight additional time complexity to the EnKF; however, it requires the same ensemble size as the EnKF, provided the improved performance over the two single filters is attained.

[43] shows that the EnKF underestimates the state covariance matrix. Therefore, we compare the model accuracy of the Kalman-type filters empirically, with UCM and CCM. UCM [5] is mostly discussed in Principal Component Analysis (PCA), hence, our use of UCM in the EnKF and the EAKF is novel. Centering the data or keeping it uncentered remains an open question in pattern recognition [20]. There are only two theoretical papers [5, 20] analyzing UCM and CCM, and they both performed eigen-analysis of certain features of both types. We empirically compare these approaches using real-world infection data [7] from the U.S. Center for Disease Control and Prevention (CDC), which contains weekly influenza-like illnesses (ILI) statistics.

Our contributions are as follows. We compare our state filter algorithms with state-of-the-art filters on the nationwide ILI data of 2011-15 and the regional ILI data of 2014-15. The empirical results demonstrate that our approach obtains the optimal RMSE and CORR amongst the examined filters, in 11 out of 14 cases. It also shows that the EnKF and our approach create more accurate predictions with UCM rather than CCM, whereas the EAKF gains better performance with CCM, given the tested scenarios.

In the rest of the paper, Sect. 2 reviews the related work. Sect. 3 elaborates on the models. Following that, Sect. 4 discusses our approach, and then conducts the empirical analysis in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Related Work

Kermack and Mckendrick [27] introduced a key early model for epidemic forecasting, the SIR model. Later, several models were derived from it, such as SI, SIRS, SEIR, SIS, etc. [4, 19]. Researchers have also developed other types of models for computational epidemiology, e.g., agent-based models, meta-population models, spatial models, and stochastic models [35].

Recently, data-driven solutions have shown great promise. Ginsberg et al. [15] used Google search data to build a logistic regression

¹ The Insight Centre for Data Analytics, Department of Computer Science, University College Cork, Cork, Ireland
Email: weipeng.huang@insight-centre.org, gprovan@cs.ucc.ie

model based on the odds ratio of the search-term frequency. Santilana et al. [33, 32] then used the term frequency directly with several machine learning techniques, such as Support Vector Machine, Least Absolute Shrinkage and Selection Operator, and AdaBoost regression etc., to outperform Google’s solution. These innovative methods obtain great performance, but at a high computational cost.

Shaman and Karspeck [34] applied the EAKF to the SIRS-humidity model (SIRS-EAKF). SIRS-humidity adds the humidity data as a component to the standard SIRS model, given a correlation between the spread of the epidemic and the humidity levels. Yang et al. [44] then compared a few filtering methods, such as EnKF, EAKF, PF, Maximum Likelihood Filter etc., on the SIRS model. They reported that the EAKF, the EnKF and the PF were the top three performers. Later, [45] estimated a few SIRS-EAKF epidemiological parameter ranges for the seasonal flu and pandemics for a few seasons. We compare the EnKF and the EAKF, with CCM and UCM, while [44] only tested on the generic cEnKF and cEAKF².

There are related works connecting the EnKF and the PF. Hoteit et al. [22] introduced the method of combing Kalman correction and resampling, and later simplified the algorithm by removing the unnecessarily complex steps in the resampling circle [21]. [41] then extended Hoteit’s method to mixture Gaussian models. Different from the above principle, [14, 9] suggested a strategy that adopts the weighted sum of the posterior states propagated with the EnKF and the PF. Slivinski et al. presented a hybrid filter EnKF-PF for Lagrangian data assimilation [40, 39]. The most complex step in most methods is the covariance matrix approximation; however we performed a comparison between CCM and UCM for deciding the simplest covariance approximation procedure. Our algorithm thereby addresses the overwhelming complexity in the existing approaches.

3 Models

3.1 Notation

In SIR modeling, the population has three sub-categories: susceptible (S), infected (I), and recovered (R). Given the total population M , the percentages of the three sub-groups are $\{s, i, r\}$, where $s = S/M$, likewise for i and r . Lastly, β and γ denote the mean contact rate and the mean recovery rate, respectively.

For the filtering approaches, we first denote the state vector by $x = [s \ i]^T$. We denote the one-element estimate vector by $y = [\hat{i}]$, the observation by $z = [\hat{z}]$, and parameter $\theta = [\beta \ \gamma]^T$. We use a transition function $f(\cdot)$, and the state to observation mapping matrix, H , to define the following dynamical state space system

$$x_{t+1} = f(x_t, \theta_t) + u_t \quad (1a)$$

$$y_{t+1} = Hx_{t+1} + v_t \quad (1b)$$

Moreover, we denote the observed data z_t for time t . In our formulation, $f(\cdot)$ is governed by the SIR dynamics and $H = [0 \ 1]$.

Let \sim denote “distributed according to”; henceforth we assume the noise is zero-mean Gaussian such that $u \sim \mathcal{N}(0, U)$ and $v \sim \mathcal{N}(0, V)$. We define an ensemble as a group of particles, where a particle is a random sample from a certain distribution. The N -ensemble of states, estimates, and parameters are respectively depicted as

$$X_t = \{x_t^{(n)}\}^N \quad Y_t = \{y_t^{(n)}\}^N \quad \Theta_t = \{\theta_t^{(n)}\}^N.$$

² We add “c” in front of the filter names to indicate the filters using CCM, and use prefix “u” for those using UCM.

Hence, the ensemble version of the dynamical system is:

$$X_{t+1} = f(X_t, \Theta_t) + \{u_t^{(n)}\}^N \quad (2a)$$

$$Y_{t+1} = HX_{t+1} + \{v_t^{(n)}\}^N \quad (2b)$$

We denote the weights of the ensemble members, by $w_t = [w_t^{(1)} \ \dots \ w_t^{(N)}]^T$. Therefore, for the ensemble or particle based methods, \hat{i}_t is the expected value of the infection rate (a.k.a. prevalence) at time t , such that

$$y_t = [\hat{i}_t] = HE[X_t] \quad (3)$$

where $E[X]$ returns the mean of X . Let I denote the identity matrix and \propto denote “proportional to”. $0_{D \times N}$ refers to a D -by- N matrix of zeros, and 1_N is a length- N vector of ones.

Parameter estimation with KFs. The parameter estimation with the EnKF and the EAKF proceeds by regarding the parameters as augmented states [29, 23, 13, 1]. Specifically, we denote the refined state vector by \tilde{x} and its corresponding ensemble set \tilde{X} , such that

$$\tilde{x} = \begin{bmatrix} x \\ \theta \end{bmatrix} \implies \tilde{X} = \begin{bmatrix} X \\ \Theta \end{bmatrix} \quad (4)$$

Given Eq. (1) and (2), we get $\tilde{H} = [0 \ 1 \ 0 \ 0]$. In implementing the KFs, we replace x , X and H by \tilde{x} , \tilde{X} and \tilde{H} , respectively. In the PF, we use the original x , X and H .

3.2 Susceptible-Infected-Recovered

We select the version of the SIR [28] that uses the ratios s , i , and r . As SIR assumes that the birth and death are negligible to the whole population during a period, the population M is constant and $s_t + i_t + r_t \equiv 1$ holds at any time within a particular period. The model depicts the dynamics by assuming the susceptible individuals become infected with probability β , and infected individuals can recover from the disease with recovery rate γ .

$$\frac{\partial s}{\partial t} = \beta s_t i_t \quad \frac{\partial i}{\partial t} = \beta s_t i_t - \gamma i_t \quad \frac{\partial r}{\partial t} = \gamma i_t \quad (5)$$

Apparently, r does not contribute to computing the prevalence i . We thus only present s and i in the state vector x , and omit the equations related to r in the rest of this paper.

3.3 Kalman Filter

The KF is a method that computes the posterior states based on the Maximum Likelihood of a linear Gaussian dynamical system [8, 36, 30]. During each KF round, Eq. (6) and (7) execute a prediction phase, and Eq. (8) to (10) run a correction phase. P denotes the covariance of the states, and K the Kalman Gain matrix. Also, we use a transition matrix B to approximate the transition function $f(\cdot)$. We thus obtain

$$x_{t|t-1} = Bx_{t-1|t-1} \quad (6)$$

$$P_{t|t-1} = BP_{t-1|t-1}B^T + U \quad (7)$$

$$K_t = P_{t|t-1}H^T (HP_{t|t-1}H^T + V)^{-1} \quad (8)$$

$$x_{t|t} = x_{t|t-1} + K_t(z_t - Hx_{t|t-1}) \quad (9)$$

$$P_{t|t} = (I - K_tH)P_{t|t-1} \quad (10)$$

The critical step of a KF is to compute the Kalman Gain K , then combine the observed data to calibrate the state vector x . The KF assumes that the *posteriori* $x_{t|t}$ is more probable as the input for the next estimation than the *priori* $x_{t|t-1}$. Eq. (9) is known as a Kalman-type (or linear) correction step.

3.3.1 Ensemble Kalman Filter

The KF is an optimal filter for linear Gaussian systems with Gaussian noise [8]. For nonlinear systems, researchers have developed the Extended Kalman Filter, the Unscented Kalman Filter, the EnKF and the EAKF, etc. [11, 36, 30]. We consider the EnKF and the EAKF as they require less parameter tuning than the other Kalman Filters.

The EnKF estimates the covariance matrix P , through the sample covariance of the ensemble [11, 25, 12, 13]. Therefore, Eq. (7) and (10) are not used in the EnKF. With the sample covariance denoted by C , we have the Kalman Gain K :

$$K_t = C_{t|t-1} H^T (H C_{t|t-1} H^T + V)^{-1} \quad (11)$$

We now show the UCM C_c and the CCM C_u are computed using:

$$C_u = \frac{1}{N-1} X X^T \quad (12a)$$

$$C_c = \frac{1}{N-1} (X - \bar{x} \mathbf{1}_N^T) (X - \bar{x} \mathbf{1}_N^T)^T, \quad (12b)$$

where the mean vector $\bar{x} = \frac{1}{N} X \mathbf{1}_N$. The CCM is the UCM of the ensemble after being centered. It follows that:

$$\begin{aligned} C_c &= \frac{1}{N-1} (X X^T - \bar{x} \mathbf{1}_N^T X^T - X \mathbf{1}_N \bar{x}^T + \bar{x} \mathbf{1}_N^T \mathbf{1}_N \bar{x}^T) \\ &= \frac{1}{N-1} X X^T - \frac{N}{N-1} \bar{x} \bar{x}^T \\ &= C_u - \frac{N}{N-1} \bar{x} \bar{x}^T \end{aligned} \quad (13)$$

Given any $X > 0_{D \times N}$, C_c is strictly smaller than C_u . The sample covariance matrix C_u and C_c both approach the corresponding population covariance matrix asymptotically as N grows, as $\lim_{N \rightarrow \infty} N-1 = N$.

Finally, the EnKF executes the correction as in Eq. (9) to update every prior state particle $x_{t|t-1}^{(n)}$ to the posterior state $x_{t|t}^{(n)}$.

3.3.2 Ensemble Adjustment Kalman Filter

The EAKF adds one more step at each round to improve the EnKF [2, 26]. This filter runs an EnKF round, and then employs a matrix A to further correct the ensemble members such that

$$\hat{x}_{t|t}^{(n)} = A^T (x_{t|t-1}^{(n)} - \bar{x}_{t|t-1}) + \bar{x}_{t|t} \quad n = 1 \dots N \quad (14)$$

Anderson [2] stated that a number of values for A exist, raising a new problem of choosing A . [34, 44] used $A = 1.03I$ for the GFT data they examined. The research to date mostly selects A based on empirical tests [2, 34, 44, 45]. Ensemble adjustment in the EAKF is superior to the EnKF in preventing the filter divergence caused by the dubiously small prior covariances.

3.4 Particle Filter

A PF [3, 10, 30] is a sequential Monte Carlo method that can perform filtering for arbitrary models. It employs sequential importance sampling and resampling to draw samples from certain distributions, in order to approximate the “true” state variables by a weighted mean that satisfies

$$E[X_t] = \sum_{n=1}^N w_t^{(n)} x_t^{(n)}. \quad (15)$$

These Monte Carlo methods approximate the true distribution of a state through sampling from a *proposal distributions*. For the simulations, we implement Storvik’s PF algorithm [42], instead of the generic PF. At each timestamp, Storvik’s PF samples θ_t and x_t from the *proposal distributions* $q_\theta(\theta_t^{(n)} | x_{t-1}^{(n)}, z_t)$ and $q_x(x_t^{(n)} | x_{t-1}^{(n)}, z_t, \theta_t^{(n)})$ in sequence. Hence, it normalizes the weights such that for every n ,

$$w_t^{(n)} \propto w_{t-1}^{(n)} \frac{p(\theta_t | s_t^{(n)}) p(z_t | x_t^{(n)}, \theta_t^{(n)}) p(x_t^{(n)} | x_{t-1}^{(n)}, \theta_t^{(n)})}{q_\theta(\theta_t^{(n)} | x_{t-1}^{(n)}, z_t) q_x(x_t^{(n)} | x_{t-1}^{(n)}, z_t, \theta_t^{(n)})} \quad (16)$$

where s_t refers to the sufficient statistics for the parameters in the distribution. A sufficient statistic for an unknown parameter in a distribution, is the statistic that provides sufficient information for deciding that parameter. We approximate the required distributions by assuming some known distribution (e.g., Gaussian) rather than using the Markov Chain Monte Carlo, since [24, 6] suggested that a PF with appropriate assumption of distributions can yield better accuracy and far better computing efficiency.

The PF suffers from *degeneracy*, where the significant weights are occupied by a minor portion of the particles [3, 30]. It then uses the quantity of the *effective sample size* S_{eff} to control the resampling switch, where

$$S_{eff} := \frac{N}{1 + \text{Var}[w_t]} \approx \left[\sum_{n=1}^N (w_t^{(n)})^2 \right]^{-1}$$

If S_{eff} is smaller than a certain threshold, it is thought to be suffering from *degeneracy*. In such a case, the PF resamples X_t indirectly by sampling the indices of the states according to the weight distribution w_t . After resampling, all weights will be reset to N^{-1} .

4 Proposed State Filter

Our approach, ensemble adjustment using resampling (BASS), incorporates the Kalman-type (linear) correction, resampling and importance weighting, which prunes the worst-performing particles and weight the particles after every Kalman correction. The resampling helps the ensemble members converge more quickly to the true posterior distributions. The linear correction reduces the ensemble size, and makes the process more tractable. BASS ideally retains a sufficiently large proportion of the states, and the information of them. That is, we conduct partial resampling, in which the particles with negligible weight are replaced by those with large weight. It then naturally protects the process from *degeneracy* if there are sufficiently many ensemble members performing well. To mitigate against *sample impoverishment*, i.e., the loss of diversity amongst the ensemble population [3, 30], we also introduce noise when resampling the states.

4.1 BASS

BASS integrates the EnKF and the PF, and focuses on the state correction step. The correction phase prepares the posterior particles $X_{t|t}$ as input for the next prediction. Since the prior particles $X_{t|t-1}$ (e.g., generated by the SIR model) and the observations z_t are collected, it first runs one EnKF correction and fetches $X_{t|t}$. Next, it proceeds with weighting and partial resampling on $X_{t|t}$, to update $X_{t|t}$ and fetch the weights w_t . The forecasting procedure is shown in Algorithm 1. The EnKF execution is contained in the algorithm BASS (Algorithm 2).

Algorithm 1: FORECASTING($X_{0|0}$, ϵ , H)

```

1 Initialization  $w_0 \leftarrow \{N^{-1}\}^N$ 
2 for  $t \leftarrow 1 \dots T$  do  $\triangleright T \leftarrow \infty$  for continuous forecasting
3    $X_{t|t-1} \leftarrow \text{SIR}(X_{t-1|t-1})$ 
4    $[i_t] \leftarrow H \sum_{n=1}^N w_{t-1}^{(n)} x_{t|t-1}^{(n)}$   $\triangleright$  the prediction
5   if  $t \neq T - 1$  then
6      $z_t$  streams in
7      $(X_{t|t}, w_t) \leftarrow \text{BASS}(X_{t|t-1}, w_{t-1}, z_t, \epsilon)$ 

```

Compared with full resampling, partial resampling decreases the computational costs and removes particles with small weights. Partial resampling in BASS uses a global threshold variable, ϵ , and a weight score variable $w_t^{(n)}$ for each particle n . More specifically, the weight represents the normalized likelihood $w_t^{(n)} = p(z_{1:t} | x_{1:t}^{(n)})$. If the particle's weight score is less than the threshold ϵ , we replace it with a randomly picked existing particle with large weight. Consider the system in a Hidden Markov representation, we have

$$\begin{aligned}
p(z_{1:t} | x_{1:t}^{(n)}) &= p(z_{1:t-1} | x_{1:t-1}^{(n)}) p(z_t | x_{1:t}^{(n)}) \\
&= p(z_{1:t-1} | x_{1:t-1}^{(n)}) p(z_t | x_t^{(n)}) \\
&\propto w_{t-1}^{(n)} p(z_t | x_t^{(n)})
\end{aligned}$$

The likelihood also coincides with the chained performance of the particular particle. From existing research [34, 44], we know that there must be state samples that consistently forecast well. Thus, the chained performance over time can be used for filtering out the worst-performing (with small likelihood) particles. Every newly resampled particle inherits the weight score from the root particle. As the weight is also the chained performance, we do not reset the weights back to $\{N^{-1}\}^N$ as in the PF, thus keeping the historical information of the particles.

BASS is detailed in Algorithm 2. The weights are initialized to a uniform one-sum vector. First, line 2 executes one EnKF execution and returns the posterior states, by $\text{ENKF}(\cdot)$. The normalization in line 5 prevents the likelihood from tending towards 0 as time grows. $\text{NORM}(\cdot)$ takes a non-negative weight vector and returns a normalized one-sum weight vector, such that the sum of the weights divides each weight. The particles fitting to the observations satisfactorily survive. Hence, we call E (in line 6) the non-survivor set, and w^s (in line 7) the survivors weight set. Normalizing the survivors' weights (in line 8) guarantees the under-performing particles are all replaced. Line 6 to 11 present the partial resampling procedure. It splits the particles into survivors and non-survivors depending on the threshold ϵ , hence it resamples the particles from the survivors (according to their weights) to replace the non-survivors.

Algorithm 2: BASS($X_{t|t-1}$, w_{t-1} , z_t , ϵ)

```

1 Function BASS( $X_{t|t-1}$ ,  $w_{t-1}$ ,  $z_t$ ,  $\epsilon$ )
2    $X_{t|t} \leftarrow \text{ENKF}(X_{t|t-1}, z_t)$ 
3   for  $n \leftarrow 1 \dots N$  do
4      $w_t^{(n)} \leftarrow w_{t-1}^{(n)} p(z_t | x_{t|t}^{(n)})$ 
5    $w_t \leftarrow \text{NORM}(w_t)$ 
6    $E \leftarrow \{n : w_{t-1}^{(n)} < \epsilon, n = 1 \dots N\}$ 
7    $w^s \leftarrow \{w_{t-1}^{(n)} : w_{t-1}^{(n)} \geq \epsilon, n = 1 \dots N\}$ 
8    $w^s \leftarrow \text{NORM}(w^s)$ 
9   Sample  $|E|$  indices  $G \sim w^s$   $\triangleright |E|$  returns the size of  $E$ 
10   $w_t^{(E)} \leftarrow w_t^{(G)}$ 
11   $X_{t|t}^{(E)} \leftarrow X_{t|t}^{(G)} + \{u_t^{(g)}\}^G$ 
12   $w_t \leftarrow \text{NORM}(w_t)$ 
13  return  $(X_{t|t}, w_t)$ 

```

4.1.1 Time Complexity

BASS consists of the EnKF and the supplement (resampling and weighting). The time of the supplement for each iteration is in $\mathcal{O}(DN)$, where D is the state dimension and N is the ensemble quantity. In the algorithm, three normalizations in line 5, 8 and 12, force $\mathcal{O}(3 \times 2N)$ steps. Checking and computing the likelihood consumes time in $\mathcal{O}(N)$, from line 3 to 4. Next, drawing the survivors sets also takes time in $\mathcal{O}(N)$, in line 6 and 7. Ideally, resampling (from line 9 to 11) is just for a small portion of the particles, while the worst case costs the time $\mathcal{O}(2N + 2DN)$. Given the assumption D is at least close to 5, the extra time is bounded by $\mathcal{O}(6N + N + N + 2N + 2DN) = \mathcal{O}(DN)$ at each round.

5 Empirical Study

5.1 Experimental Setup

The ILI data [7] records the weekly infection statistics for the U.S., both nationwide and regionally. We select the data of 2011-15, and the 10 regions in 2014-15. Our simulations focus on the forecasts of the infection rate, for the national cases (4 cases) and the regional cases (10 cases) separately. We mainly focus on the RMSE between our predictions and the observations, and also present their CORR.

Initialization. Every initial infection percentage $i_0^{(n)}$, for the n -th particle, is sampled from a uniform distribution $\mathcal{U}(0, b)$, in which b approximately doubles the first observation of the ILI data z , such that the population mean $\mu = \frac{a+b}{2} \approx z_0$. Given $s_0^{(n)} + i_0^{(n)} + r_0^{(n)} = 1$ and $r_0^{(n)} = 0$ for the particle n , we have $s_0^{(n)} = 1 - i_0^{(n)}$, $n = 1 \dots N$. Hence, β and γ are sampled from $\mathcal{U}(0, 1)$. The process will resample the state vector when $\beta \leq \gamma$ is detected. The reproduction number is given by $R_0 = \beta/\gamma$, and $R_0 \leq 1$ means there would not be an epidemic outbreak. With respect to the Kalman Filters, we set the noise $r \sim \mathcal{N}(0, 10^{-4}I)$ and $s \sim \mathcal{N}(0, 10^{-4})$. For the EAKF, we pick $A = 1.001I$ for A in Eq. (14). For uBass and cBass, we find that the resampling threshold $\epsilon = 10^{-5}$ is rather robust for all cases. For the PF solution, given Eq. (16), we select the *proposal distributions* according to the setting:

$$\begin{aligned}
q_\theta(\theta_t^{(n)} | x_{t-1}^{(n)}, z_t) &= p(\theta_t | s_t^{(n)}) \\
q_x(x_t^{(n)} | x_{t-1}^{(n)}, z_t, \theta_t^{(n)}) &= p(x_t^{(n)} | x_{t-1}^{(n)}, \theta_t^{(n)})
\end{aligned}$$

Table 1. Mean RMSE% for the methodologies, with the 0.99 confidence interval within the parentheses. Every approach is repeated 50 times. In each tested situation, the best performer is labeled with bold face.

	2011-12	2012-13	2013-14	2014-15	
cBass	0.202 (0.200, 0.204)	0.469 (0.458, 0.480)	0.376 (0.369, 0.383)	0.594 (0.580, 0.608)	
cEAKF	0.651 (0.637, 0.664)	0.510 (0.498, 0.522)	0.522 (0.508, 0.536)	0.520 (0.512, 0.528)	
cEnKF	0.391 (0.391, 0.391)	1.373 (1.253, 1.494)	1.053 (0.990, 1.115)	1.372 (1.243, 1.501)	
PF	0.449 (0.413, 0.486)	0.689 (0.657, 0.721)	0.569 (0.523, 0.615)	0.677 (0.641, 0.714)	
uBass	0.163 (0.158, 0.168)	0.406 (0.395, 0.417)	0.269 (0.260, 0.279)	0.427 (0.415, 0.440)	
uEAKF	0.562 (0.550, 0.574)	1.005 (0.989, 1.020)	0.847 (0.834, 0.860)	0.925 (0.910, 0.939)	
uEnKF	0.146 (0.145, 0.146)	0.417 (0.417, 0.417)	0.295 (0.295, 0.296)	0.446 (0.446, 0.447)	
	2014-15 Region 1	2014-15 Region 2	2014-15 Region 3	2014-15 Region 4	2014-15 Region 5
cBass	0.351 (0.337, 0.365)	0.339 (0.330, 0.348)	0.898 (0.873, 0.922)	0.861 (0.833, 0.889)	0.630 (0.609, 0.651)
cEAKF	0.531 (0.520, 0.543)	0.610 (0.597, 0.623)	0.855 (0.845, 0.866)	0.734 (0.723, 0.744)	0.715 (0.697, 0.733)
cEnKF	0.955 (0.955, 0.956)	1.757 (1.732, 1.782)	1.897 (1.897, 1.898)	1.696 (1.695, 1.696)	2.207 (2.132, 2.283)
PF	0.855 (0.677, 1.033)	0.607 (0.569, 0.646)	1.325 (1.259, 1.392)	1.046 (0.825, 1.267)	1.589 (1.353, 1.825)
uBass	0.276 (0.269, 0.283)	0.267 (0.260, 0.274)	0.809 (0.793, 0.825)	0.718 (0.702, 0.735)	0.467 (0.453, 0.481)
uEAKF	0.845 (0.833, 0.858)	0.797 (0.781, 0.812)	1.218 (1.206, 1.230)	1.219 (1.204, 1.234)	1.116 (1.102, 1.129)
uEnKF	0.445 (0.441, 0.449)	0.986 (0.934, 1.037)	1.740 (1.660, 1.821)	1.567 (1.392, 1.743)	0.874 (0.779, 0.969)
	2014-15 Region 6	2014-15 Region 7	2014-15 Region 8	2014-15 Region 9	2014-15 Region 10
cBass	0.639 (0.622, 0.655)	0.462 (0.445, 0.478)	0.462 (0.443, 0.481)	0.395 (0.386, 0.403)	0.436 (0.423, 0.449)
cEAKF	0.506 (0.499, 0.513)	0.455 (0.445, 0.466)	0.477 (0.462, 0.492)	0.617 (0.607, 0.628)	0.402 (0.397, 0.407)
cEnKF	2.690 (2.670, 2.719)	1.473 (1.434, 1.512)	1.096 (1.096, 1.096)	1.157 (1.156, 1.159)	1.086 (1.086, 1.086)
PF	0.934 (0.880, 0.987)	2.012 (1.649, 2.376)	0.818 (0.620, 1.016)	0.681 (0.650, 0.711)	1.138 (0.915, 1.361)
uBass	0.601 (0.556, 0.647)	0.472 (0.461, 0.483)	0.331 (0.320, 0.342)	0.341 (0.330, 0.351)	0.374 (0.366, 0.381)
uEAKF	1.100 (1.086, 1.114)	1.106 (1.094, 1.119)	0.985 (0.969, 1.002)	0.867 (0.848, 0.876)	0.808 (0.792, 0.820)
uEnKF	3.111 (2.171, 4.050)	1.054 (1.039, 1.069)	0.742 (0.721, 0.763)	1.155 (1.103, 1.207)	0.824 (0.795, 0.852)

We also assume that $p(x_t^{(n)} | z_t)$ is distributed by Gaussian. The sufficient statistics for the Gaussian $\mathcal{N}(\mu, \sigma^2)$ are the sample mean for μ and sample variance for σ^2 , or sample covariance matrix for Σ in the multivariate Gaussian $\mathcal{N}(\mu, \Sigma)$. Finally, let $S_{eff} = N/2$.

Implementation. The program is developed in Python, and is available at <https://github.com/weipeng/pyepi>.

5.2 Discussion

5.2.1 Performance Result

For this task, we find that the CORR is strongly correlated to the RMSE and therefore we focus on the RMSE. We average our results over 50 for each case. In each season and region, we carry out the Analysis of Variance (ANOVA) on the mean RMSE, with the null hypothesis that all methods gain equal RMSE. The data show that for all the scenarios, the null hypothesis is rejected with the P-values all less than $2e-16$. A statistical comparison is thought to be significant when its P-value is less than 0.05. We apply the Tukey test to conduct pairwise comparisons of the approaches. The Tukey test is commonly thought to be better than the pairwise t-test, as it embeds the protection to the rise in the risk of Type I error. The statistical analysis is carried out through the built-in functions in the statistical computing language R [31].

Mean RMSE. The mean RMSE result is shown in Table 1 and the Tukey result involving the Bass solutions is shown in Table 2. From 2011-15, uBass, uEnKF and cBass are the top 3 performers in order. The algorithm uBass achieves the optimal prediction accuracy in 3 seasons (2012-15), while uEnKF achieves the optimal accuracy for 2011-12. Also, cBass is the third best performer in the nationwide cases, gaining the third best performance in 3 cases. In 10 regional

cases, uBass, cBass and cEAKF are the top three performers in order. The mean RMSE of uBass is significantly lower than that of both cBass and cEAKF in 5 regions, however, demonstrates better accuracy than both cBass and cEAKF in 4 regions (amongst the other 5 regions) although the differences between them are not significant. Our cBass achieves significantly lower RMSE than cEAKF in 2 regions, and lower (but not significantly) RMSE in 7 regions.

RMSE of the Mean Estimates. Fig. 1 displays the RMSE and the CORR of the mean estimates over 50 repetitions. In the RMSE heatmap, the lighter color implies better accuracy. In the CORR heatmap, the darker color implies higher correlation, and red indicates the positive direction while blue indicates the negative. In view of such a situation, uBass gains both the optimal RMSE and CORR in 11 out of 14 cases. Both heat-maps demonstrate uBass is clearly optimal, and cBass is the third across all cases. In conclusion, uBass and cBass consistently perform better in predicting the seasonal influenza level in the U.S.

Confidence Interval. Table 1 exhibits the mean RMSE, with the 99% confidence intervals. Our uBass approach consistently obtains high accuracy, and maintains a small/tight confidence interval gap ($\text{gap} < 0.03\%$), except for region 6. Besides, cBass yields the gaps smaller than 0.03% in 13 situations, and cEAKF achieves the tight gaps in all situations. Notice that, uEnKF has the smallest (compared with others) interval gaps for the 4 nationwide predictions, but gets large uncertainty in the regional predictions of 8 regions. However, as a derivation from uEnKF, uBass overcomes the adversity in the regional forecasting simulations.

Fig. 2 illustrates that most approaches have tight confidence intervals even at the 99% level. The PF and cEnKF show the visible intervals for all scenarios, and uEnKF illustrates strong uncertainty in regional scenarios.

Table 2. Tukey test of 0.99 confidence interval for measuring the mean of the RMSE of all approaches on the national influenza level amongst 2011-15 and the regional influenza level in 2014-15. Only the comparisons involving uBass and cBass are presented.

	2011-12		2012-13		2013-14		2014-15			
	◇	P-value	◇	P-value	◇	P-value	◇	P-value		
cEAKF-cBass	4.483	0.000	0.132	0.999	1.462	0.000	-0.739	0.098		
cEnKF-cBass	1.891	0.000	8.768	0.000	6.768	0.000	7.782	0.000		
PF-cBass	2.472	0.000	1.926	0.000	1.932	0.000	0.837	0.037		
uBass-cBass	-0.394	0.000	-0.933	0.006	-1.064	0.000	-1.666	0.000		
uEAKF-cBass	3.597	0.000	5.081	0.000	4.717	0.000	3.309	0.000		
uEnKF-cBass	-0.566	0.000	-0.796	0.033	-0.805	0.000	-1.476	0.000		
uBass-cEAKF	-4.878	0.000	-1.065	0.001	-2.526	0.000	-0.927	0.013		
uBass-cEnKF	-2.286	0.000	-9.701	0.000	-7.832	0.000	-9.449	0.000		
uBass-PF	-2.866	0.000	-2.859	0.000	-2.996	0.000	-2.503	0.000		
uEAKF-uBass	3.992	0.000	6.014	0.000	5.781	0.000	4.975	0.000		
uEnKF-uBass	-0.172	0.265	0.136	0.998	0.259	0.599	0.190	0.993		
	2014-15 Region 1		2014-15 Region 2		2014-15 Region 3		2014-15 Region 4		2014-15 Region 5	
	◇	P-value	◇	P-value	◇	P-value	◇	P-value	◇	P-value
cEAKF-cBass	1.804	0.000	2.705	0.000	-0.422	0.463	-1.273	0.277	0.851	0.685
cEnKF-cBass	6.046	0.000	14.182	0.000	9.997	0.000	8.346	0.000	15.770	0.000
PF-cBass	5.041	0.000	2.681	0.000	4.275	0.000	1.848	0.021	9.586	0.000
uBass-cBass	-0.749	0.357	-0.721	0.000	-0.890	0.001	-1.427	0.158	-1.632	0.038
uEAKF-cBass	4.945	0.000	4.574	0.000	3.203	0.000	3.579	0.000	4.855	0.000
uEnKF-cBass	0.939	0.121	6.466	0.000	8.424	0.000	7.065	0.000	2.436	0.000
uBass-cEAKF	-2.554	0.000	-3.427	0.000	-0.468	0.334	-0.154	1.000	-2.484	0.000
uBass-cEnKF	-6.795	0.000	-14.904	0.000	-10.887	0.000	-9.773	0.000	-17.402	0.000
uBass-PF	-5.790	0.000	-3.402	0.000	-5.165	0.000	-3.275	0.000	-11.218	0.000
uEAKF-uBass	5.694	0.000	5.296	0.000	4.093	0.000	5.006	0.000	6.488	0.000
uEnKF-uBass	1.688	0.000	7.188	0.000	9.314	0.000	8.492	0.000	4.068	0.000
	2014-15 Region 6		2014-15 Region 7		2014-15 Region 8		2014-15 Region 9		2014-15 Region 10	
	◇	P-value	◇	P-value	◇	P-value	◇	P-value	◇	P-value
cEAKF-cBass	-1.325	0.992	-0.063	1.000	0.148	1.000	2.229	0.000	-0.341	0.989
cEnKF-cBass	20.548	0.000	10.117	0.000	6.336	0.000	7.629	0.000	6.499	0.000
PF-cBass	2.949	0.702	15.509	0.000	3.553	0.000	2.863	0.000	7.018	0.000
uBass-cBass	-0.372	1.000	0.105	1.000	-1.314	0.020	-0.540	0.001	-0.624	0.809
uEAKF-cBass	4.612	0.180	6.448	0.000	5.231	0.000	4.672	0.000	3.698	0.000
uEnKF-cBass	24.718	0.000	5.922	0.000	2.799	0.000	7.608	0.000	3.875	0.000
uBass-cEAKF	0.952	0.999	0.168	1.000	-1.462	0.006	-2.769	0.000	-0.283	0.996
uBass-cEnKF	-20.920	0.000	-10.012	0.000	-7.649	0.000	-8.169	0.000	-7.123	0.000
uBass-PF	-3.322	0.571	-15.405	0.000	-4.867	0.000	-3.403	0.000	-7.642	0.000
uEAKF-uBass	4.984	0.114	6.343	0.000	6.544	0.000	5.211	0.000	4.322	0.000
uEnKF-uBass	25.090	0.000	5.817	0.000	4.113	0.000	8.148	0.000	4.500	0.000

◇ the difference ($\times 10^3$)

5.2.2 Curve Fitting

We describe the **calibration power/capacity** and discuss the curve fitting (in Fig. 2) for each filter in the following paragraphs.

Fig. 2 demonstrates the curve fitting plots for the chosen nationwide and regional cases, including the 99% confidence interval. A Kalman-type filter is thought to have strong calibration power if it raises a big numerical change when correcting the prior states to the posterior states (Eq. (9)). The calibration power is decided by the Kalman Gain and thus decided by the covariance matrix. A calibration that is too strong introduces oscillations in the predicting curve, while a calibration capacity that is too weak fails to make the predictions close to the observed data. Hence a suitable calibration helps the filter forecasts more accurately.

We find that uEnKF works well in the nationwide cases, but achieves relatively large RMSE in 9/10 regional cases. The method

cEnKF perceptibly fails to predict the epidemic. In comparison, the capacity of calibration in uEnKF is stronger than that in cEnKF. The predicting curves of uEnKF are oscillating, whereas cEnKF fails to approach the observations, particularly in the regional cases. According to Eq. (13), CCM of the non-negative state ensemble is numerically less than or equal to its UCM. We also find that a numerically small matrix will be influenced by noise easily. However, an excessively strong calibration makes the methods hit the boundaries of the states (e.g. $0 \leq s, i \leq 1$). For instance, in regions 6 and 8, it shows that, the states hitting the bounds will be corrected by the hard limits (Fig. 2), rather than by the filter, distorting the nature of the filtering methods. This problem falls into the category of constrained Kalman Filter [17, 18, 38, 37].

It shows that cEAKF have higher RMSE than uEAKF in all situations. In contrast to EnKF, uEAKF holds a weaker calibration ca-

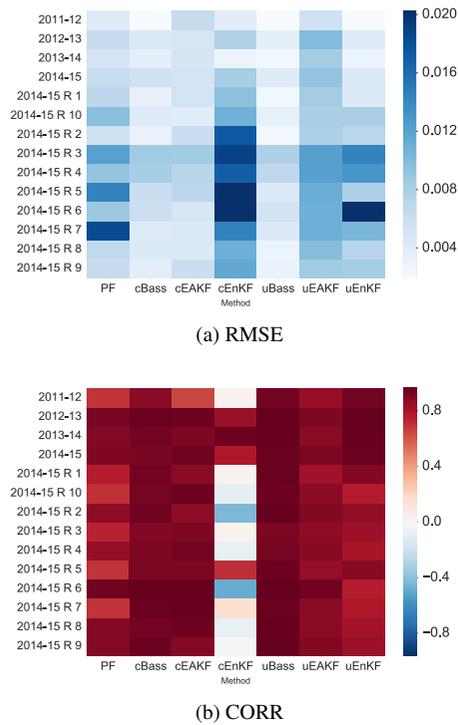


Figure 1. Heat map of the performance metrics, with the mean estimates of the 50 repeated forecasting simulations. R is short for Region.

capacity. For all scenarios after the epidemic peak, the predictions of uEAKF never manage to approach the observations until the very end of the flu seasons. Additionally, cEAKF converges to the observations slower than uEAKF before the epidemic peak, however outperforms uEAKF after the peak. It implies that the calibration capacity of uEAKF gradually declines over time.

The plots show that the performance of the PF strongly relies on the underlying models. The curve fitting for region 3 and 5 illustrates that the PF is not able to accommodate the fluctuations in the observation curves. [34, 44] availed of the SIRS model with the humidity data as extra features, which is superior to the SIR model. Their model addresses the epidemic peak, or multiple peaks, through the filter or humidity component. For a standalone PF, it is not capable of handling multiple peaks, since the SIR model foremost decides the shape of the prediction curve.

BASS with UCM consistently outperforms that with CCM (in 13 scenarios). The comparison between uBass and cBass follows the similar pattern of that in the EnKF. Both methods fit the observation curves well, but uBass has a more suitable calibration capacity and achieves better inferential accuracy.

5.2.3 Resampling Size Analysis

Table 3 exhibits the expected value of the average resampling size over time for the two Bass candidates of the 50 simulations. The two algorithms both resample only a small portion of their particles on average, even when providing accurate forecasts. Amongst all cases, the maximum percentage is merely 13.04% generated by cBass for the region 6 in 2014-15. It also shows that uBass consumes significantly smaller average resampling size per round, compared with cBass (P-values smaller than $10e-11$ for 10 cases). Only in region 6 and 10, does cBass resample less (with P-values 1 and 0.75). In our 500-particle simulations, the mean resampling size interval is

[34.67, 67.06] for cBass, and [19.64, 63.54] for uBass.

Table 3. Average resampling size in the BASS algorithm of 500 particles, with the error threshold $\epsilon = 10^{-5}$. The 0.99 confidence intervals are shown in the parentheses. R is short for region in the first column, for 2014-15. The rightmost column shows the t-test on the null hypothesis $\mu_1 \leq \mu_2$, where μ_1 and μ_2 are the expectation of average resampling quantities of cBass and uBass respectively.

	average resampling size		$\mu_1 \leq \mu_2$
	cBass	uBass	
2011-12	34.67 (34.34, 35.00)	19.64 (18.44, 20.84)	$< 2e-16$
2012-13	52.77 (51.20, 54.22)	35.19 (32.27, 38.11)	$< 2e-16$
2013-14	47.98 (46.14, 49.81)	25.60 (23.96, 27.24)	$< 2e-16$
2014-15	55.02 (53.15, 56.90)	32.94 (31.07, 34.81)	$< 2e-16$
R 1	42.81 (41.75, 43.88)	34.55 (33.23, 35.87)	$< 2e-16$
R 2	43.54 (41.62, 45.46)	35.54 (34.06, 37.01)	$2.75e-14$
R 3	67.06 (64.08, 70.05)	53.62 (51.16, 56.08)	$2.4e-15$
R 4	60.57 (58.27, 62.87)	51.54 (49.61, 53.45)	$1.13e-12$
R 5	53.85 (51.33, 56.36)	35.07 (33.13, 37.01)	$< 2e-16$
R 6	65.21 (62.17, 68.26)	63.54 (60.37, 66.72)	0.16
R 7	48.69 (46.85, 50.53)	62.43 (60.38, 64.48)	1
R 8	49.96 (48.28, 51.63)	34.04 (32.55, 35.54)	$< 2e-16$
R 9	48.73 (47.23, 50.24)	40.01 (38.54, 41.48)	$< 2e-16$
R 10	49.55 (48.45, 50.65)	50.02 (48.56, 51.47)	0.75

5.2.4 Drawback of the Filtering Approach

The shortcoming of the SIR-filter approaches is a one-week lag for predicting the epidemic peak, although the Kalman filters address the micro details to a certain extent. Shaman et al. [34] and Yang et al. [44] employed the SIRS assimilated with the humidity component. More orthogonal features may add value to the standalone SIR predictions. In future work, we plan to investigate how social content (web searches, Tweets, etc.) could help improve the model by allowing proactive predictions to address the peak timing.

6 Conclusion

This paper proposed an improved SIR-based filter algorithm, BASS, for predicting the seasonal influenza level. It empirically achieves the optimal RMSE and CORR in 11 out of 14 major real-world cases. We also examined UCM and CCM in the BASS, EnKF and EAKF. The experimental results indicate that, in our formulation, the BASS and EnKF perform better with UCM, and it is ideal for the EAKF to utilize CCM. Our future work includes combining social data to further enhance the approach of model and filters. We are also interested in combining the SIRS model with the filters to predict seasonal flu continuously. In addition, we would also like to assess whether our filtering algorithm is applicable to other general problems.

ACKNOWLEDGEMENTS

This research was supported by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289. We thank the reviewers for their insightful comments that helped us improve the paper considerably.

REFERENCES

- [1] Saadetin Aksoy, Aydin Muhurcu, and Hakan Kizmaz, 'State and parameter estimation in induction motor using the Extended Kalman Filtering algorithm', *2010 Modern Electric Power Systems*, (3), 1-5, (2010).

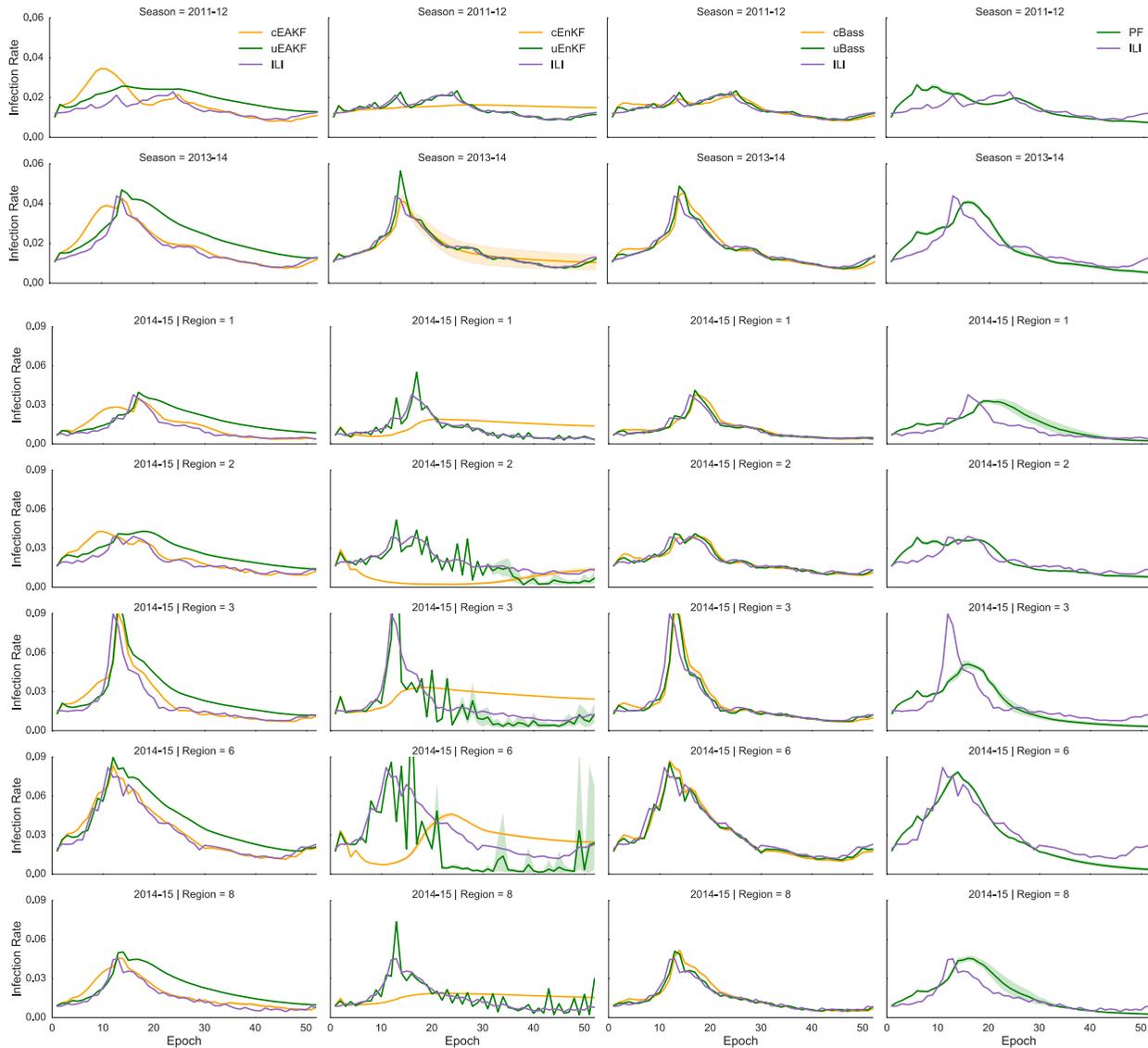


Figure 2. The curve fitting plots for selected national and regional cases. All predictions are with 0.99 confidence interval.

[2] Jeffrey L. Anderson, 'An Ensemble Adjustment Kalman Filter for Data Assimilation', *Monthly Weather Review*, **129**(12), 2884–2903, (2001).

[3] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, 'A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking', *IEEE Transactions on Signal Processing*, **50**(2), 174–188, (2002).

[4] Norman T.J. Bailey, *The Mathematical Theory of Infectious Diseases and Its Applications*, Books on cognate subjects, Griffin, 1975.

[5] Jorge Cadima and Ian Jolliffe, 'On relationships between uncentred and column-centred principal component analysis', *Pak J Statist*, **25**(4), 473–503, (2009).

[6] Carlos M. Carvalho, Michael S. Johannes, Hedibert F. Lopes, and Nicholas G. Polson, 'Particle Learning and Smoothing', *Statistical Science*, **25**(1), 88–106, (2010).

[7] Centers for Disease Control and Prevention. United States Influenza-like Illnesses Data. <http://gis.cdc.gov/grasp/fluview/fluportaldashboard.html>.

[8] Zhe Chen, 'Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond', *Statistics*, **182**(1), 1–69, (2003).

[9] Nawinda Chustagulprom, Sebastian Reich, and Maria Reinhardt, 'A hybrid ensemble transform filter for nonlinear and spatially extended dynamical systems', *ArXiv e-prints*, (sep 2015).

[10] Arnaud Doucet and Am Johansen, 'A tutorial on particle filtering and smoothing: fifteen years later', *Handbook of Nonlinear Filtering*, (December), 656–704, (2011).

[11] Geir Evensen, 'The Ensemble Kalman Filter: Theoretical formulation and practical implementation', *Ocean Dynamics*, **53**(4), 343–367, (2003).

[12] Geir Evensen, *Data assimilation: the ensemble Kalman filter*, Springer Science & Business Media, 2009.

[13] Geir Evensen, 'The ensemble Kalman filter for combined state and parameter estimation', *IEEE Control Systems*, **29**(3), 83–104, (2009).

[14] Marco Frei and Hans R Künsch, 'Bridging the ensemble kalman and particle filters', *Biometrika*, **100**(4), 781–800, (2013).

[15] Jeremy Ginsberg, Matthew H. Mohebbi, Rajan S. Patel, Lynnette Brammer, Mark S. Smolinski, and Larry Brilliant, 'Detecting influenza epidemics using search engine query data.', *Nature*, **457**(7232), 1012–1014, (2009).

[16] Google Inc. Google Flu Trends. <https://www.google.org/flutrends>.

[17] Nachi Gupta, 'Kalman Filtering in the Presence of State Space Equality', in *Control Conference, 2007. CCC 2007. Chinese*, number 07, pp. 107 – 113. IEEE, (2007).

[18] Nachi Gupta and Raphael Hauser, 'Kalman Filtering with Equality and Inequality State Constraints', *arXiv preprint arXiv:0709.2791*, (07), 26, (2007).

- [19] Herbert W. Hethcote, 'The Mathematics of Infectious Diseases', *SIAM Review*, **42**(4), 599–653, (2000).
- [20] Paul Honeine, 'An eigenanalysis of data centering in machine learning', *arXiv preprint arXiv:1407.2904*, (2014).
- [21] Ibrahim Hoteit, Xiaodong Luo, and Dinh-Tuan Pham, 'Particle Kalman Filtering: A Nonlinear Bayesian Framework for Ensemble Kalman Filters', *Monthly weather review*, **140**(2), 528–542, (2012).
- [22] Ibrahim Hoteit, Dinh-Tuan Pham, George Triantafyllou, and Gerasimos Korres, 'A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography', *Monthly Weather Review*, **136**(1), 317–334, (2008).
- [23] John P. Jensen, *Ensemble Kalman filtering for state and parameter estimation on a reservoir model*, Master's thesis, Norges teknisk-naturvitenskapelige universitet, 2007.
- [24] Michael Johannes and Nicholas Polson, 'Particle Filtering and Parameter Learning', *Available at SSRN 983646*, (April 2006), 1–42, (2008).
- [25] Craig J. Johns and Jan Mandel, 'A two-stage ensemble kalman filter for smooth data assimilation', *Environmental and Ecological Statistics*, **15**(1), 101–110, (2008).
- [26] Alicia R. Karspeck and Jeffrey L. Anderson, 'Experimental implementation of an ensemble adjustment filter for an intermediate ENSO model', *Journal of Climate*, **20**(18), 4638–4658, (2007).
- [27] William O. Kermack and Anderson G. McKendrick, 'A contribution to the mathematical theory of epidemics', in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 115, pp. 700–721. The Royal Society, JSTOR, (1927).
- [28] Madhav Marathe and Anil Kumar S. Vullikanti, 'Computational epidemiology', *Communications of the ACM*, **56**(7), 88–96, (July 2013).
- [29] Hamid Moradkhani, Soroosh Sorooshian, Hoshin V Gupta, and Paul R. Houser, 'Dual state-parameter estimation of hydrological models using ensemble Kalman filter', *Advances in Water Resources*, **28**(2), 135–147, (2005).
- [30] Kevin P. Murphy, *Machine learning: a probabilistic perspective*, MIT press, 2012.
- [31] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.
- [32] Mauricio Santillana, André T. Nguyen, Mark Dredze, Michael J. Paul, Elaine O. Nsoesie, and John S. Brownstein, 'Combining Search, Social Media, and Traditional Data Sources to Improve Influenza Surveillance', *PLoS Computational Biology*, **11**(10), e1004513, (2015).
- [33] Mauricio Santillana, Elaine O. Nsoesie, Sumiko R. Mekaru, David Scales, and John S. Brownstein, 'Using Clinicians' Search Query Data to Monitor Influenza Epidemics', *Clinical Infectious Diseases*, **59**(10), 1446–1450, (2014).
- [34] Jeffrey Shaman and Alicia Karspeck, 'Forecasting seasonal outbreaks of influenza', *Proceedings of the National Academy of Sciences of the United States of America*, **109**(50), 20425–20430, (2012).
- [35] Constantinos I. Siettos and Lucia Russo, 'Mathematical modeling of infectious disease dynamics.', *Virulence*, **4**(4), 295–306, (2013).
- [36] Dan Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*, John Wiley & Sons, 2006.
- [37] Dan Simon, 'Kalman filtering with state constraints: a survey of linear and nonlinear algorithms', *IET Control Theory & Applications*, **4**(8), 1303, (2010).
- [38] Vincent Sircoulomb, Ghaleb Hoblos, Houcine Chafouk, and José Ragot, 'State estimation under nonlinear state inequality constraints. A tracking application', *2008 16th Mediterranean Conference on Control and Automation*, 1669–1674, (2008).
- [39] Laura Slivinski, Elaine Spiller, and Amit Apte, 'A hybrid particle-ensemble kalman filter for high dimensional lagrangian data assimilation', in *Dynamic Data-Driven Environmental Systems Science*, 263–273, Springer, (2015).
- [40] Laura Slivinski, Elaine Spiller, Amit Apte, and Björn Sandstede, 'A hybrid particle-ensemble kalman filter for lagrangian data assimilation', *Monthly Weather Review*, **143**(1), 195–211, (2015).
- [41] Andreas S. Stordal, Hans A. Karlsen, Geir Nævdal, Hans J. Skaug, and Brice Vallès, 'Bridging the ensemble Kalman filter and particle filters: The adaptive Gaussian mixture filter', *Computational Geosciences*, **15**(2), 293–305, (2011).
- [42] Geir Storvik, 'Particle filters for state-space models with the presence of unknown static parameters', *IEEE Transactions on Signal Processing*, **50**(2), 281–289, (2002).
- [43] Jeffrey S. Whitaker and Thomas M. Hamill, 'Ensemble data assimilation without perturbed observations', *Monthly Weather Review*, **130**(7), 1913–1924, (2002).
- [44] Wan Yang, Alicia Karspeck, and Jeffrey Shaman, 'Comparison of Filtering Methods for the Modeling and Retrospective Forecasting of Influenza Epidemics', *PLoS Computational Biology*, **10**(4), (2014).
- [45] Wan Yang, Marc Lipsitch, and Jeffrey Shaman, 'Inference of seasonal and pandemic influenza transmission dynamics', *Proceedings of the National Academy of Sciences*, **112**(9), 201415012, (2015).

Socially-Aware Multiagent Learning: Towards Socially Optimal Outcomes

Xiaohong Li^{1,2} and Chengwei Zhang² and Jianye Hao³ and Karl Tuyls⁴
and Siqi Chen⁵ and Zhiyong Feng⁶

Abstract. In multiagent systems the capability of learning is important for an agent to behave appropriately in face of unknown opponents and a dynamic environment. From the system designer's perspective, it is desirable if the agents can learn to coordinate towards socially optimal outcomes, while also avoiding being exploited by selfish opponents. To this end, we propose a novel gradient ascent based algorithm (SA-IGA) which augments the basic gradient-ascent algorithm by incorporating social awareness into the policy update process. We theoretically analyze the learning dynamics of SA-IGA using dynamical system theory, and SA-IGA is shown to have linear dynamics for a wide range of games including symmetric games. The learning dynamics of two representative games (the prisoner's dilemma game and coordination game) are analyzed in detail. Based on the idea of SA-IGA, we further propose a practical multiagent learning algorithm, called SA-PGA, based on the Q-learning update rule. Simulation results show that an SA-PGA agent can achieve higher social welfare than previous social-optimality oriented Conditional Joint Action Learner (CJAL) and also is robust against individually rational opponents by reaching Nash equilibrium solutions.

1 Introduction

In multiagent systems the ability of learning is important for an agent to adaptively adjust its behaviours in response to coexisting agents and unknown environments in order to optimize its performance. Multiagent learning algorithms have received extensive attention in the literature, and many learning strategies [6, 15, 4, 14, 17] have been proposed to facilitate coordination among agents.

The multi-agent learning criteria proposed in [5] require that an agent should be able to converge to a stationary policy against some class of opponents (*convergence*) and the best-response policy against any stationary opponent (*rationality*). If both agents adopt a rational learning strategy in the context of repeated games and also their strategies converge, then they will converge to a Nash equilibrium of the stage game. Indeed, convergence to Nash equilibrium has been the most commonly accepted goal to pursue in multiagent learning literature. Until now, a number of gradient-ascent based multiagent learning algorithms [20, 5, 1, 24] have been subsequently pro-

posed in order to converge to Nash equilibriums with improved convergence performance and more relaxed assumptions (less information is required). Another well-studied family of multiagent learning strategies is based on reinforcement learning (e.g., Q-learning [23]). Representative examples include distributed Q-learning in cooperative games [11], minimax Q-learning in zero-sum games [12], Nash Q-learning in general-sum games [10], and other extensions [13, 6], to name just a few.

1's payoff 2's payoff	Agent 2's actions		
	C	D	
Agent 1's actions	C	3/3	0/5
	D	5/0	1/1

Table 1: The Prisoner's Dilemma Game

All the aforementioned learning strategies pursue converging to Nash equilibriums under self-play, however, Nash equilibrium solutions may be undesirable in many scenarios. One well-known example is the prisoner's dilemma (PD) game shown in Table 1. By converging to the Nash equilibrium (D, D), both agents obtain the payoff of 1, while they could have obtained a much higher payoff of 3 by coordinating on the non-equilibrium outcome (C, C). In situations like the PD game, converging to the socially optimal outcome under self-play would be more preferred. To address this issue, one natural modification for a gradient-ascent learner is to update its policy along the direction of maximizing the sum of all agents' expected payoff instead of its own. However, in an open environment, the agents are usually designed by different parties and may have not the incentive to follow the strategy we design. The above way of updating strategies would be easily exploited and taken advantage by (equilibrium-driven) self-interested agents. Thus it would be highly desirable if an agent can converge to socially optimal outcomes under self-play and Nash equilibrium against self-interested agents to avoid being exploited.

In this paper, we first propose a new gradient-ascent based algorithm (SA-IGA) which augments the basic gradient ascent algorithm by incorporating social awareness into the policy update process. A SA-IGA agent holds a social attitude to reflect its socially-aware degree, which can be adjusted adaptively based on its relative performance compared to its opponent. An SA-IGA agent seeks to update its policy in the direction of increasing its overall payoff, which is defined as the average of its individual and the social payoff, weighted by its socially-aware degree. We theoretically show that for a wide

¹ Tianjin Key Laboratory of Advanced Networking, China.

² School of Computer Science and Technology, Tianjin University, China., Email: {xiaohongli, chenzy}@tju.edu.cn

³ School of Computer Software, Tianjin University, China. Email: jianye.hao@tju.edu.cn; Corresponding author

⁴ University of Liverpool, UK. Email: k.tuyls@liverpool.ac.uk

⁵ Southwest University, China. Email: siqichen@swu.edu.cn

⁶ School of Computer Software, Tianjin University, China. Email: zhiyongfeng@tju.edu.cn

range of games (e.g., symmetric games), the dynamics of SA-IGAs under self-play exhibit linear characteristics. For general-sum games, it may exhibit non-linear dynamics which can still be analyzed numerically. The learning dynamics of two representative games (PD game and coordination game) are analyzed in details. Like previous theoretical multiagent learning algorithms, SA-IGA also requires to know the opponent's policy and the game structure.

To relax the above assumption, we then propose a practical gradient ascent based multiagent learning strategy, called Socially-aware Policy Gradient Ascent (SA-PGA). SA-PGA relaxes the above assumption by estimating the performance of itself and the opponent using Q-learning techniques. We empirically evaluate its performance in different types of benchmark games and simulation results show that SA-PGA agent outperforms previous learning strategies in terms of maximizing the social welfare and Nash product of the agents. Besides, SA-PGA is also shown to be robust against individually rational opponents and converges to Nash equilibrium solutions.

The remainder of the paper is organized as follows. Section 2 reviews normal-form games and the basic gradient ascent approach. Section 3 introduces the SA-IGA algorithm and analyzes its learning dynamics theoretically. Section 4 presents the practical multiagent learning algorithm SA-PGA in detail. In Section 5, we extensively evaluate the performance of SA-PGA under various benchmark games. Lastly we conclude the paper and point out future directions in Section 6.

2 Background

2.1 Normal-form games

In a two-player, two-action, general-sum normal-form game, the payoffs for each player $i \in \{\mathbf{r}, \mathbf{c}\}$ can be specified by a matrix as follows,

$$R_i = \begin{bmatrix} r_{11}^i & r_{12}^i \\ r_{21}^i & r_{22}^i \end{bmatrix}$$

Each player i simultaneously selects an action from its action set $A_i = \{1, 2\}$, and the payoff of each player is determined by their joint actions. For example, if player \mathbf{r} selects the pure strategy action 1 while player \mathbf{c} selects the pure strategy action 2, then player \mathbf{r} receives a payoff of $r_{12}^{\mathbf{r}}$ and player \mathbf{c} receives the payoff of $r_{12}^{\mathbf{c}}$.

Apart from pure strategies, each player can also employ a mixed strategy to make decisions. A mixed strategy can be represented as a probability distribution over the action set and a pure strategy is a special case of mixed strategies. Let $p_{\mathbf{r}} \in [0, 1]$ and $p_{\mathbf{c}} \in [0, 1]$ denote the probability of choosing action 1 by player \mathbf{r} and player \mathbf{c} respectively. Given a joint mixed strategy $(p_{\mathbf{r}}, p_{\mathbf{c}})$, the expected payoffs of player \mathbf{r} and player \mathbf{c} can be specified as follows,

$$\begin{aligned} V_{\mathbf{r}}(p_{\mathbf{r}}, p_{\mathbf{c}}) &= r_{11}^{\mathbf{r}} p_{\mathbf{r}} p_{\mathbf{c}} + r_{12}^{\mathbf{r}} p_{\mathbf{r}} (1 - p_{\mathbf{c}}) + r_{21}^{\mathbf{r}} (1 - p_{\mathbf{r}}) p_{\mathbf{c}} \\ &\quad + r_{22}^{\mathbf{r}} (1 - p_{\mathbf{r}}) (1 - p_{\mathbf{c}}) \\ V_{\mathbf{c}}(p_{\mathbf{r}}, p_{\mathbf{c}}) &= r_{11}^{\mathbf{c}} p_{\mathbf{r}} p_{\mathbf{c}} + r_{12}^{\mathbf{c}} p_{\mathbf{r}} (1 - p_{\mathbf{c}}) + r_{21}^{\mathbf{c}} (1 - p_{\mathbf{r}}) p_{\mathbf{c}} \\ &\quad + r_{22}^{\mathbf{c}} (1 - p_{\mathbf{r}}) (1 - p_{\mathbf{c}}) \end{aligned} \quad (1)$$

respectively.

A joint strategy is called a Nash Equilibrium (NE), if no player can get a better expected payoff by changing its current strategy unilaterally. Formally, $(p_{\mathbf{r}}^*, p_{\mathbf{c}}^*) \in [0, 1]^2$ is a NE, iff $V_{\mathbf{r}}(p_{\mathbf{r}}^*, p_{\mathbf{c}}^*) \geq V_{\mathbf{r}}(p_{\mathbf{r}}, p_{\mathbf{c}}^*)$ and $V_{\mathbf{c}}(p_{\mathbf{r}}^*, p_{\mathbf{c}}^*) \geq V_{\mathbf{c}}(p_{\mathbf{r}}^*, p_{\mathbf{c}})$ for any $(p_{\mathbf{r}}, p_{\mathbf{c}}) \in [0, 1]^2$.

2.2 Gradient Ascent (GA)

When a game is played repeatedly, an individually rational player updates its strategy in order to maximize its expected payoffs. A player i employing GA-based algorithms updates its policy towards the direction of its expected reward gradient, which can be shown in the following equations.

$$\Delta p_i^{(t+1)} \leftarrow \eta \frac{\partial V_i(p^{(t)})}{\partial p_i} \quad (2)$$

$$p_i^{(t+1)} \leftarrow \Pi_{[0,1]}(p_i^{(t)} + \Delta p_i^{(t+1)}) \quad (3)$$

where parameter η is the gradient step size, and $\Pi_{[0,1]}$ is the projection function mapping the input value to the valid probability range of $[0, 1]$, used to prevent the gradient moving the strategy out of the valid probability space. Formally, we have,

$$\Pi_{[0,1]}(x) = \operatorname{argmin}_{z \in [0,1]} |x - z| \quad (4)$$

To simplify the notations, let us denote $u_i = r_{11}^i + r_{22}^i - r_{12}^i - r_{21}^i$, $c_i = r_{12}^i - r_{22}^i$ and $d_i = r_{21}^i - r_{12}^i$. For the two-player case, the above way of GA-based updating in Equation 2 and 3 can be represented as follows,

$$p_{\mathbf{r}}^{(t+1)} \leftarrow \Pi_{[0,1]}(p_{\mathbf{r}}^{(t)} + \eta(u_{\mathbf{r}} p_{\mathbf{c}}^{(t)} + c_{\mathbf{r}})) \quad (5)$$

$$p_{\mathbf{c}}^{(t+1)} \leftarrow \Pi_{[0,1]}(p_{\mathbf{c}}^{(t)} + \eta(u_{\mathbf{c}} p_{\mathbf{r}}^{(t)} + d_{\mathbf{c}})) \quad (6)$$

In the case of infinitesimal gradient step size ($\eta \rightarrow 0$), the learning dynamics of the players can be modeled as a system of differential equations and analyzed using dynamic system theory [20]. It is proved that the agents will converge to a Nash equilibrium, or if the strategies themselves do not converge, then their average payoffs will nevertheless converge to the average payoffs of a Nash equilibrium.

Following [20], various GA-based algorithms have been proposed to improve the convergence performance towards Nash equilibria and representative examples include IGA-WoLF (Win or Learn Fast) [5], Weighted Policy Learner (PWL) [1] and Gradient Ascent With Policy Prediction (IGA-PP) [24]. In contrast, in this work, we seek to incorporate the social awareness into GA-based strategy update and aim to improve social welfare of the players under self-play rather than pursuing Nash equilibrium solutions. Meanwhile, individually rational behaviour is employed when playing against a selfish agent. Similar idea of adaptively behaving differently against different opponents was also employed in previous algorithms [13, 9, 16, 7]. However, all the existing works focus on maximizing an agent's individual payoff against different opponents in different types of games, but do not directly take into consideration the goal of maximizing social welfare (e.g., cooperate in the prisoner's dilemma game).

3 Socially-Aware Infinitesimal Gradient Ascent (SA-IGA)

In our daily life, people usually do not behave as a purely rational entity that seeks to achieve Nash equilibrium solutions. For example, when two persons play a PD game, reaching mutual cooperation may be observed frequently. Similar phenomena have also been observed in extensive human-based experiments in games such as the Public Good game and Ultimatum game, in which human subjects are usually found to obtain much higher payoffs by mutual cooperation rather than pursuing Nash equilibrium solutions. If the above

phenomenon is transformed into computational models, it indicates that an agent may not only update its policy in the direction of maximizing its own payoff, but also take into consideration the payoff of others. We call this type of agents *socially-aware agents*.

In this paper, we incorporate the social awareness into the gradient-ascent based learning algorithm. As such, apart from learning to maximize its individual payoff, an agent is also equipped with the social awareness such that it can (1) reach mutually cooperative solutions faced with another socially-aware opponent (self-play); (2) behave in a purely individually rational manner against a purely rational opponent.

Specifically, for each agent $i \in \{\mathbf{r}, \mathbf{c}\}$, we distinguish two types of expected payoffs, namely V_i^{idv} and V_i^{soc} . The payoff $V_i^{idv}(p_r, p_c)$ and $V_i^{soc}(p_r, p_c)$ represent the individual and social payoff (the average payoff of both players) that agent i perceives under the joint strategy (p_r, p_c) respectively. The payoff $V_i^{idv}(p_r, p_c)$ follows the same definition as Equation (1) and the payoff $V_i^{soc}(p_r, p_c)$ can be defined as follows,

$$V_i^{soc}(p_r, p_c) = \frac{1}{2}[V_r^{idv}(p_r, p_c) + V_c^{idv}(p_r, p_c)], \forall i \in \{\mathbf{r}, \mathbf{c}\} \quad (7)$$

Each agent i adopts a social attitude w_i to reflect its socially-aware degree. The social attitude intuitively models an agent's socially friendly degree towards its partner. Specifically, it is used as the weighting factor to adjust the relative importance between V_i^{idv} and V_i^{soc} , and agent i 's overall expected payoff is defined as follows,

$$V_i(p_r, p_c) = (1 - w_i)V_i^{idv}(p_r, p_c) + w_iV_i^{soc}(p_r, p_c) \quad (8)$$

where $i \in \{\mathbf{r}, \mathbf{c}\}$. Each agent i updates its strategy in the direction of maximizing the value of V_i . Formally we have,

$$\begin{aligned} \Delta p_i &\leftarrow \eta_p \frac{\partial V_i(p_r, p_c)}{\partial p_i}, \\ p_i &\leftarrow \Pi_{[0,1]}(p_i + \Delta p_i) \end{aligned} \quad (9)$$

where parameter η_p is the gradient step size of p_i . If $w_i = 0$, it means that the agent seeks to maximize its individual payoff only, which is reduced to the case of traditional gradient-ascent updating; if $w = 1$, it means that the agent seeks to maximize the sum of the payoffs of both players.

Finally, each agent i 's socially-aware degree is adaptively adjusted in response to the relative value of V_i^{idv} and V_i^{soc} as follows. During each round, if player i 's own expected payoff V_i^{idv} exceeds the value of V_i^{soc} , then player i increases its social attitude w_i , (i.e., it becomes more social-friendly because it perceives itself to be earning more than the average). Conversely, if V_i^{idv} is less than V_i^{soc} , then the agent tends to care more about its own interest by decreasing the value of w_i . Formally we have,

$$w_i = \begin{cases} \Pi_{[0,1]}(w_i + \Delta w_i) & \text{if } V_i^{idv} > V_i^{soc} \\ \Pi_{[0,1]}(w_i - \Delta w_i) & \text{if } V_i^{idv} < V_i^{soc} \end{cases} \quad (10)$$

where Δw_i is the adjustment step size of w_i .

3.1 Theoretical Modeling and Analysis of SA-IGA

An important aspect of understanding the behaviour of a multiagent learning algorithm is theoretically modelling and analyzing its underlying dynamics [22, 18, 4, 2]. In this section, we first show that the learning dynamics of SA-IGA under self-play can be modeled as a system of differential equations.

Based on the adjustment rules in Eq (9) and (10), the learning dynamics of a SA-IGA agent can be modeled as a set of equations in (11). For ease of exposition, we concentrate on unconstrained update equations by removing the policy projection function which does not affect our qualitative analytical results. Any trajectory with linear (non-linear) characteristic without constraints is still linear (non-linear) when a boundary is enforced.

$$\begin{aligned} \Delta p_i^{(t+1)} &\leftarrow \eta_p \frac{\partial V_i(p_r^{(t)}, p_c^{(t)})}{\partial p_i} \\ \Delta w_i^{t+1} &\leftarrow \eta_w (V_i^{idv} - V_i^{soc}) \\ p_i^{(t+1)} &\leftarrow p_i^{(t)} + \Delta p_i^{(t+1)} \\ w_i^{(t+1)} &\leftarrow w_i^{(t)} + \Delta w_i^{(t+1)} \end{aligned} \quad (11)$$

As $\eta_p \rightarrow 0$ and $\eta_w \rightarrow 0$, it is straightforward to show that the above equations become differential. Substituting V_i^{idv} and V_i^{soc} by their definitions (Eq. (1) and (7)). Thus the unconstrained dynamics of the strategy pair and social attitudes as a function of time is modelled by the following system of differential equations:

$$\begin{aligned} \dot{p}_r &= \left(u_r + \frac{u_c - u_r}{2} w_r\right) p_c + \frac{c_c - c_r}{2} w_r + c_r \\ \dot{p}_c &= \left(u_c + \frac{u_r - u_c}{2} w_c\right) p_r + \frac{d_r - d_c}{2} w_c + d_c \\ \dot{w}_r &= \varepsilon [(u_r - u_c) p_r p_c + (c_r - c_c) p_r + (d_c - d_r) p_c + e] \\ \dot{w}_c &= -\varepsilon [(u_r - u_c) p_r p_c + (c_r - c_c) p_r + (d_c - d_r) p_c + e] \end{aligned} \quad (12)$$

where $u_i = r_{11}^i + r_{22}^i - r_{12}^i - r_{21}^i$, $c_i = r_{12}^i - r_{22}^i$, $d_i = r_{21}^i - r_{22}^i$, $e = r_{22}^r - r_{22}^c$ with $i \in \{\mathbf{r}, \mathbf{c}\}$ and $\varepsilon = \frac{\eta_w}{\eta_p} > 0$.

Based on the above theoretical modelling, next we analyze the learning dynamics of SA-IGA qualitatively as follows.

Theorem 1 SA-IGA has non-linear dynamics when $u_r \neq u_c$.

Proof 1 From the system of differential equations in (12), it is straightforward to verify that the dynamics of SA-IGA learners are non-linear when $u_r \neq u_c$ due to the existence of $w_r p_c$, $w_c p_r$ or $p_r p_c$ in all equations.

Since SA-IGA's dynamics are non-linear when $u_r \neq u_c$, in general we cannot obtain a closed-form solution, but we can still resort to solve the equations numerically to obtain useful insight in the system's dynamics. Moreover, a wide range of important games fall into the category of $u_r = u_c$, in which the system of equations become linear. Therefore, it allows us to use dynamic system theory to systematically analyze the underlying dynamics of SA-IGA.

r's payoff c's payoff	Agent c's actions	
	action 1	action 2
Agent r's actions	action 1	a/a c/d
	action 2	d/c b/b

Table 2: The General Form of a Symmetric Game

Theorem 2 SA-IGA has linear dynamics when the game itself is symmetric.

Proof 2 A two-player two-action symmetric game can be represented in Table 2 in general. It is obvious to check that it satisfies the constraint of $u_r = u_c$, given that $u_i = r_{11}^i + r_{22}^i - r_{12}^i - r_{21}^i$, $i \in \{\mathbf{r}, \mathbf{c}\}$. Thus the theorem holds.

3.2 Dynamics Analysis of SA-IGA

In the previous section we mainly analyzed the dynamics of SA-IGA in a qualitative manner. In this section, we provide a detailed analysis of SA-IGA's learning dynamics in two representative games: the Prisoner's Dilemma game (Table 3) (as a symmetric game example) and Coordination game (Table 4) (as an asymmetric game example). Specifically we analyze the SA-IGA's learning dynamics by identifying the existing equilibrium points, which provides useful insights into understanding the dynamics of SA-IGA.

Theorem 3 *The dynamics of SA-IGA algorithm under Prisoner's Dilemma (PD) game have three types of equilibrium points:*

1. $(0, 0, w_r^*, w_c^*)$, where $w_r^*, w_c^* < \min \left\{ \frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S} \right\}$;
2. $(1, 1, w_r^*, w_c^*)$, where $w_r^*, w_c^* > \max \left\{ \frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S} \right\}$;
3. (p^*, p^*, w_r^*, w_c^*) , others

The first and second type of equilibrium points are stable, while the last is not. We say that an equilibrium point is stable if once the strategy starts "close enough" to the equilibrium (within a distance δ from it), it will remain "close enough" to the equilibrium point forever.

r's payoff c's payoff	Agent c's actions		
	C	D	
Agent r's actions	C	R/R	S/T
	D	T/S	P/P

Table 3: The Prisoner's Dilemma Game (where $T > R > P > S$)

Proof 3 *Following the system of differential equations in Equations (12), we can express the dynamics of SA-IGA in PD game as follows:*

$$\begin{aligned}
 \dot{p}_r &= (u) p_c + \frac{T-S}{2} w_r + S - P \\
 \dot{p}_c &= (u) p_r + \frac{T-S}{2} w_c + S - P \\
 \dot{w}_r &= \varepsilon (S - T) (p_r - p_c) \\
 \dot{w}_c &= -\varepsilon (S - T) (p_r - p_c)
 \end{aligned} \quad (13)$$

where $\varepsilon = \frac{\eta w}{\eta p} > 0, u = R + P - S - T$. We start by proving the last type of equilibrium points: If there exist an equilibrium eq = $(p_r^*, p_c^*, w_r^*, w_c^*)^T \in (0, 1)^4$, then we have $\dot{p}_i(\text{eq}) = 0$ and $\dot{w}_i(\text{eq}) = 0, i \in \{r, c\}$. By solving the above equations, we have $p_r^* = p_c^* = \frac{S-T}{2u} w_r^* + \frac{P-S}{u}$ and $w_r^* = w_r^* = w_c^*$. Since $p_r^*, p_c^* \in (0, 1)$, then we have,

$$\begin{aligned}
 w_r, w_c &> \min \left\{ \frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S} \right\} \\
 w_r, w_c &< \max \left\{ \frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S} \right\}
 \end{aligned}$$

Then eq = $(p_r^*, p_c^*, w_r^*, w_c^*)^T$ is an equilibrium. The stability of eq can be verified using theories of non-linear dynamics[19]. By expressing the unconstrained update differential equations in the form of $\dot{x} = Ax + B$, we have

$$A = \begin{bmatrix} 0 & u & T-S & 0 \\ u & 0 & 0 & T-S \\ \varepsilon(S-T) & \varepsilon(T-S) & 0 & 0 \\ \varepsilon(T-S) & \varepsilon(S-T) & 0 & 0 \end{bmatrix}$$

After calculating matrix A's eigenvalue, then we have $\lambda_1 = 0, \lambda_2 = u, \lambda_3 = -\frac{u}{2} + k$ and $\lambda_4 = -\frac{u}{2} - k$, where k is a constant. Since there exist an eigenvalue $\lambda > 0$, the equilibrium eq is not stable.

Next we turn to prove the first type of equilibrium. In this case, we need to put the projection function back since we are dealing with boundary cases. If $p_i = 0, i \in \{r, c\}$, according to the known conditions, we have $w_r, w_c < \min \left\{ \frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S} \right\}$. Combined with the unconstrained update differential equations, we have $\lim_{p_i} \dot{p}_i < 0$, then p_i remains unchanged. And because $p_r = p_c = 0$, then for $\forall w_i \in [0, 1], \dot{w}_i((0, 0, w_r^*, w_c^*)) = 0$, then $((0, 0, w_r^*, w_c^*))$ is an equilibrium.

Because $w_r, w_c < \min \left\{ \frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S} \right\}$, there exist a $\delta > 0$, and a set $U(\text{eq}, \delta) = \{x \in [0, 1]^4 \mid |x - \text{eq}| < \delta\}$, that for $\forall x \in U(\text{eq}, \delta), \lim_{p_i} \dot{p}_i < 0$. Thus p will stabilize on the point of 0. Also, because

$$\lim_{t \rightarrow 0} \dot{w}_i = (S - T) \lim_{t \rightarrow 0} (p_r - p_c) = (S - T) \lim_{t \rightarrow 0} (0 - 0) = 0$$

then w is also stable, and thus the equilibrium eq is stable.

The second type of equilibrium can be proved similarly, which is omitted here.

Intuitively, for a PD game, from Theorem 3, we know that if both SA-IGA players are initially sufficiently social-friendly (the value of w is larger than a certain threshold), then they will always converge to mutual cooperation of (C, C) . In other words, given that the value of w exceeds a certain threshold, the strategy point of $(1, 1)$ (or (C, C)) in the strategy space is asymptotically stable. If both players start with a low socially-aware degree (w is smaller than certain threshold), then they will always converge to mutual defection of (D, D) eventually. For the rest of cases, there exist infinite number of equilibrium points in-between the above two extreme cases, all of which are not stable.

Next we turn to analyze the dynamics of SA-IGA in a coordination game by identifying all equilibrium points. The general form of a coordination game is shown in Table 4. Intuitively, both Nash equilibria (C, C) and (D, D) can be part of the equilibrium points depending on the agents' social-aware degrees. Formally we have,

r's payoff c's payoff	Agent c's actions		
	C	D	
Agent r's actions	C	R/r	S/s
	D	T/t	P/p

Table 4: The General Form of a Coordination Game (where $R > T \wedge P > S$ and $r > s \wedge p > t$)

Theorem 4 *The dynamics of SA-IGA algorithm under a coordination game have three types of equilibrium points:*

1. $(0, 0, w_r^*, w_c^*)$, with $w_r^* = 1 \wedge w_c^* = 0$ when $P > p > s; w_r^* = 0 \wedge w_c^* = 1$ when $T < P < p$; and $(\frac{s-S}{2} w_r^* < P - S) \wedge (\frac{T-t}{2} w_c^* < p - t)$ when $P = p$;
2. $(1, 1, w_r^*, w_c^*)$, with $w_r^* = 1 \wedge w_c^* = 0$ when $R > r > t; w_r^* = 0 \wedge w_c^* = 1$ when $T < R < r$; and $(\frac{T-t}{2} w_r^* < R - T) \wedge (\frac{S-s}{2} w_c^* < r - s)$ when $R = r$;
3. others non-boundary equilibrium points $(p_r^*, p_c^*, w_r^*, w_c^*)$

The first and second types of equilibrium points are stable, while the last non-boundary equilibrium points are not. The definition of a stable equilibrium point is the same as in Theorem 3.

Proof 4 Following the system of differential equations in Equations (12), we can express the dynamics of SA-IGA in coordination game as follows:

$$\begin{aligned}\dot{p}_r &= \left(u_r + \frac{u_c - u_r}{2} w_r\right) p_c + \frac{c_c - c_r}{2} w_r + c_r \\ \dot{p}_c &= \left(u_c + \frac{u_r - u_c}{2} w_c\right) p_r + \frac{d_r - d_c}{2} w_c + d_c \\ \dot{w}_r &= \varepsilon [(u_r - u_c) p_r p_c + (c_r - c_c) p_r + (d_c - d_r) p_c + e] \\ \dot{w}_c &= -\dot{w}_r\end{aligned}\quad (14)$$

where $\varepsilon = \frac{\eta w}{\eta p} > 0$, $u_r = R + P - S - T > 0$, $u_c = r + p - s - t > 0$, $c_r = S - P$, $c_c = s - p$, $d_r = T - P$, $d_c = t - p$, and $e = P - p$.

We can see that the dynamic of coordination game is nonlinear when $u_r \neq u_c$. We start with proving the last type of equilibrium points first:

If there exist an equilibrium $eq = (p_r^*, p_c^*, w_r^*, w_c^*)^T \in (0, 1)^4$, then we have $\dot{p}_i(eq) = 0$ and $\dot{w}_i(eq) = 0$, $i \in \{r, c\}$. By linearizing the unconstrained update differential equations into the form of $\dot{x} = Ax + B$ in point $eq = (p_r^*, p_c^*, w_r^*, w_c^*)^T$, we have

$$A = \begin{bmatrix} 0 & u_r^* & a_{13} & 0 \\ u_c^* & 0 & 0 & a_{24} \\ -\varepsilon a_{13} & \varepsilon a_{24} & 0 & 0 \\ \varepsilon a_{13} & -\varepsilon a_{24} & 0 & 0 \end{bmatrix}$$

where $u_r^* = u_r + \frac{u_c - u_r}{2} w_r^*$, $u_c^* = u_c + \frac{u_r - u_c}{2} w_c^*$, $c_r^* = \frac{c_c - c_r}{2} w_r^* + c_r$, and $d_c^* = \frac{d_r - d_c}{2} w_c^* + d_c$. The parameters a_{ij} are represented as functions of p_r^* , p_c^* , w_r^* and w_c^* . Without loss of generality, we set $u_r \geq u_c$. Because of $u_r \geq u_c > 0$, and $w_r^*, w_c^* \in [0, 1]$, we have $u_r^* \in [\frac{u_c + u_r}{2}, u_r]$ and $u_c^* \in [u_c, \frac{u_c + u_r}{2}]$, which means $u_r^* > u_c^* > 0$.

After calculating matrix A 's eigenvalue in Matlab, we have an eigenvalue $\lambda_1 = 0$, an eigenvalue λ_2 with its real part $Re(\lambda_2) > 0$, an eigenvalue λ_3 with $Re(\lambda_3) < 0$ and an eigenvalue λ_4 close to 0. Since there exists an eigenvalue $\lambda > 0$, the equilibrium eq is not stable[19].

Next we turn to prove the first type of equilibrium. In this case, we need to put the projection function back since we are dealing with boundary cases.

For the case $P > p > s$, we have $V_i^{idv}(eq) > V_i^{soc}(eq)$, thus $\dot{w}_r(eq) > 0$ and $\dot{w}_c(eq) < 0$, which means w_r and w_c will keeps $w_r = 1$ and $w_c = 0$. Because $\dot{p}_r(eq) = \frac{s-p+S-P}{2} < 0$ and $\dot{p}_c(eq) = t - p < 0$, then p_r and p_c will keeps $p_r = 0$ and $p_c = 0$. According to the continuity theorem of differential equations [8], $(0, 0, 1, 0)$ is a stable equilibrium. The case $p > P > T$ can be proved similarly, which is omitted here.

For the case $P = p$, we have $V_i^{idv} = V_i^{soc}$, then $\dot{w}_r(eq) = -\dot{w}_c(eq) = \varepsilon (V_r^{idv} - V_r^{soc}) = 0$. Because $(\frac{T-t}{2} w_c^* < p - t)$, we have $\dot{p}_r = \frac{T-t}{2} w_c^* + t - p < 0$. Because $(\frac{s-S}{2} w_r^* < P - S)$, we have $\dot{p}_c = \frac{s-S}{2} w_r^* + S - P < 0$. According to the continuity theorem of differential equations, $(0, 0, w_r^*, w_c^*)$ is a stable equilibrium. The stability of the second type of equilibrium points can be proved similarly, which is omitted here.

4 A Practical Algorithm

In SA-IGA, each agent needs to know the policy of its opponent and the payoff matrix, which are usually not available before a repeated

game starts. Based on the idea of SA-IGA, we relax these assumptions and propose a practical multiagent learning algorithm called Socially-Aware Policy Gradient Ascent (SA-PGA). The overall flow of SA-PGA is shown in Algorithm 1. In SA-PGA, each agent only needs to observe the payoffs of both agents by the end of each round. In SA-IGA, we know that agent i 's policy (the probability of selec-

Algorithm 1 SA-PGA for player i

- 1: Let $\alpha \in (0, 1)$ and $\delta_p, \delta_w \in (0, 1)$ be learning rates.
 - 2: Initialize $Q_i^{idv}(a) \leftarrow 0$, $Q_i^{op}(a) \leftarrow 0$, $Q_i(a) \leftarrow 0$, $w_i \leftarrow 0.5$, $\pi_i(a) \leftarrow \frac{1}{|A_i|}$.
 - 3: **repeat**
 - 4: Select action $a \in A_i$ according to mixed strategy π_i with suitable exploration.
 - 5: Observing reward r and its opponent's reward r' ,
 $Q_i^{idv}(a) \leftarrow (1 - \alpha) Q_i^{idv}(a) + \alpha r$,
 $Q_i^{op}(a) \leftarrow (1 - \alpha) Q_i^{op}(a) + \alpha r'$,
 - 6: $Q_i(a) \leftarrow (1 - \frac{w}{2}) Q_i^{idv}(a) + \frac{w}{2} Q_i^{op}(a)$,
 - 7: Average payoff $V_i = \sum_{a \in A_i} \pi_i(a) Q_i(a)$
 - 8: **for** each action $a \in A_i$ **do**
 - 9: $\pi_i(a) \leftarrow \pi_i(a) + \delta_p (Q_i(a) - V_i(s))$
 - 10: **end for**
 - 11: $\pi_i \leftarrow \Pi_{\Delta}[\pi_i]$
 - 12: $V_i^{idv} = \sum_{a \in A_i} \pi_i(a) Q_i^{idv}(a)$
 - 13: $V_i^{op} = \sum_{a \in A_i} \pi_i(a) Q_i^{op}(a)$
 - 14: $V_i^{soc} = \frac{1}{2} (V_i^{idv} + V_i^{op})$
 - 15: $w_i \leftarrow w_i + \delta_w (V_i^{idv} - V_i^{soc})$
 - 16: **until** the repeated game ends
-

tion each action) is updated based on the partial derivative of the expected value V_i , while the social attitude w is adjusted according to the relative value of V_i^{idv} and V_i^{soc} . In SA-PGA, we first estimate the value of V_i^{idv} and V_i^{op} using Q-values, which are updated based on the immediate payoffs received during repeated interactions. Specifically, each agent i keeps a record of the Q-value of each action for both its own and its opponent (Q_i^{idv} and Q_i^{op}) (Line 2). Both Q-values are updated according to the Q-learning update rule at the end of each round (Line 5). The overall Q-value of each agent is calculated as the weighted average of Q_i^{idv} and Q_i^{op} , weighted by its social attitude w (Line 6). Based on the Q-values, we estimate the value of V_i in SA-IGA as the expected Q-value over all actions given the current policy (Line 7). However, V_i is simply an estimated value instead of a function which cannot be differentiated. To obtain the derivative of V_i with respect to different actions, we estimate it as the difference between each action's Q-value and the expected Q-value over all actions (the value of V_i) (Line 9). Agent i 's probability of selecting an action is updated in the direction of the estimated derivative of the action's expected value (Line 8-10). After that, agent i 's policy is mapped back to the valid probability space (Line 11). Similarly, the expected individual payoff and its opponent's payoff when agent i plays policy π_i are estimated based on its current policy and Q-values (Line 12-13). The value of V_i^{soc} is calculated as the average between V_i^{idv} and V_i^{op} (Line 14). Finally, the social attitude of agent i is updated in the same way as we introduced in SA-IGA based on the estimated V -values (Line 15). The updating direction of w_i is estimated as the difference between V_i^{idv} and V_i^{soc} .

5 Experimental Evaluation

We start the performance evaluation with analyzing the learning performance of SA-PGA under two-player two-action repeated games.

In general a two-player two-action game can be classified into three categories[21]:

Category 1: $(r_{11}^r - r_{21}^r)(r_{12}^r - r_{22}^r) > 0$ or $(r_{11}^c - r_{12}^c)(r_{21}^c - r_{22}^c) > 0$. In this case, each player has a dominant strategy and thus the game only has one pure strategy NE.

Category 2: $(r_{11}^r - r_{21}^r)(r_{12}^r - r_{22}^r) < 0$ and $(r_{11}^c - r_{12}^c)(r_{21}^c - r_{22}^c) < 0$ and $(r_{11}^r - r_{21}^r)(r_{12}^c - r_{22}^c) > 0$. In this case, there are two pure strategy NEs and one mixed strategy NE.

Category 3: $(r_{11}^r - r_{21}^r)(r_{12}^r - r_{22}^r) < 0$ and $(r_{11}^c - r_{12}^c)(r_{21}^c - r_{22}^c) < 0$ and $(r_{11}^r - r_{21}^r)(r_{12}^c - r_{22}^c) < 0$. In this case, there only exists one one mixed strategy NE.

where r_{ij}^r and r_{ij}^c are payoffs of player **r** and player **c** respectively when player **r** takes action i while player **c** takes action j . We select one representative game for each category for illustration.

5.1 Category 1

For category 1, we consider the PD game as shown in Table 1. In this game, both players have one dominant strategy D , and (D, D) is the only pure strategy NE, while there also exists one socially optimal outcome (C, C) under which both players can obtain higher payoffs.

Figure 1(a) show the learning dynamics of the practical SA-PGA algorithm playing the PD game. The x-axis $p1$ represents player 1's probability of playing action C and the y-axis $p2$ represents player 2's probability of playing action C . We randomly selected 20 initial policy points as the starting point for the SA-PGA agents. We can observe that the SA-PGA agents are able to converge to the mutual cooperation equilibrium point starting from different initial policies.

Figure 1(b) illustrates the learning dynamics predicted by the theoretical SA-IGA approach. Similar to the setting in Figure 1(a), the same set of initial policy points are selected and we plot all the learning curves accordingly. We can see that for each starting policy point, the learning dynamics predicted from the theoretical SA-IGA is well consistent with the learning curves from simulation. This indicates that we can better understand and predict the dynamics of SA-PGA algorithm using its corresponding theoretical SA-IGA model.

5.2 Category 2

For category 2, we consider the CG game as shown in Table 5. In this game, there exist two pure strategy Nash equilibria (C, D) and (D, C) , and both of them are also socially optimal.

Figure 2(a) illustrates the learning dynamics of the practical SA-PGA algorithm playing a CG game. The x-axis $p1$ represents player 1's probability of playing action C and the y-axis $p2$ represents player 2's probability of playing action C . Similar to the case of PD game, 20 initial policy points are randomly selected as the starting points. We can see that the SA-PGA agents can converge to either of the aforementioned two equilibrium points depending on the initial policies they start from.

Figure 2(b) shows the learning dynamics predicted by the theoretical SA-IGA approach. Similar to the setting in Figure 2(a), we adopt the same set of 20 initial policy points for comparison purpose. All the learning curves starting from these 20 policy points are drawn accordingly. We can observe that for each starting policy point, the learning dynamics predicted from the theoretical SA-IGA

is well consistent with the learning curves obtained from simulation. Therefore, the theoretical model can facilitate a better understanding and prediction of the dynamics of SA-PGA algorithm.

1's payoff 2's payoff	Agent 2's actions		
	C	D	
Agent 1's actions	C	3/4	0/0
	D	0/0	4/3

Table 5: Coordination game (Category 2)

5.3 Category 3

The game we use in Category 3 is shown in Table 6. In this game, there only exist one mixed strategy Nash equilibrium, while the pure strategy outcome (C, D) is socially optimal.

Figure 3(a) illustrates the learning dynamics of the practical SA-PGA algorithm playing the game in Table 6. The x-axis $p1$ and y-axis $p2$ represent player 1's probability of playing action C and player 2's probability of playing action C respectively. Similar to the previous cases, 20 initial policy points are randomly selected as the starting points. From Figure 3(a), we can see that the SA-PGA agents can always converge to the socially optimal outcome (C, D) no matter where the initial policies start from.

Figure 3(b) presents the learning dynamics of agents predicted by the theoretical SA-IGA approach. Similar to the setting in Figure 3(a), we adopt the same set of 20 initial policy points for comparison purpose, and the corresponding learning curves are drawn accordingly. From Figure 3(b), we can observe that for each starting policy point, the theoretical SA-IGA model can well predict the simulation results of SA-PGA algorithm. Therefore, a better understanding and insights of the dynamics of SA-PGA algorithm can be obtained through analyzing its corresponding theoretical model.

1's payoff 2's payoff	Agent 2's actions		
	C	D	
Agent 1's actions	C	3/2	4/4
	D	1/3	5/1

Table 6: An example game of Category 3

5.4 Performance in General-sum Games

In this section we turn to evaluate the performance of SA-PGA with previous representative learning strategies CJAL [3] and WoLF-PHC [5] in two-player's repeated games under self-play. CJAL is selected since this algorithm is specifically designed to enable agents to achieve mutual cooperation (i.e., maximizing social welfare) instead of inefficient NE for games like prisoner's dilemma. WoLF-PHC is selected as one representative NE-oriented algorithm for baseline comparison purpose. For all previous strategies the same parameter settings as communicated in their original papers are adopted.

We use all possible structurally distinct two-player, two-action conflict games as a testbed for SA-PGA. In each game, each player ranks the four possible outcomes from 1 to 4. We use the rank of an

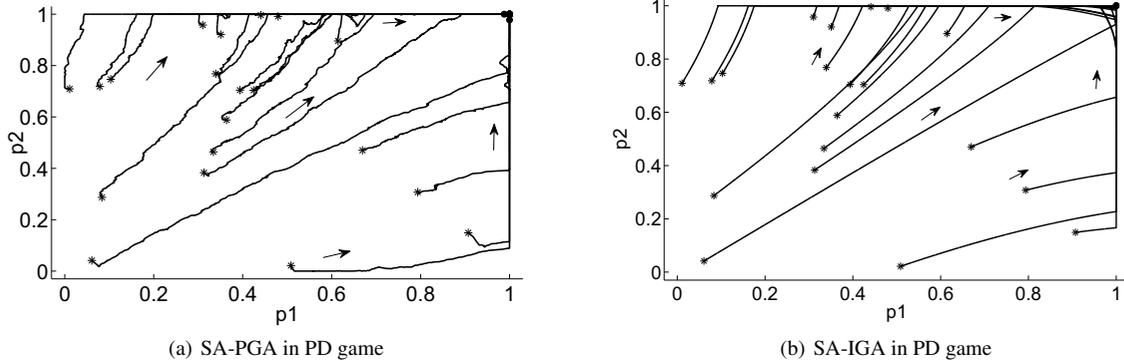


Figure 1: The Learning Dynamics of SA-IGA and SA-PGA in PD game (parameter $w_r(0) = w_c(0) = 0.85$, $\delta_p = 0.001$, $\alpha = 0.8$ and $\varepsilon = 0.02$)

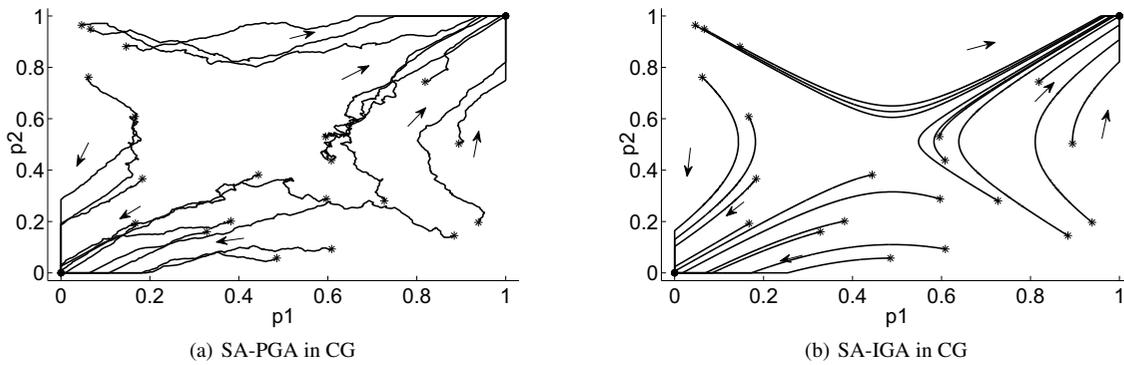


Figure 2: The Learning Dynamics of SA-IGA and SA-PGA in coordination game (parameter $w_r(0) = w_c(0) = 0.85$, $\delta_p = 0.001$, $\alpha = 0.8$ and $\varepsilon = 0.02$)

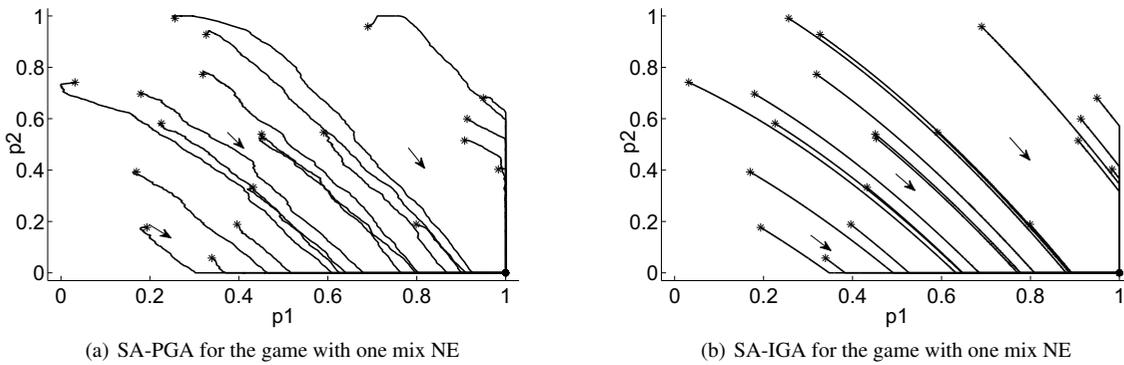


Figure 3: The Learning Dynamics of SA-IGA and SA-PGA in game with one mix NE (parameter $w_r(0) = w_c(0) = 0.85$, $\delta_p = 0.001$, $\alpha = 0.8$ and $\varepsilon = 0.02$)

outcome as the payoff to that player for any outcome. We perform the evaluation under 100 randomly generated games with strict ordinal payoffs. We perform 10,000 interactions for each run and the results are averaged over 20 runs for each game.

We compare their performance based on the the following two criteria: utilitarian social welfare and Nash social welfare. Utilitarian social welfare is the sum of the payoffs obtained by the two players in their converged state, averaged over 100 randomly generated games. Nash social welfare is the product of the payoffs obtained by two players in their converged state, averaged over 100 randomly generated games. Both criteria reflect the system-level efficiency of different learning strategies in terms of the total payoffs received for the agents. Besides, Nash social welfare also partially reflects the fairness in terms of how equal the agents' payoffs are. The overall comparison results are summarized in Table 7. We can see that SA-IGA outperforms the previous CJAL strategy under both criteria. The WoLF-PHC strategy is designed to achieve NE and thus can only achieve the same level of performance as adopting NE solutions.

Table 7: Performance comparison with CJAL and WoLF-PHC

	Utilitarian Social Welfare	Nash Product
SA-PGA (our strategy) ($w_r(0) = w_c(0) = 0.85$)	7.241 ± 0.003	12.706 ± 0.015
CJAL [3]	6.504 ± 0.032	10.887 ± 0.114
WoLF-IGA [5]	6.536 ± 0.004	10.943 ± 0.145

5.5 Against Selfish Agents

If a learning agent is facing selfish agents that attempt to exploit others, one reasonable choice for an effective algorithm is to learn a Nash equilibrium. In this section, we evaluate the ability of SA-PGA against selfish opponents. We adopt the same three representative games used in previous sections as the testbed and the results are given in Figure 4, 5 and 6 respectively. We can observe that for the PD and coordination games, the SA-PGA agent can successfully achieve the corresponding NE solution. This property is desirable since it prevents the SA-PGA agent from being taken advantage of by selfish opponents. The results also show how the socially-aware degree w of SA-PGA agent changes, which varies depending on the game structure. For PD and coordination game, a SA-PGA agent eventually behaves as a purely individually rational entity and one pure strategy NE is eventually converged to. In contrast, for the third type of game (Table 6), a SA-PGA agent behaves as a purely socially rational agent and cooperates with the selfish agent towards the socially optimal outcome (C, D) without fully exploiting the opponent. This indicates the cleverness of SA-PGA algorithm since higher individual payoff can be achieved under the outcome (C, D) than pursuing the Nash equilibrium (C, C) .

6 Conclusion and Future Work

In this paper, we proposed a novel way of incorporating social awareness into traditional gradient-ascent algorithms to facilitate reaching mutually beneficial solutions (e.g., (C, C) in PD game). We presented a theoretical gradient-ascent based policy updating approach (SA-IGA) and analyzed its learning dynamics using dynamical system theory. For PD games, we showed that mutual cooperation (C, C) is a stable equilibrium point as long as both agents are strongly socially-aware. For CG games, either of the Nash equilibria (C, C) and (D, D) can be a stable equilibrium point depending on the agents' socially-aware degrees. Following that, we proposed a practical learning algorithm SA-PGA relaxing the impractical assumptions of SA-IGA.

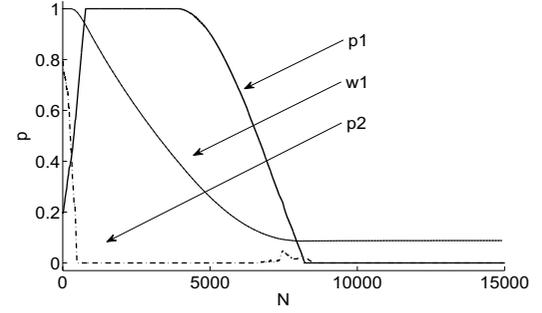


Figure 4: SA-PGA against a selfish agent for in PD game ($w_r(0) = 1$, $p_r(0) = 0.2$ and $p_c(0) = 0.8$)

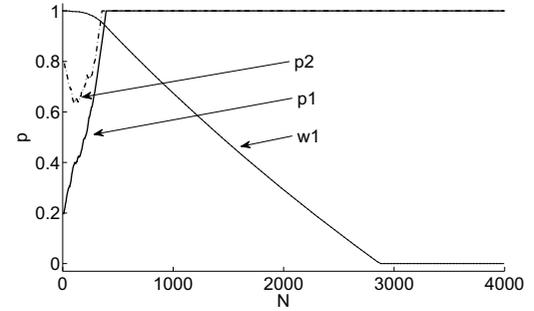


Figure 5: SA-PGA against a selfish agent for in coordination game ($w_r(0) = 1$, $p_r(0) = 0.2$ and $p_c(0) = 0.8$)

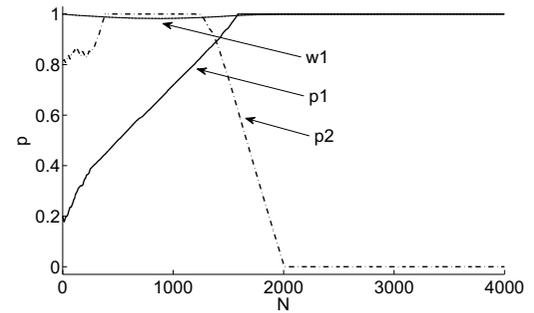


Figure 6: SA-PGA against a selfish agent for the game with only one mix NE ($w_r(0) = 1$, $p_r(0) = 0.2$ and $p_c(0) = 0.8$)

Experimental results show that a SA-PGA agent can achieve higher social welfare than previous algorithms under self-play and also is robust against individually rational opponents. As future work, more testbed scenarios (e.g., population of agents) will be applied to further evaluate the performance of SA-PGA. Another interesting direction is to investigate how to further improve the convergence rate of SA-PGA.

7 ACKNOWLEDGEMENTS

This work has partially been sponsored by the National Science Foundation of China (No. 61572349, 61272106) and Tianjin Research Program of Application Foundation and Advanced Technology (No.: 16JCQNJC00100)

REFERENCES

Artificial Intelligence, pp. 927–934, (2010).

- [1] Sherief Abdallah and Victor Lesser, 'A multiagent reinforcement learning algorithm with non-linear dynamics', *Journal of Artificial Intelligence Research*, 521–549, (2008).
- [2] Ingela Alger and Jörgen W Weibull, 'Homo moralis|preference evolution under incomplete information and assortative matching', *Econometrica*, **81**(6), 2269–2302, (2013).
- [3] D. Banerjee and S. Sen, 'Reaching pareto optimality in prisoner's dilemma using conditional joint action learning', *AAMAS'07*, 91–108, (2007).
- [4] Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers, 'Evolutionary dynamics of multi-agent learning: a survey', *Journal of Artificial Intelligence Research*, 659–697, (2015).
- [5] M. H. Bowling and M. M. Veloso, 'Multiagent learning using a variable learning rate', *Artificial Intelligence*, 215–250, (2003).
- [6] Lucian Busoniu, Robert Babuska, and Bart De Schutter, 'A comprehensive survey of multiagent reinforcement learning', *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, **38**(2), 156–172, (2008).
- [7] Doran Chakraborty and Peter Stone, 'Multiagent learning in the presence of memory-bounded agents', *Autonomous agents and multi-agent systems*, **28**(2), 182–213, (2014).
- [8] Levinson N Coddington E A, *Theory of ordinary differential equations*, McGraw-Hill, 1955.
- [9] Vincent Conitzer and Tuomas Sandholm, 'Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents', *Machine Learning*, **67**(1-2), 23–43, (2007).
- [10] Junling Hu and Michael P Wellman, 'Nash q-learning for general-sum stochastic games', *The Journal of Machine Learning Research*, **4**, 1039–1069, (2003).
- [11] M. Lauer and M. Rienmiller, 'An algorithm for distributed reinforcement learning in cooperative multi-agent systems', in *ICML'00*, pp. 535–542, (2000).
- [12] M. Littman, 'Markov games as a framework for multi-agent reinforcement learning', in *Proceedings of the 11th international conference on machine learning*, pp. 322–328, (1994).
- [13] Michael L Littman, 'Friend-or-foe q-learning in general-sum games', in *ICML*, volume 1, pp. 322–328, (2001).
- [14] Jason R Marden, H Peyton Young, and Lucy Y Pao, 'Achieving pareto optimality through distributed learning', *SIAM Journal on Control and Optimization*, **52**(5), 2753–2770, (2014).
- [15] Laetitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat, 'Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems', *The Knowledge Engineering Review*, **27**(01), 1–31, (2012).
- [16] Rob Powers and Yoav Shoham, 'Learning against opponents with bounded memory.', in *IJCAI*, volume 5, pp. 817–822, (2005).
- [17] Bary SR Pradelski and H Peyton Young, 'Learning efficient nash equilibria in distributed systems', *Games and Economic behavior*, **75**(2), 882–897, (2012).
- [18] Eduardo Rodrigues Gomes and Ryszard Kowalczyk, 'Dynamic analysis of multiagent q-learning with ϵ -greedy exploration', in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 369–376. ACM, (2009).
- [19] Leonid P Shilnikov, Andrey L Shilnikov, Dmitry V Turaev, and Leon O Chua, *Methods of qualitative theory in nonlinear dynamics*, volume 5, World Scientific, 2001.
- [20] Satinder Singh, Michael Kearns, and Yishay Mansour, 'Nash convergence of gradient dynamics in general-sum games', in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 541–548, (2000).
- [21] Karl Tuyls, Pieter JanT Hoen, and Bram Vanschoenwinkel, 'An evolutionary dynamical analysis of multi-agent learning in iterated games', *Autonomous Agents and Multi-Agent Systems*, **12**(1), 115–153, (2006).
- [22] Karl Tuyls, Katja Verbeeck, and Tom Lenaerts, 'A selection-mutation model for q-learning in multi-agent systems', in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pp. 693–700. ACM, (2003).
- [23] C. J. C. H. Watkins and P. D. Dayan, 'Q-learning', *Machine Learning*, 279–292, (1992).
- [24] Chongjie Zhang and Victor R Lesser, 'Multi-agent learning with policy prediction', in *Proceedings of the Twenty-Fourth AAAI Conference on*

Factors of Collective Intelligence: How Smart Are Agent Collectives?

Nader Chmait and David L. Dowe and Yuan-Fang Li and David G. Green¹ and Javier Insa-Cabrera²

Abstract. The dynamics and characteristics behind intelligent cognitive systems lie at the heart of understanding, and devising, successful solutions to a variety of multiagent problems. Despite the extant literature on collective intelligence, important questions like “how does the effectiveness of a collective compare to its isolated members?” and “are there some general rules or properties shaping the spread of intelligence across various cognitive systems and environments?” remain somewhat of a mystery. In this paper we develop the idea of collective intelligence by giving some insight into a range of factors hindering and influencing the effectiveness of interactive cognitive systems. We measure the influence of each examined factor on intelligence independently, and empirically show that collective intelligence is a function of them all simultaneously. We further investigate how the organisational structure of equally sized groups shapes their effectiveness. The outcome is fundamental to the understanding and prediction of the collective performance of multiagent systems, and for quantifying the emergence of intelligence over different environmental settings.

1 INTRODUCTION

Collective intelligence emerges in all sorts of cognitive systems, from natural (e.g., animal and human) to artificial (e.g., software agents and robotics), by cause of diverse social organizations (human societies, efficient markets, social insect colonies, group collaborations via the web, etc.). It seems that the complex structure and operation of these systems hinder our understanding of the dynamics and characteristics behind intelligent collectives, which are fundamental for devising successful models and solutions to a variety of multiagent problems. Despite the extant literature on collective intelligence (CI), the questions consisting of, “how does the effectiveness of a collective compare to its isolated members?” and, more importantly, “are there some general rules shaping the spread of intelligence which can be perceived across different cognitive systems and environments?” remain somewhat of a mystery. Now imagine we had a series of performance tests over which we can administer any type of cognitive system, could we then disclose any patterns or factors at all, explaining the emergence of intelligence among all of these systems? In this paper, we give insight into the main components and characteristics of collective intelligence, by applying formal tests for the purpose of measuring and quantifying the influence of several factors on the collective behaviour and the accuracy of a group of agents, and analysing how the results compare to individual agent scenarios. We attempt to uncover some of the dynamics and circumstances behind

intelligent collectives in general, hoping this would reinforce the understanding and prediction of the behaviour of groups, by bringing some new results into the AI community.

2 BACKGROUND

Earlier studies [13, 43] have revealed that a collective intelligence factor can emerge in human groups. We know that collectives can outperform individuals, and further that their performance is controlled by one or more of a) their organisational or network structure [29, 3, 30], b) the information aggregation details among their individuals [1], and c) the diversity between their members [20, 17]. Crowd-computing and crowd-sourcing [32, 24, 2] methodologies are excellent examples of CI that harness the wisdom of the crowd [37].

After carefully looking at the literature on collective intelligence including the abovementioned works and others including [28, 42, 38, 8, 41], we filter a set of factors or features from these works - that are not coupled to one particular cognitive system, problem or environment - which are intimately relevant to the performance of collectives, some of which are the number of members in a group, the communication or interaction protocol, as well as the difficulty of the environment. Curiously, there are some other factors which are often relatively neglected, such as the reasoning/learning speed of the agents and the interaction time of the collective as a whole. These features, in addition to some hypothetical combinations of them (grouped in ellipses) are depicted in Figure 1. It is not known *in which circumstances* and *how much* each these features individually influences the intelligence of the group, let alone the simultaneous influence of multiple features combined, which is what we attempt to *quantitatively* investigate in this paper.

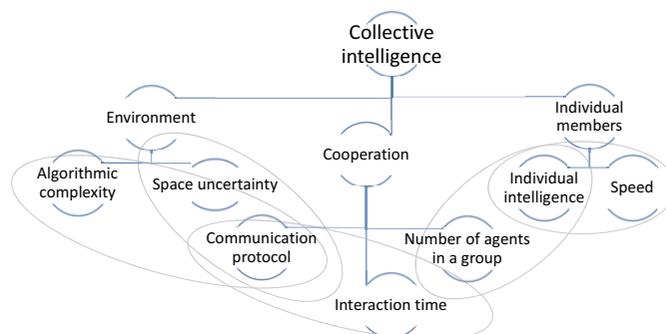


Figure 1: Factors and features relevant to the notion of collective intelligence (CI) perceived throughout various cognitive systems, and some hypothetical relationships between them grouped in ellipses.

¹ Faculty of IT, Clayton, Monash University, Australia, email: {nader.chmait,david.dowe,yuanfang.li,david.green}@monash.edu

² Universitat Politècnica de València, Spain, email: jinsa@dsic.upv.es

The next section introduces our methodology for assessing indi-

vidual and collective agent performances. The agent behaviours to be evaluated and their communication and interaction protocols are described in Sections 4 and 5 respectively. After we present our experimental setup in Section 6, we discuss and analyse our results from these experiments (in Sections 7 and 8) by making a series of observations on how the intelligence of the evaluated agents was influenced by a collection of factors, and draw some interesting conclusions connecting the research outcomes. We conclude in Section 9 by a brief summary and give some directions for future work.

3 EVALUATING INTELLIGENCE

To achieve our aims, we need a dynamic environment in which we can assess the influence of the factors appearing in Figure 1 on the performance of various types of cognitive systems over different environmental settings. While many environments could be appropriate, we have chosen for our purpose the *Anytime Universal Intelligence Test* (ANYNT) [18], which is derived from formal information theoretic backgrounds that have been practically used to evaluate *diverse kinds of entities* [21, 5, 6, 22], and was proven [18] to be an unbiased, dynamic setting which can be stopped at anytime.

3.1 The Anytime Universal Intelligence Test

We introduce the Λ^* (Lambda Star) environment class that focuses on a restricted - but important - set of tasks in AI. This environment extends the Λ environment class [18, Sec. 6][23] which implements the theory behind the Anytime Universal Intelligence Test [18]. The general idea is to evaluate an agent that can perform a set of actions, by placing it in a grid of cells with two special objects, *Good* (\oplus) and *Evil* (\ominus), travelling in the space using movement patterns of measurable complexities. Rewards are defined as a function of the position of the evaluated agent with respect to the positions of \oplus and \ominus .

3.1.1 Structure of the test

We generate an environment space as an m -by- n grid-world populated with objects from $\Omega = \{\pi_1, \pi_2, \dots, \pi_x, \oplus, \ominus\}$, the finite set of objects. The set of evaluated agents $\Pi \subseteq \Omega$ is $\{\pi_1, \pi_2, \dots, \pi_x\}$. Each element in Ω can have actions from a finite set of actions $\mathcal{A} = \{left, right, up, down, up-left, up-right, down-left, down-right, stay\}$. All objects can share the same cell at the same time except for \oplus and \ominus where in this case, one of them is randomly chosen to move to the intended cell while the other one keeps its old position. In the context of the agent-environment framework [25], a test episode consisting of a series of ϑ iterations is modelled as follows:

1. the environment space is first initialised to an m -by- n toroidal grid-world, and populated with a subset of evaluated agents from $\Pi \subseteq \Omega$, and the two special objects \oplus and \ominus ,
2. the environment sends to each agent a description of its range of 1 Moore neighbour cells [16, 40] and their contents, the rewards in these cells, as an observation,
3. the agents (communicate/interact and) respond to the observations by performing an action in \mathcal{A} , and the special objects perform the next action in their movement pattern,
4. the environment then returns a reward to each evaluated agent based on its position (distance) with respect to the locations of the special objects,
5. this process is repeated again from point #2 until a test episode is completed.

We are using a toroidal grid space in the sense that moving off one border makes an agent appear on the opposite one. Consequently, the distance between two agents is calculated using the surpassing rule (toroidal distance) such that, in a 5-by-5 grid space for example, the distance between cell (1, 3) and (5, 3) is equal to 1 cell.

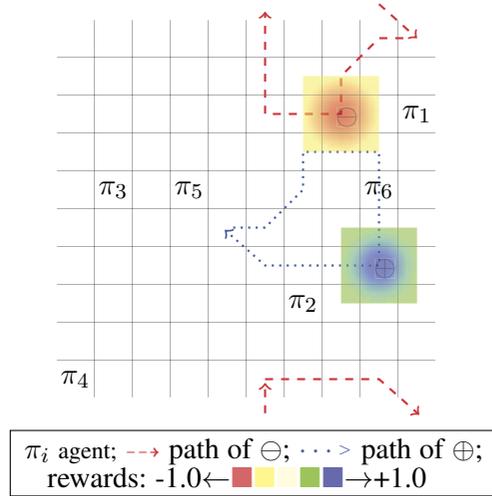


Figure 2: A diagrammatic representation of a sample 10-by-10 Λ^* environment, used to implement the theory behind the Anytime Universal Intelligence Test [18].

3.1.2 Rewarding function

The environment sends a reward to each evaluated agent from the set of rewards $\mathcal{R} \subseteq \mathbb{Q}$ where $-1.0 \leq \mathcal{R} \leq 1.0$.

Let $d(a, b)$ denote the (toroidal) distance between two objects a and b . Given an agent π_j , its positive reward at one test iteration is calculated as: $1/(d(\pi_j, \oplus) + 1)$ if $d(\pi_j, \oplus) < 2$, or 0 otherwise. Likewise its negative reward at that iterations is: $-1/(d(\pi_j, \ominus) + 1)$ if $d(\pi_j, \ominus) < 2$, or 0 otherwise. Its total reward, r_j^i at iteration i , is the sum of its positive and negative rewards at that iteration.

3.2 Algorithmic Complexity

We regard the Kolmogorov complexity [27] of the movement patterns of the special objects as a measure of the algorithmic complexity $K(\mu)$ of the environment μ in which they operate. For instance, a Λ^* environment of high Kolmogorov complexity is sufficiently rich and structured to generate complicated (special object) patterns/sequences of seeming randomness. We measure the Lempel-Ziv complexity [26] of the movement patterns as an approximation to $K(\mu)$ as suggested in [26, 14]. Note that, at one test episode, the movement patterns of \oplus and \ominus are different but (algorithmically) equally complex. The recurrent segment of the movement pattern is at least of length one and at most $\lfloor \vartheta/2 \rfloor$, cyclically repeated until the final iteration (ϑ) of the test.

3.3 Search Space Complexity

We measure the search space complexity $\mathcal{H}(\mu)$ as the amount of uncertainty in μ , expressed by Shannon's entropy [35]. Let N be the set of all possible states of an environment μ such that a state s_μ , is the set holding the current positions of the special objects

$\{\oplus, \ominus\}$ in the m -by- n space. Thus the number of states $|N|$ increases with the increase in the space dimensions m and n , and it is equal to the number of permutation $m \times n P_2 = \frac{(m \times n)!}{(m \times n - 2)!}$. The entropy is maximal at the beginning of the test as, from an agent's perspective, there is complete uncertainty about the current state of μ . Therefore $p(s_\mu)$ follows a uniform distribution and is equal to $1/|N|$. Using \log_2 as a base for our calculations, we end up with: $\mathcal{H}(\mu) = -\sum_{s_\mu \in N} p(s_\mu) \log_2 p(s_\mu) = \log_2 |N|$ bits.

Despite the test's being originally designed to return a general measure of intelligence, we do not make this assumption in this paper. Nevertheless, we appraise the test, at a minimum, as an accurate measure of the testee's ability of performing over a class of: inductive inference, compression and search problems, all of which are particularly related to intelligence [9, 10, 11, 19, 34, 12]. Note, however, that we will use the term *intelligence* to describe the effectiveness or accuracy of an evaluated agent over this test. It is of great importance that the illustrative class of problems assessed by the test is shared across, and applies to, various types of cognitive systems since this meets our criteria for the evaluation, as raised in the introduction.

4 AGENT TYPES AND BEHAVIOURS

We evaluated agents of five different behaviours, both in isolation and collectively in (cooperative) groups over the Λ^* environment. A description of these agents is given in the following paragraphs.

Local search agents: given an agent π_j , we denote by c_j^i and $r(c_j^i)$ the cell where π_j is located at iteration i , and the reward in this cell respectively. Let N_j^i and $R(N_j^i)$ denote respectively the set of range of 1 *Moore* neighbour cells [16, 40] of agent π_j (including c_j^i) at iteration i , and the reward values in these cells. $R(c_j^i, a)$ is a function that returns the reward agent π_j gets after performing action $a \in \mathcal{A}$ when it is in cell c_j^i . The behaviour of local search agents is defined as follows: $a_j^i \leftarrow \arg \max_{a \in \mathcal{A}} R(c_j^i, a)$. If all actions return an equivalent reward, then a random action in \mathcal{A} is selected.

Reinforcement learning agents: two of the most frequently used RL (reinforcement learning) behaviours are Q-learning [39] and Sarsa [33, 39]. In the Q-learning behaviour, agents learn using an action-quality function in order to find the best action-selection policy for a given MDP (Markov Decision Process). Alternatively, Sarsa agents learn a MDP policy using an on-policy temporal-difference learning technique. Before learning starts, we initialise the elements of the Q-table to 2.0 so that the quality of a state-action pair, $Q \leftarrow S \times \mathcal{A}$, is always positive despite that rewards fall in the range $[-1.0, 1.0]$. Because the testing environment is dynamic, each state in S was designated to be the unique combination of one cell position c at one iteration i of the test, leading to a total number of states³ $|S| = (m \times n)^\vartheta$. Before evaluation, we trained the RL agents for 100 rounds previous to each episode using both a discount factor γ and a learning rate α of 0.30, selected after fine-tuning these parameters on a single agent scenario to reach a general (average) optimal payoff. Our agents learn offline, and thus cease to update their Q-table once their training is complete.

Oracle agents: an *oracle* agent knows the future movements of \oplus , the *Good* special object. At each step i of an episode this agent approaches the subsequent $i + 1$ cell destination of \oplus seeking maximum payoff. However, if \oplus has a constant movement pattern (e.g., moves constantly to the right) pushing it away from the oracle, then the oracle will move in the opposite direction in order to intercept

\oplus in the upcoming test steps. Once it intercepts \oplus , it then continues operating using its normal behaviour.

Random agents: a random agent randomly chooses an action from the finite set of actions \mathcal{A} at each iteration until the end of an episode.

The scores of the random and oracle agents will be used as a baseline for our experiments, where a random agent is used as a lower bound on performance while the oracle is used as an upper bound.

5 COMMUNICATION PROTOCOLS

The agents were also evaluated collectively in groups. A description of the interaction and communication protocols used in these collectives are given below.

Stigmergy or indirect communication: we propose a simple algorithm for enabling communication between local search agents using stigmergy [15] (indirect communication). For instance, we let the agents induce fake rewards in the environment, thus indirectly inform neighbour agents about the proximity of the special objects. Note that fake rewards will not affect the score (real reward payoff) of the agents. Let $\hat{R}(N_j^i)$ denote the set of fake rewards in the neighbour cells of agent π_j (including c_j^i) at iteration i , and $\hat{R}(c_j^i, a)$ is a function returning the fake reward agent π_j gets after performing action $a \in \mathcal{A}$ when it is in cell c_j^i at iteration i . Fake rewards are induced in the environment according to Algorithm 1. Each agent proceeds

Algorithm 1 Stigmergic or indirect communication: fake reward generation over one iteration i of the test.

```

1: Input:  $\Pi$  (set of evaluated agents),  $0 < \gamma < 1$  (fake reward discounting factor), a test iteration  $i$ .
2: Initialize:  $\forall \pi_j \in \Pi: \hat{R}(N_j^i) \leftarrow 0.0$ .
3: Begin
4:   for  $j \leftarrow 1$  to  $|\Pi|$  do ▷ loop over agents
5:      $r^{max} \leftarrow \max R(N_j^i)$ 
6:      $r^{min} \leftarrow \min R(N_j^i)$ 
7:      $\hat{r} \leftarrow \gamma(r^{max} + r^{min})$  ▷ average expected reward
8:      $\hat{R}(N_j^i) \leftarrow R(N_j^i) + \hat{r}$ 
9:   end for
10: End

```

by selecting an action by relying on fake rewards this time instead of the real rewards, as follows: $a_j^i \leftarrow \arg \max_{a \in \mathcal{A}} \hat{R}(c_j^i, a)$. If all actions are equally rewarding, then a random action is selected. Thereupon, we expect local search agents using stigmergy to form non-strategic coalitions after a few iterations of the test as a result of tracing the most elevated fake rewards in the environment.

Implicit leadership through auctions and bidding: in this cooperative setting, local search agents go into a *single dimensional English auction* [31] at each iteration i , and bid on the right to lead the other agents in their group by appointing one target cell to be approached. At each iteration, each auctioneer (agent) generates a value of the maximum reward existing in its neighbourhood, which is then used as its bidding “money” for the auction. The richest agent⁴ wins the auction visibly to all the other agents. It then selects the *target cell* to be approached by all other agents in the collective. This bidding behaviour is described in Algorithm 2 in which $n_j^i \in N_j^i$ and $r(n_j^i)$ denote one of the *Moore* neighbour cells of agent π_j (without excluding c_j^i) at iteration i , and the reward in this cell respectively.

Imitating super-solver agents: a group of isolated local search agents is put in the same space with one (unevaluated) oracle agent.

³ Recall that m and n refer to the grid space dimensions, while ϑ is the number of iterations in a single test episode.

⁴ If more than one agent are equally rich then, for the sake of simplicity, the last one to participate in the auction wins.

Algorithm 2 Single dimensional English auction at one iteration i of the test.

```

1: Input:  $\Pi$  (set of evaluated agents),  $-1.0 < \text{bid} < 1.0$ , a test iteration  $i$ .
2: Initialize:  $\text{bid} \leftarrow -1.0$ 
3: Begin
4:   for  $j \leftarrow 1$  to  $|\Pi|$  do ▷ loop over agents
5:      $\text{money} \leftarrow \max R(N_j^i)$ 
6:     if  $\text{money} >= \text{bid}$  then
7:        $\text{bid} \leftarrow \text{money}$ 
8:        $\text{target} \leftarrow \arg \max_{n_j^i \in N_j^i} r(n_j^i)$  ▷ set the target to the neighbour
           cell  $n_j^i$  holding the highest reward  $r(n_j^i)$  at iteration  $i$ 
9:     end if
10:  end for
11: End

```

Local search agents imitate the oracle by following it into the same cell only when it is in their visibility range (neighbourhood) otherwise, they operate using their normal behaviour.

Wisdom of the crowd (WOC) by information aggregation: where the collective opinion of the evaluated agents is aggregated from the opinions of all the members of the collective.

In the case of reinforcement learning collectives, we let their members share and update a common Q-table, thus making them all learn and coordinate simultaneously. We evaluated both Q-learning and Sarsa collectives independently.

In the case of local search collectives, the observations of all agents in the collective are aggregated into one global observation (and rewards from these observations are averaged in the case of overlap). Then, each member proceeds by selecting the action maximising its reward in line with the global observation.

6 EXPERIMENTAL SETUP

Each experiment consists of 1000 episodes (runs) of the test, each consisting of a number of iterations equal to 50. In each episode, agents are administered over a different task with complexity $K(\mu)$, such that $K(\mu) \in [2, 23]$, where a $K(\mu)$ of 23 corresponds to a, more or less, complex pattern prediction or recognition task. Moreover, in each episode, the collectives are re-initialised with different spatial (network) arrangements between their members.

Local search agents were evaluated in isolation as well as collectively using four communication or interaction protocols: stigmergy, implicit leadership, imitation (of the oracle agent) and harnessing the WOC through information aggregation. Likewise, reinforcement learning agents were evaluated in isolation and collectively by harnessing the wisdom of the crowd (WOC) through sharing and updating a common Q-table.

Test experiments were conducted over different search space uncertainties $\mathcal{H}(\mu)$, and the (intelligence) scores (in the range $[-1.0, 1.0]$) of the evaluated agents/collectives averaged over the 1000 episodes were recorded. The score of the collective is calculated as the mean of the scores of its members. For instance, the metric of (individual agent) universal intelligence defined in [18, Definition 10] was extended into a collective intelligence metric (Definition 2) returning an average reward accumulation per-agent measure of success (Definition 1) for a group of agents Π , over a selection of Λ^* environments.

Definition 1 Given a Λ^* environment μ and a set of (isolated or interactive) agents $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ to be evaluated, the (average per-agent per-iteration) reward $\tilde{R}_{\Pi, \mu, \vartheta}$ of Π over one test episode of ϑ iterations is calculated as: $\tilde{R}_{\Pi, \mu, \vartheta} = \frac{\sum_{j=1}^n \sum_{i=1}^{\vartheta} r_j^i}{n\vartheta}$.

Definition 2 The (collective) intelligence of a set of agents Π is calculated as: $\frac{1}{\omega} \sum_{\mu \in L} \tilde{R}_{\Pi, \mu, \vartheta}$, where L is a set of ω environments $\{\mu_1, \mu_2, \dots, \mu_\omega\}$ such that $\forall \mu_t, \mu_q \in L: \mathcal{H}(\mu_t) = \mathcal{H}(\mu_q)$, and $\forall \mu_i \in L, K(\mu_i)$ is extracted from a range of (special object movement patterns with) algorithmic complexities in $]1, K_{max}]$.

7 RESULTS AND DISCUSSION

Table 1: Intelligence test scores for collectives of 10 agents across different environment uncertainties $\mathcal{H}(\mu) \in [13.2, 19.6]$ bits, evaluated for 50 test-iterations. A plot of these results is also found in Figure 3.

	$\mathcal{H}(\mu)$ value in bits	13.2	15.6	17.2	18.5	19.6
1	Random agent	-0.00079	0.00048	0.00008	-0.00013	0.00002
2	Local search (LS) agent	0.3365	0.1696	0.0936	0.0575	0.0423
3	LS collective using stigmergy	0.4025	0.2555	0.1431	0.0829	0.0579
4	LS collective harnessing the WOC	0.3828	0.3475	0.3118	0.2601	0.2110
5	LS collective using implicit leadership	0.3744	0.2842	0.2143	0.1722	0.1438
6	LS collective using imitation	0.5729	0.2880	0.1666	0.1022	0.0731
7	Q-learning agent	0.2516	0.0950	0.0484	0.0301	0.0207
8	Q-learning collective harnessing the WOC	0.4030	0.1832	0.0870	0.0482	0.0309
9	Sarsa agent	0.2708	0.1007	0.0501	0.0308	0.0228
10	Sarsa collective harnessing the WOC	0.4511	0.2042	0.1010	0.0563	0.0348
11	Oracle agent	0.8207	0.7905	0.7619	0.7339	0.7059

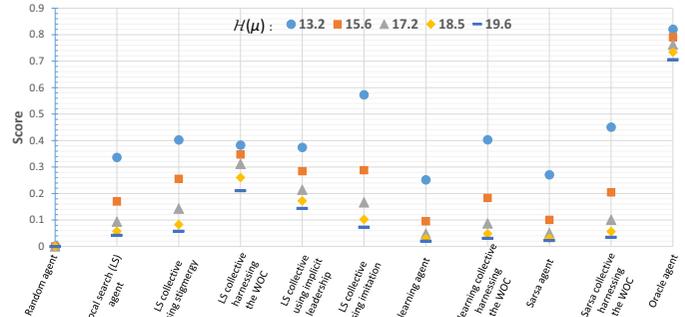


Figure 3: A plot of the test scores appearing in Table 1.

Sample results from the previously-mentioned experiments run in different environment (search space) uncertainties $\mathcal{H}(\mu)$ are listed in Table 1 for isolated agents and collectives Π , each having a number of agents or members $|\Pi| = 10$ agents. The standard deviation of the test scores σ is less than 0.001 between identical experiments.

7.1 Collectives outperform individuals

Results in Figure 3 clearly show (in at least three separate cases) that cooperative or interactive individuals can be more effective than isolated ones. (From Definition 1, the score of the whole is more than the sum of its parts.) This is consistent with earlier results (e.g., [29, 1]) for obvious reasons owing to diffusion of information (synergy) leading to the reduction of uncertainty inside the collective.

Yet the question remains, what are the dynamics which have led to such results? We recall our main aims, which consist of investigating and quantifying the influence of a list of factors on (individual and collective) intelligence. We address each of these factors in detail in the remainder of this paper.

7.2 Communication and interaction protocol

We observe in Figure 3 that the effectiveness of the (same selection of) agents is highly dependent on the collective decision-making technique or the communication protocol used to aggregate the information received from these agents. For instance, adopting auctions in local search collectives to claim leadership can be more effective than using stigmergy over some settings. Figure 3 also shows that, under certain circumstances, introducing heterogeneity in a group of local search agents by imitating a (super-solver) oracle agent leads to more effective coalitions that outperform their homogeneous (and isolated) peers by aggregating new information into the collective. However, the comparison between local search collectives is rather more complicated as their intelligence measures seem to further depend on the uncertainty of the testing environment, and not only on the interaction protocol. We also observe that harnessing the wisdom of the crowd by aggregating the observations of local search agents is very effective over highly uncertain environments, yet not exceptionally efficient in the opposite situation. The latter protocol seems to be very robust (in comparison to others) with respect to the changes in the uncertainty of the search space. A more thorough analysis on the efficiency of the examined communication protocols over different problem uncertainties is addressed in the following subsection.

In the case of RL agents, we observe that agents of different types (Q-learning and Sarsa) using the same cooperation technique to aggregate their information have achieved different scores. Sarsa agents outperform Q-learning agents up to about a similar extent both in cooperative and isolated settings. This indicates that the collective intelligence of the group also depends on, and is correlated with, the individual intelligence (or the type) of the agents in the group.

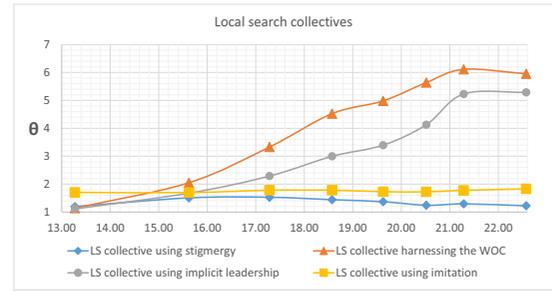
Furthermore, despite the broad differences in the interaction protocols and the wide range of task complexities, collective intelligence manifested across the various collectives, showing that CI can also emerge in a non-human context or environment, thus reinforcing and adding to the conclusions of [13] conclusions.

7.3 Uncertainty in the environment

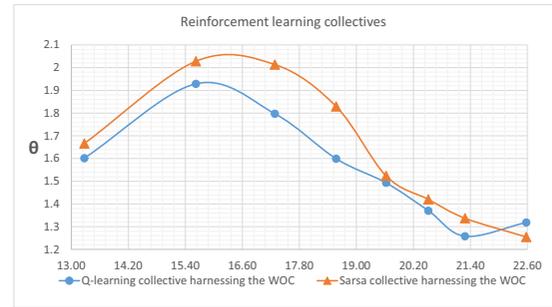
Figure 3 shows that the performance of the evaluated agents decreases with the increase in uncertainty⁵ $\mathcal{H}(\mu)$, in accordance with former tests [22] that have been applied on humans and artificial agents. Moreover, the gap between the scores of the isolated and cooperative agents varies in view of the uncertainty in the environment, but the relationship between both variables cannot be easily grasped from the figure.

We wish to measure the variation in the weight of the cooperative agents' scores to their score in the isolated setting, across different environment uncertainties. Therefore, we define the *coefficient of effectiveness* $\theta = \alpha/\beta$, as the ratio of the score of a set of agents Π working in some cooperative scenario ($\alpha = \text{score}(\Pi^{\text{coop}})$), to its score in the isolated scenario ($\beta = \text{score}(\Pi^{\text{isolated}})$). We calculated θ for the different agent types across different uncertainties and plotted the results in Figure 4.

Figure 4a shows that the θ values corresponding to local search groups using imitation ($\theta^{\text{imitation}}$) and those relying on stigmergy ($\theta^{\text{stigmergy}}$) are more or less steady across the different $\mathcal{H}(\mu)$ values, implying that each of these protocols is approximately equally advantageous over different problem uncertainties. In addition, $\theta^{\text{imitation}} > \theta^{\text{stigmergy}}$ over the selected uncertainties, and thus collectives relying on imitation are more effective than those



(a) Local search (LS) collectives.



(b) Reinforcement learning (RL) collectives.

Figure 4: Shift in effectiveness θ for local search and RL agents over different environment uncertainties in bits.

using stigmergy. The observations are more interesting for collectives using auctions to claim leadership. For instance, in environments of uncertainties lower than 16 bits, imitating a smart agent is more advantageous than following a leader. Whereas, the inverse is true for environments of higher uncertainties. The effectiveness of local search agents using auctions significantly increases to become much higher than that of the same group of agents imitating an oracle. Similar results are observed for local search collectives harnessing the wisdom of the crowd. We conclude that relying on the best (super-solver) agent in the groups does not guarantee an optimal performance. This is somewhat consistent with [20]'s claims re diversity vs. ability, even though the intuitions here are different. These results have a fundamental impact on the choice of the communication protocol to be used in order to aggregate the information received from a group of agents, especially over problems where the search complexity can be estimated in advance.

For RL agents, Figure 4b shows an overall similar shift in effectiveness for both Q-learning and Sarsa collectives. Their performances significantly increase over isolated agents to reach a peak around $\mathcal{H}(\mu) = 16$ bits, but then start to drop down over higher uncertainties. This illustrates the fact that cooperative RL collectives are most advantageous over environments which are somewhat highly-uncertain for isolated RL agents to be efficient, yet not too uncertain for them (the cooperative collectives) to be considerably efficient. In other words, collective intelligence might only be slightly perceived in groups operating in very simple environments (or problems) where individuals could perform relatively well, or in those too difficult (broad) to be explored within a limited *interaction time*, given a limited *number of members* in the group. Thus, in order to understand the global picture of collective behaviour and its dynamics, it is crucial to look into the latter two factors (interaction time and number of members) and measure their effects, if any exist, on group intelligence.

⁵ Except for random agents which always score around zero.

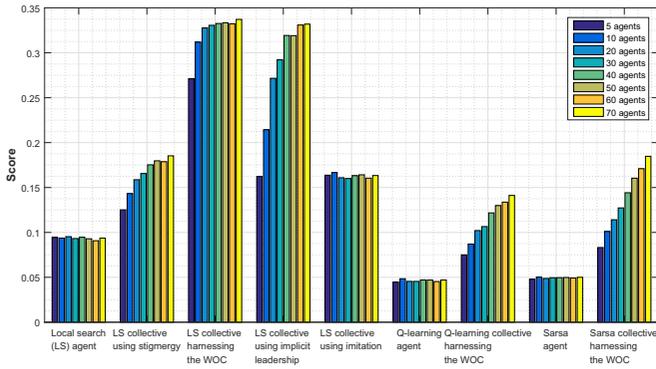


Figure 5: Intelligence scores recorded across different numbers of agents $5 \leq |\Pi| \leq 70$, in 17.8-bit $\mathcal{H}(\mu)$ environments.

7.4 Number of agents in a group

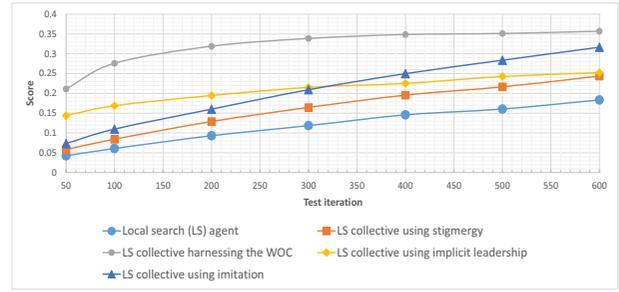
In all our previous experiments the number of evaluated agents in each collective was set to 10. Whereas, Figure 5 illustrates the scores of the evaluated collectives across different number of agents varying between 5 and 70. The general picture shows that local search collectives relying on stigmergy and auctions gradually improve in performance as more agents are added into the collective. This is not the case for collectives relying on imitation, which only show a shallow variation in score. In fact, local search agents relying on imitation performed better than those using auctions when $|\Pi|$ was set to 5 agents. However, the opposite was true when we increased the number of agents to 10 and higher. This illustrates that, when the group is small in number, relying on a super-solver agent might be more advantageous than interacting between the individual members, however, as the group gets larger, more information is added into the collective and the expertise of a single oracle becomes rudimentary in comparison to the aggregated experiences (synergy) from individual members. Moreover, we observe that local search collectives harnessing the WOC improve faster in performance than those following a leadership. Nonetheless, when the number of agents gets higher the performances of these two collectives get closer to one another.

We also observe that, increasing the number of local search agents is more effective and has greater influence on the scores for agents relying on auctions than those using stigmergy. On another hand, the increase in efficiency is slightly non-linear to the number of agents introduced. For instance, the main improvements in scores are more concentrated at the early introductions of agents. Afterwards the scores continue to rise, but less and less significantly.

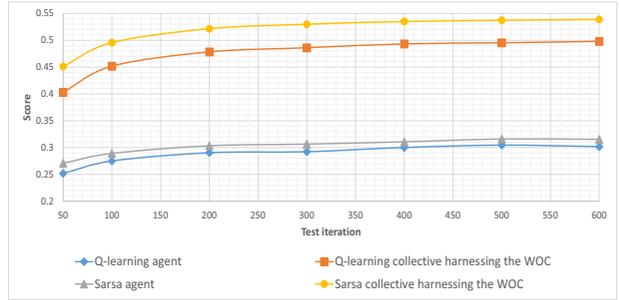
Similar observations illustrate that the effectiveness of RL collectives harnessing the WOC improves as we increment the number of agents. Moreover, Sarsa collectives seem to be slightly more efficient than Q-learning collectives as new agents are introduced into the group. The key issue in this experiment is that, collective intelligence cannot be considered independently of the number of members in the group. Instead, it is a function of - so far - at least three factors, each having a different influence that we have measured, and bearing distinctive properties of which we have identified some.

7.5 Time and intelligence

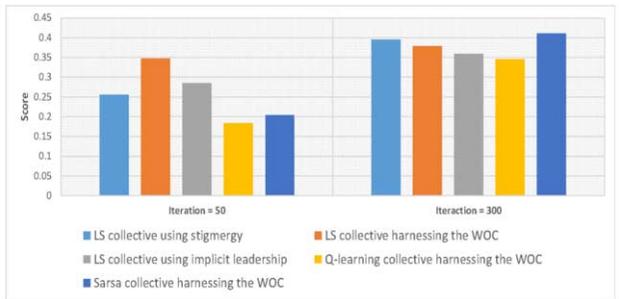
In this paragraph we address the relevance of time to intelligence, which is often ignored in the assessment of collective intelligence. Figure 6 shows the variations in the intelligence scores as we extend



(a) Local search agent collectives in $\mathcal{H}(\mu) = 17.2$ bits.



(b) RL collectives in $\mathcal{H}(\mu) = 13.2$ bits.



(c) Scores in $\mathcal{H}(\mu) = 13.2$ bits after 50 and 300 iterations.

Figure 6: Variations in the agents' intelligence scores as we extend the evaluation time (number of iterations) of the test.

the interaction time (number of iterations or interaction steps) of the test. We observe that some scores incline to converge as more time is given to the members to perform on the test. Figure 6a shows that the advantage of cooperative local search agent groups over isolated agents is higher at the early stages of the test in the case of agents using auctions to claim leadership. Afterwards, the gap in performance slowly decreases with time until iteration number 600. On the contrary, the gap between the scores of local search agents imitating an oracle and their isolated peers grows as we let the test run, implying that local search agents relying on imitation require longer periods of time to reach their best performance. We have already shown in Figure 3 that over some uncertainties, local search agents relying on auctions outperform those imitating an oracle, which is again consistent with the results in Figure 6a (up to iteration 300). However, this experiment also suggests that imitating a super-solver is highly rewarding over time, leading to better-scoring collectives than when using leadership through auctions. These results illustrate how diverse social organisations between the members of the collective determine its performance over time. For instance, a (dynamic) leadership scheme or organisation seems to be more rewarding than a simple flat hierarchy relying on stigmergic communication given a

limited interaction time with the environment.

Moreover, the general picture shows that a local search collective harnessing the WOC is most advantageous over isolated agents (and other collectives) mainly before the 300th iterations, at which point its performance begins to converge slowly.

In the case of RL agents (Figure 6b), both isolated agents and collectives improve in performance with time, keeping an overall steady relationship between the differences in their scores. This raises another concern re the intelligence of artificial agents. It is intriguing as to what ideally counts as more intelligent, a fast re-active agent with a humble performance, or a slow one with an exceptional performance over an extended period of time? Should we consider the *potential intelligence* of an agent instead? To understand the importance of time in measuring intelligence, we compare the scores of RL and local search collectives over 13.2-bit $\mathcal{H}(\mu)$ environments after 50 and 300 test iterations as illustrated in Figure 6c. We find that local search collectives outscored Sarsa collectives up to the first 50 iterations while the opposite is true at iteration 300. This type of experiment is one of the most revealing of how the (communication and interaction) reasoning/learning speed of multiagent systems influences their measured performance given a finite/bounded operation or interaction time.

7.6 Algorithmic complexity and intelligence

In this paragraph we shed some light on how the performance of (groups of) agents is influenced by the algorithmic complexity of the task. To minimise the effect of search and exploration (relative to exploitation) on the scores we initialised all agents to neighbour locations from \oplus . We then evaluated the agents over tasks of different algorithmic complexities (randomness) $K(\mu)$ grouped into three difficulty levels: easy $\in [6, 8]$, medium $\in [9, 13]$ and hard $\in [14, 19]$. This experiment stands out from previous related experiments in the field, as collectives are assessed against tasks of quantifiable algorithmic complexity, as opposed to ones qualitatively ranked based of their difficulty. Results illustrated in Figure 7 show that the perfor-

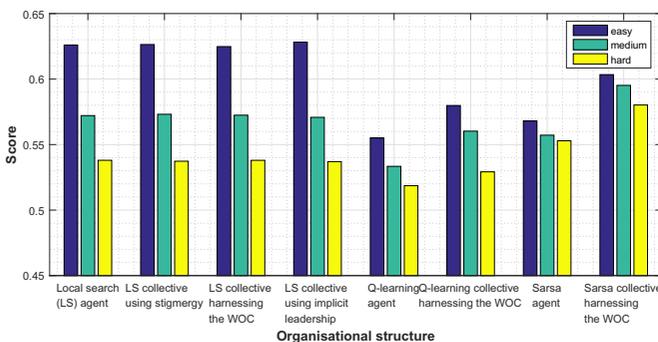


Figure 7: Scores over different task complexities $K(\mu)$ using collectives of $|\Pi| = 10$ agents, evaluated in 13.2-bit $\mathcal{H}(\mu)$ environments for 50 interactions.

mance of artificial agents, similar to that of individual human performance [22], decreases when evaluated over patterns of higher algorithmic complexities. For instance, learning and predicting random patterns is more difficult, per se, than learning or inferring compressible ones. Moreover, this experiment suggests that RL collectives are better learners than their isolated peers since the difference between the cooperative agents' scores over the 3 levels of difficulties is significantly smaller than that of the isolated ones. What's more

intriguing in Figure 7 is the difference in behaviour between cooperative RL agents and local search collectives. While RL collectives are still more effective over isolated agents when evaluated over learning problems, all local search agents (isolated and collectives) performed equally when the effect of *search and exploration* was minimised. More importantly, we find that RL collectives are more robust with respect to the change in algorithmic complexity as opposed to local search agents which display a wide gap in scores over the three levels of complexity.

All in all, what this experiment suggests is that, further to the previously examined factors, (collective) intelligence is a function of the agent type and the algorithmic complexity of the given task, both combined.

8 ORGANISATIONAL BEHAVIOUR

In spite of the different communication protocols we have evaluated, it is still not clear how the organisational structure of the group [3, 30] affects its performance on intelligence tests. Therefore, we have further evaluated the performance of equally sized collectives of local search agents organised in four different (divisional and network) structures and studied their organisational behaviour. These structures are illustrated in Figure 8 below. In the flat, fully connected,

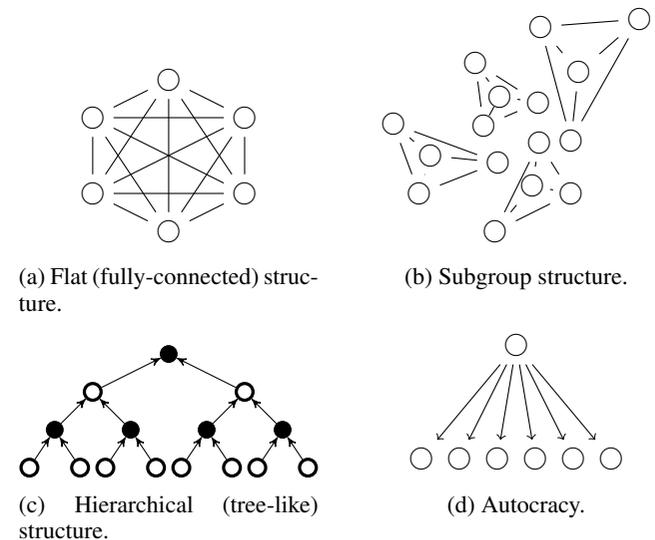


Figure 8: Graphical representation of different group organisational structures. Nodes represent agents and edges reflect the flow of communication and interaction between these agents.

structure (Figure 8a) all agents share their observations between one another. This absolute aggregation of information leads to a similar effect as that of local search collectives harnessing the wisdom of the crowd. In the subgroup structure (Figure 8b) we divide the collective into four smaller subgroups. Each one of those subgroups then implements a flat structure as the one described previously. In the hierarchical structure (Figure 8c), each (non-leaf) agent receives feedback from its children at each iteration of the test before selecting an action. Leaf-nodes operate in isolation. Finally, in the autocratic structure (Figure 8d), a single agent controls the actions of the rest of the collectives irrespective of its members' observations.

The results from our experiments show that flat, fully-connected, network structures are the most efficient since they maximise the aggregation of information received from the members of the collec-

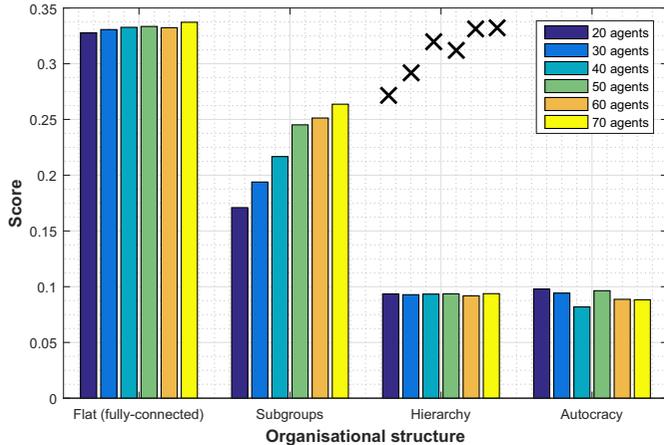


Figure 9: Scores of local search collectives organised in different network structures, across various number of agents. The collectives are evaluated in 17.2-bit $\mathcal{H}(\mu)$ environments for 50 iterations. The label (\times) in this figure depicts the average score of the agent at the top of the hierarchy (root node of the binary-tree).

tive. However, it is known that this type of structure is very costly as it requires a large number of connections ($n(n-1)/2$ connections, where n is the number of agents) to be introduced between the members of the group [36]. In this type of organisation (and for the corresponding environment uncertainty and evaluation-time parameters) the number of agents did not significantly affect the performance of the collective. Whereas, after dividing this collective into smaller subgroups, the number of agents turns out to be of major importance⁶. The effectiveness of each subgroup improved gradually with the increase in the number of agents thus reducing the gap in performance between this organisational structure and fully-connected one. This shows that dividing a collective into smaller groups is most beneficial for highly populated collectives, especially when the number of connections inside the collective grows very large and becomes a bottleneck on communication.

In the hierarchical and autocratical structures, the measured effectiveness is low compared to the previous two models. We observe that the average performance of a hierarchical group is slightly steadier than that of a group governed by single agent with absolute control on decision-making. Interestingly, we have noticed that in the hierarchical structure, high-scoring agents are the ones at the top of the hierarchy since (in our model) they receive feedback from their children (which in turn receive feedback from theirs) while the ones at the bottom perform in isolation and have low scores. Figure 9 shows that the average scores of the root agents in the hierarchy are significantly higher than the average score of the collective, indicating a high standard deviation between the members' scores in this organisational structure. Since the number of leaves is almost half the number of nodes⁷, and the number of agents declines quickly as we move up the hierarchy, this organisation does not deliver a high average group performance.

Finally, our results show that the performance of a local search collective implementing an autocracy is similar on average to that of isolated local search agents. Agents in this organisational structure do not show any significant discrepancies in their scores or behaviours.

⁶ Note that all four evaluated subgroups showed a similar performance, but scores were only plotted for the first subgroup to enhance readability.

⁷ Number of leaves in a full binary tree is equal to $(\#nodes + 1)/2$.

9 CONCLUSIONS AND FUTURE WORK

We have addressed the relevance of several factors and their interaction to the notion of intelligence and its emergence. We first started by looking at the different contexts in which collective intelligence has been shown to emerge, from face-to-face human groups, group collaborations via the web, social insect colonies and swarms, etc. Accordingly, we filtered a series of factors and features that are not coupled to one particular cognitive system, problem or environment, and illustrated how they influence the collective behaviour of the group, and hinder its intelligence.

The studied factors were shown to have a major influence on the performance of collectives that we have also measured. But, what made our conclusions more intriguing is the peculiar nature of collective intelligence seen as a function of all the examined factors simultaneously, as well as some of them combined. We identified circumstances where one cooperative system outperformed another under some values or setups of the studied factors yet failed to do so under others (e.g., in Section 7.5, limited vs. extended interaction time and, in Section 7.3, low vs. high environment uncertainty), reflecting on how these factors independently but also jointly shape the effectiveness of multiagent systems, and the spread of intelligence in these systems. Some of our conclusions (in Section 7.3) reflected how relying on an expert (super-solver) agent in the group does not guarantee its optimal performance. We also measured the effect of introducing more agents into the group (Section 7.4), and showed that it is tightly controlled by the communication protocol used between its members. We have highlighted scenarios (in Section 7.6) where only some types of collectives outperform their equally sized group of isolated agents over (algorithmically) complex environments, and shown how the influence of the environment difficulty (uncertainty and complexity) is a major factor controlling the capacity for intelligence. Moreover, we looked (in Section 8) into how the effectiveness of (the same selection of) agents adopting different organisational and network structures can significantly vary from one structure to another.

We have answered a fundamental question by showing the existence, and *quantitatively* measuring the influence, of some general factors and principles shaping the spread of intelligence that are regularly perceived across different cognitive systems.

In order for our results to be transferred to a guideline for designing multiagent cooperation, we have released the source code and scripts to run our experiments as open-source in [7, Section 5.1]. This will allow both additional testing and extensions to (the current version of) the Λ^* environment. The motivation is to encourage people in the AI community to quantitatively evaluate new types of heuristics, algorithms, communication protocols and network structures.

Another future goal is to further evaluate agents and collectives over a wide range of general AI problems. For example, agents (isolated or collectives) could be evaluated over exploration/exploitation problems in an environment consisting of a hidden fitness landscape with many local, and only one global, optima as further elaborated in [7, Section 6.2]. Other possible examples might include pattern recognition (and sequence completion) problems, in which payoff is determined by how accurately a subject learns and predicts a pattern. Other general multiagent problems that require coordination [7, Section 6.1] (e.g., lifting and moving an object), or scheduling [4] (e.g., job shop scheduling), can be used for alternative evaluation.

Finally, we hope that - by following this direction in the quantification of intelligence - we would pave the way towards a rigorous and unified model of collectively intelligent groups and societies.

REFERENCES

- [1] Luís M. A. Bettencourt, 'The rules of information aggregation and emergence of collective intelligent behavior', *Topics in Cognitive Science*, **1**(4), 598–620, (2009).
- [2] Eric Bonabeau, 'Decisions 2.0: The Power of Collective Intelligence', *MIT Sloan Management Review*, (Winter), 45–52, (2009).
- [3] John Child, 'Organizational structure, environment and performance: The role of strategic choice', *sociology*, **6**(1), 1–22, (1972).
- [4] Nader Chmait and Khalil Challita, 'Using simulated annealing and ant-colony optimization algorithms to solve the scheduling problem', *Computer Science and Information Technology*, **1**(3), 208–224, (2013).
- [5] Nader Chmait, David L. Dowe, David G. Green, and Yuan-Fang Li, 'Observation, communication and intelligence in agent-based systems', in *Proc. 8th Int. Conf. Artificial General Intelligence, Berlin, Germany*, volume 9205 of *Lecture Notes in Artificial Intelligence (LNAI)*, pp. 50–59. Springer, (Jul 2015).
- [6] Nader Chmait, David L. Dowe, David G. Green, Yuan-Fang Li, and Javier Insa-Cabrera, 'Measuring universal intelligence in agent-based systems using the anytime intelligence test', Technical Report 2015/279, FIT, Clayton, Monash University, (2015).
- [7] Nader Chmait, Yuan-Fang Li, David L. Dowe, and David G. Green, 'A dynamic intelligence test framework for evaluating AI agents', in *Proc. Evaluating General-Purpose AI (EGPAI), ECAI workshop*, (2016).
- [8] Iain D. Couzin, 'Collective cognition in animal groups', *Trends in Cognitive Sciences*, **13**(1), 36–43, (2009).
- [9] David L. Dowe and Alan R. Hajek, 'A computational extension to the Turing Test', *Proc. 4th Conf. of the Australasian Cognitive Science Society, University of Newcastle, NSW, Australia*, (1997).
- [10] David L. Dowe and Alan R. Hajek, 'A computational extension to the Turing Test', *Technical Report #97/322, Dept Computer Science, Monash University, Melbourne, Australia*, (1997).
- [11] David L. Dowe and Alan R. Hajek, 'A non-behavioural, computational extension to the Turing Test', in *International conference on computational intelligence & multimedia applications (ICCIAMA'98), Gippsland, Australia*, pp. 101–106, (1998).
- [12] David L. Dowe, José Hernández-Orallo, and Paramjit K. Das, 'Compression and intelligence: Social environments and communication', in *Proc. 4th Int. Conf. on AGI*, pp. 204–211, Berlin, (2011). Springer.
- [13] David Engel, Anita W. Woolley, Ishani Aggarwal, Christopher F. Chabris, Masamichi Takahashi, Keiichi Nemoto, Carolin Kaiser, Young Ji Kim, and Thomas W. Malone, 'Collective intelligence in computer-mediated collaboration emerges in different contexts and cultures', in *Proc. 33rd Conf. on CHI*, pp. 3769–3778, NY, USA, (2015). ACM.
- [14] Scott C. Evans, John E. Hershey, and Gary Saulnier, 'Kolmogorov complexity estimation and analysis', in *6th World Conf. on Systemics, Cybernetics and Informatics*, (2002).
- [15] Pierre-P. Grassé, 'La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs', *Insectes sociaux*, **6**(1), 41–80, (1959).
- [16] Lawrence Gray, 'A mathematician looks at Wolfram's new kind of science', *Notices of the American Mathematical Society*, **50**(2), 200–211, (2003).
- [17] Vahid Hashemi and Ulle Endriss, 'Measuring diversity of preferences in a group', in *ECAI*, pp. 423–428, (2014).
- [18] José Hernández-Orallo and David L. Dowe, 'Measuring universal intelligence: Towards an anytime intelligence test', *Artificial Intelligence*, **174**(18), 1508–1539, (December 2010).
- [19] José Hernández-Orallo and Neus Minaya-Collado, 'A formal definition of intelligence based on an intensional variant of Kolmogorov complexity', in *Proc. of the Int. Symposium of EIS*, pp. 146–163. ICSC Press, (1998).
- [20] Lu Hong and Scott E. Page, 'Groups of diverse problem solvers can outperform groups of high-ability problem solvers', *PNAS*, **101**(46), 16385–16389, (2004).
- [21] Javier Insa-Cabrera, José-Luis Benacloch-Ayuso, and José Hernández-Orallo, 'On measuring social intelligence: Experiments on competition and cooperation', in *Proc. 5th Conf. on AGI*, eds., Joscha Bach, Ben Goertzel, and Matthew Iklé, volume 7716 of *LNCS*, pp. 126–135. Springer, (2012).
- [22] Javier Insa-Cabrera, David L. Dowe, Sergio España-Cubillo, M. Victoria Hernandez-Lloreda, and José Hernández-Orallo, 'Comparing humans and AI agents.', in *AGI*, volume 6830 of *LNCS*, pp. 122–132. Springer, (2011).
- [23] Javier Insa-Cabrera, José Hernández-Orallo, David L. Dowe, Sergio España, and M. Victoria Hernández-Lloreda, 'The ANYNT project intelligence test: λ_{one} ', in *AISB/IACAP 2012 Symposium "Revisiting Turing and his Test"*, pp. 20–27, (2012).
- [24] Mark Klein and Ana Cristina B. Garcia, 'Crowd computing: From human computation to collective intelligence (tutorial)', in *Proc. 24th IJCAI, Buenos Aires, Argentina, July*, (2015).
- [25] Shane Legg and Marcus Hutter, 'Universal intelligence: A definition of machine intelligence', *Minds and Machines*, **17**(4), 391–444, (2007).
- [26] Abraham Lempel and Jacob Ziv, 'On the Complexity of Finite Sequences', *Information Theory, IEEE Transactions on*, **22**(1), 75–81, (January 1976).
- [27] Ming Li and Paul Vitányi, *An introduction to Kolmogorov complexity and its applications (3rd ed.)*, Springer-Verlag New York, Inc., 2008.
- [28] Thomas W. Malone and Michael S. Bernstein, *Handbook of collective intelligence*, MIT Press, 2015.
- [29] Winter Mason and Duncan J. Watts, 'Collaborative learning in networks', *PNAS*, **109**(3), 764–769, (2012).
- [30] Henry Mintzberg, 'The structuring of organizations: A synthesis of the research', *University of Illinois at Urbana-Champaign's Academy for Entrepreneurial Leadership Historical Research Reference in Entrepreneurship*, (1979).
- [31] Simon Parsons, Juan A. Rodríguez-Aguilar, and Mark Klein, 'Auctions and bidding: A guide for computer scientists', *ACM*, **43**(2), 1–59, (Feb 2011).
- [32] Massimo Poesio, Jon Chamberlain, Udo Kruschwitz, Livio Robaldo, and Luca Ducceschi, 'Phrase detectives: Utilizing collective intelligence for internet-scale language resource creation (extended abstract)', in *Proc. 24th IJCAI, Buenos Aires, Argentina*, pp. 4202–4206, (2015).
- [33] G. A. Rummery and M. Niranjan, 'On-line Q-learning using connectionist systems', Technical Report TR 166, Cambridge University Engineering Department, Cambridge, England, (1994).
- [34] Pritika Sanghi and David L. Dowe, 'A computer program capable of passing I.Q. tests', in *Proc. of the Joint International Conference on Cognitive Science, 4th ICCS International Conference on Cognitive Science & 7th ASCS Australasian Society for Cognitive Science (ICCS/ASCS-2003)*, ed., P. P. Slezak, pp. 570–575, Sydney, Australia, (13-17 July 2003).
- [35] Claude E. Shannon, 'A mathematical theory of communication', *Bell System Technical Journal*, **27**(3), 379–423, (July 1948).
- [36] Steven H Strogatz, 'Exploring complex networks', *Nature*, **410**(6825), 268–276, (2001).
- [37] James Surowiecki, *The Wisdom of Crowds*, Anchor, 2005.
- [38] Kagan Tumer and David Wolpert, 'A survey of collectives', in *In collectives and the design of complex systems*, pp. 1–42. Springer, (2004).
- [39] Christopher J. C. H. Watkins and Peter Dayan, 'Technical note: Q-learning', *Mach. Learn.*, **8**(3-4), 279–292, (May 1992).
- [40] Eric W. Weisstein. Moore neighborhood, from mathworld - a wolfram web resource. <http://mathworld.wolfram.com/MooreNeighborhood.html>, 2015. Last accessed: 2015-11-10.
- [41] Michael Wooldridge, *An Introduction to MultiAgent Systems*, Wiley Publishing, 2nd edn., 2009.
- [42] Anita W. Woolley, Ishani Aggarwal, and Thomas W. Malone, 'Collective intelligence and group performance', *Current Directions in Psychological Science*, **24**(6), 420–424, (2015).
- [43] Anita W. Woolley, Christopher F. Chabris, Alex Pentland, Nada Hashmi, and Thomas W. Malone, 'Evidence for a collective intelligence factor in the performance of human groups', *Science*, **330**(6004), 686–688, (2010).

Synthesizing Argumentation Frameworks from Examples

Andreas Niskanen and Johannes P. Wallner and Matti Järvisalo¹

Abstract. Argumentation is nowadays a core topic in AI research. Understanding computational and representational aspects of abstract argumentation frameworks (AFs) is a central topic in the study of argumentation. The study of realizability of AFs aims at understanding the expressive power of AFs under different semantics. We propose and study the *AF synthesis problem* as a natural extension of realizability, addressing some of the shortcomings arising from the relatively stringent definition of realizability. Specifically, AF synthesis seeks to construct, or synthesize, AFs that are semantically closest to the knowledge at hand even when no AFs exactly representing the knowledge exist. Going beyond defining the AF synthesis problem, we (i) prove NP-completeness of AF synthesis under several semantics, (ii) study basic properties of the problem in relation to realizability, (iii) develop algorithmic solutions to AF synthesis using constrained optimization, (iv) empirically evaluate our algorithms on different forms of AF synthesis instances, as well as (v) discuss variants and generalization of AF synthesis.

1 INTRODUCTION

The study of representational and computational aspects of argumentation is a core topic in modern artificial intelligence (AI) research [5]. A current strong focus of argumentation research is the extension-based setting of abstract argumentation frameworks (AFs) [14] and its generalizations. A fundamental knowledge representational aspect related to AFs is *realizability* [15], i.e., the question of whether a specific AF semantics allows for exactly representing a given set of extensions as an AF. With important motivations from various perspectives—including the analysis of the relationships of central AF semantics [15] (in terms of the range of sets of extensions different semantics allow for representing as AFs) and connections to the study of argumentation dynamics [13, 11] (in terms of the ability to construct an AF for revised extensions)—realizability has recently been studied by several authors [15, 3, 16, 23, 19, 20].

While the study of realizability has provided various fundamental insights into AFs, the concept of realizability is quite strict in that a set E of extensions is considered realizable (under a specific AF semantics σ) if and only if there is an AF the σ -extensions of which are *exactly* those in E . Implicitly, this definition hence requires that *all* other sets of arguments *must not* be extensions of the AF of interest. This strictness requires that we have *complete* knowledge of the extensions of interest, and further, in order to actually construct a corresponding AF of interest, relies on the assumption that the set of extensions are *not conflicting* in terms of allowing them to be exactly represented by an AF. However, from more practical perspectives, we foresee these requirements to be somewhat cumbersome. Firstly,

the requirement of complete knowledge implies in the worst case taking into account an exponential number of extensions. Secondly, the definition does not allow for “mistakes” or noise in the process of obtaining the extensions, and also rules out the possibility of dealing with multiple sources of potentially conflicting sets of extensions.

In this work, with a central goal of generalizing the concept of realizability to accommodate incomplete and noise information on extensions, we propose and study what we call the *AF synthesis problem*². Specifically, AF synthesis relaxes the notion of realizability to incomplete information—assuming only partial knowledge of extensions and non-extensions as *positive* and *negative examples*—and noisy settings, by allowing for expressing relative trust in the examples via weights. In this generalized setting, we define AF synthesis as the constrained optimization task of finding an AF that optimally represents the given examples in terms of minimizing the costs (defined via the weights of the given examples) incurred from the AF by including a negative example or not including a positive example. Beyond precisely defining the AF synthesis problem, our main contributions include the following.

- We formally analyze the relationship of AF synthesis and realizability in terms of necessary and sufficient conditions for an AF synthesis instance to be realizable under different AF semantics (Section 3).
- We provide complexity results for AF synthesis under three central AF semantics, namely, the conflict-free, admissible, and stable semantics, with the main result that AF synthesis is in the general case NP-complete under each of these semantics (Section 4).
- We develop a first constraint-based approach to optimal AF synthesis, by providing declarative encodings for AF synthesis in the Boolean optimization paradigm of maximum satisfiability (MaxSAT), and furthermore, discuss how by simple modifications to the encoding one can account for structural constraints over the AFs to be synthesized (Section 5).
- We present results from an empirical evaluation of the approach based on benchmarks from the recent ICCMA’15 argumentation solver competition [26] as well as additional randomly generated AF synthesis instances (Section 6).
- We discuss further variants and generalizations of AF synthesis from representational and computational complexity perspectives, including how to adapt the problem to multiple sources of extensions and allowing for mixtures of different AF semantics, as well as symbolic representations of examples via Boolean formulas (Section 7).

A more detailed overview of connections to related work is provided after the main contributions (Section 8). For readability, more complicated formal proofs are detailed in Appendix A.

¹ Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Finland

² Alternatively, one could refer to the problem focused on in this paper as an *AF learning* problem.

2 ARGUMENTATION FRAMEWORKS

We start by briefly recalling argumentation frameworks [14] (see also [2]) as the central formalism in abstract argumentation.

Definition 1. An argumentation framework (AF) is a pair $F = (A, R)$, where A is a finite non-empty set of arguments and $R \subseteq A \times A$ is the attack relation. The pair $(a, b) \in R$ indicates that a attacks b . An argument $a \in A$ is defended (in F) by a set $S \subseteq A$ if, for each $b \in A$ such that $(b, a) \in R$, there is a $c \in S$ such that $(c, b) \in R$.

Semantics for AFs are defined through functions σ which assign to each AF $F = (A, R)$ a set $\sigma(F) \subseteq 2^A$ of extensions. We consider for σ the functions *stb*, *adm*, *com*, and *grd*, which stand for stable, admissible, complete, and grounded, respectively.

Definition 2. Given an AF $F = (A, R)$, the characteristic function $\mathcal{F}_F : 2^A \rightarrow 2^A$ of F is $\mathcal{F}_F(S) = \{x \in A \mid x \text{ is defended by } S\}$. Moreover, for a set $S \subseteq A$, the range of S is $S_R^+ = S \cup \{x \mid (y, x) \in R, y \in S\}$.

Definition 3. Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is conflict-free (in F) if there are no $a, b \in S$ such that $(a, b) \in R$. We denote the collection of conflict-free sets of F by $cf(F)$. For a conflict-free set $S \in cf(F)$ it holds that

- $S \in stb(F)$ iff $S_R^+ = A$;
- $S \in adm(F)$ iff $S \subseteq \mathcal{F}_F(S)$;
- $S \in com(F)$ iff $S = \mathcal{F}_F(S)$;
- $S \in grd(F)$ iff S is the least fixed-point of \mathcal{F}_F .

For any AF F , we have $cf(F) \supseteq adm(F) \supseteq com(F) \supseteq stb(F)$. We use “ σ -extension” to denote an extension under a semantics σ .

3 THE AF SYNTHESIS PROBLEM

In this section we introduce the AF synthesis problem. For a given set of weighted examples that represent semantical information, with weights intuitively representing relative trust in the examples, the task is to synthesize an AF that has minimum cost over the examples not satisfied. We assume a given non-empty set of arguments A from which we are to construct an AF. Formally, an example $e = (S, w)$ is a pair with S a subset of the set of arguments, i.e., $S \subseteq A$, and a positive integer $w > 0$ representing the example’s weight. We denote the set of arguments of an example $e = (S, w)$ by $S_e = S$ and weight by $w_e = w$. For a set E of examples, we define $\mathbb{S}_E = \{S_e \mid e \in E\}$ as a shorthand for the set of all sets of arguments occurring in E .

An instance of the AF synthesis problem is a quadruple $P = (A, E^+, E^-, \sigma)$, with a non-empty set A of arguments, two sets of examples, E^+ and E^- , that we call positive and negative examples, respectively, and semantics σ . An AF F satisfies a positive example e if $S_e \in \sigma(F)$; similarly, F satisfies a negative example if $S_e \notin \sigma(F)$. For a given AF F , the associated cost w.r.t. P , denoted by $cost(P, F)$, is the sum of weights of examples not satisfied by F . Formally, $cost(P, F)$ is

$$\sum_{e \in E^+} w_e \cdot I(S_e \notin \sigma(F)) + \sum_{e \in E^-} w_e \cdot I(S_e \in \sigma(F)),$$

where $I(\cdot)$ is the indicator function that returns 1 if the property (membership in a set) is satisfied, and otherwise 0. The task in AF synthesis is to find an AF of minimum cost over all AFs.

AF Synthesis

INPUT: $P = (A, E^+, E^-, \sigma)$

TASK: Find an AF F^* with

$$F^* \in \arg \min_{F=(A,R)} (cost(P, F)).$$

Example 1. Consider the set of positive examples $E^+ = \{\{a, b\}, 1\}, \{\{a, c\}, 1\}, \{\{b, c\}, 5\}\}$ and the set of negative examples $E^- = \{\{a\}, 1\}, \{\{a, b, c\}, 5\}$. We illustrate these examples in Figure 1. Here we see that the positive examples claim together that each pair of arguments of A is a σ -extension, and the negative examples claim that the whole set A is not a σ -extension and that the singleton set $\{a\}$ is likewise not a σ -extension. Let $P_{cf} = (A, E^+, E^-, cf)$ with $A = \{a, b, c\}$ be an AF synthesis instance under conflict-free semantics. An optimal solution AF $F_{cf} = (A, R_{cf})$ with $cost(P_{cf}, F_{cf}) = 2$ is given by $R_{cf} = \{(a, b)\}$. This AF F_{cf} does not satisfy the positive example $(\{a, b\}, 1)$ and the negative example $(\{a\}, 1)$.

Regarding admissible semantics, let $P_{adm} = (A, E^+, E^-, adm)$. In this case AF $F_{adm} = (A, R_{adm})$ is an optimal solution with $R_{adm} = \{(b, a)\}$. Except for positive examples $(\{a, b\}, 1)$ and $(\{a, c\}, 1)$, all other examples are satisfied by F_{adm} for P_{adm} . Thus $cost(P_{adm}, F_{adm}) = 2$.

For stable semantics, let $P_{stb} = (A, E^+, E^-, stb)$. An optimal solution AF to P_{stb} is given by $F_{stb} = (A, R_{stb})$ with $R_{stb} = \{(a, b), (b, a)\}$. Here $stb(F_{stb}) = \{\{a, c\}, \{b, c\}\}$ and $cost(P_{stb}, F_{stb}) = 1$.

We now investigate the existence of 0-cost solutions for the AF synthesis problem by relating the problem with realizability results from [15]. In contrast to the AF synthesis problem, [15] consider the problem of a given unweighted set \mathbb{S} of sets of arguments, and ask whether there is an AF F s.t. $\mathbb{S} = \sigma(F)$. In words, in the setting of realizability, the given set exactly specifies which sets have to be σ -extensions and which must not be σ -extensions. Further, [15] do not consider weights attached to examples, and the set of arguments A is not specified and may contain more arguments than occurring in \mathbb{S} . Restricting the set of arguments to only arguments occurring in \mathbb{S} is studied in [3, 20], although not directly applicable to our problem.

We make use of and generalize the notions proposed in [15] by specifying conditions under which 0-cost solutions exist and as well as properties 0-cost solutions satisfy. We first focus on the conflict-free semantics. We utilize the following concept adapted from [15, Definitions 6 and 7], defining a consequence operator that states which sets must be conflict-free if we assume a given set of sets \mathbb{S} to be conflict-free in an AF. Let $ImpliedCF(\mathbb{S}) = \{X \mid a, b \in X \text{ implies } \exists S \in \mathbb{S} \text{ with } \{a, b\} \subseteq S\}$. Intuitively, if each set in \mathbb{S} is conflict-free, and each pair of arguments in a set X is contained in one set of \mathbb{S} , then X is conflict-free as well. Note that a and b in this definition need not be distinct ($\{a, b\}$ is equal to $\{a\}$ if $a = b$). Further, \emptyset is in $ImpliedCF(\mathbb{S})$ for any \mathbb{S} .

Lemma 1. Let $F = (A, R)$ be an AF and $\mathbb{S} \subseteq 2^A$. If $\mathbb{S} \subseteq cf(F)$, then $ImpliedCF(\mathbb{S}) \subseteq cf(F)$.

Example 2. Continuing from Example 1, consider $\mathbb{S}_{E^+} = \{\{a, b\}, \{a, c\}, \{b, c\}\}$. If each element of \mathbb{S}_{E^+} is conflict-free in an AF F ($\mathbb{S}_{E^+} \subseteq cf(F)$), then, e.g., $\{a, b, c\} \in cf(F)$, since there cannot be an attack between any of these three arguments. In particular, we have $ImpliedCF(\mathbb{S}_{E^+}) = \mathbb{S}_{E^+} \cup \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b, c\}\}$. This directly shows for P_{cf} from Example 1 that there is no solution

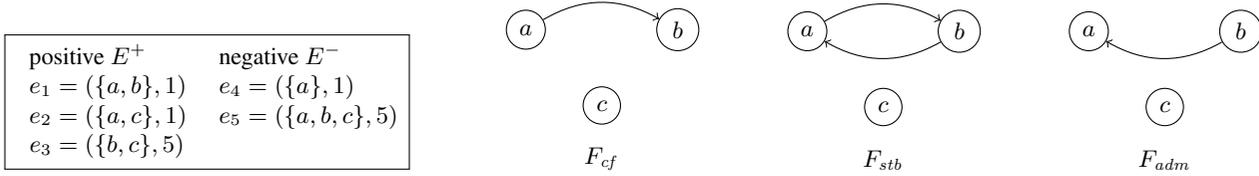


Figure 1. AF synthesis example with optimal solution AFs

AF to P_{cf} of cost 0. In fact, there is no AF satisfying both the positive example $(\{a, b\}, 1)$ and the negative example $(\{a\}, 1)$ under the conflict-free semantics. Also, there is no AF satisfying all three positive examples and negative example $(\{a, b, c\}, 5)$ under the conflict-free semantics.

Equipped with the preceding lemma, we give a necessary and sufficient condition for 0-cost solutions for AF synthesis under the conflict-free semantics.

Proposition 2. Let $P = (A, E^+, E^-, cf)$ be an instance of AF synthesis. There is a solution AF F to P with $\text{cost}(P, F) = 0$ iff $\text{ImpliedCF}(\mathbb{S}_{E^+}) \cap \mathbb{S}_{E^-} = \emptyset$.

Intuitively, to synthesize an AF F that has \mathbb{S}_{E^+} as its conflict-free sets, $\text{ImpliedCF}(\mathbb{S}_{E^+})$ need to be conflict-free, too. Moreover, using results from [15], one can show that there is an AF F with $cf(F) = \text{ImpliedCF}(\mathbb{S}_{E^+})$. Furthermore, if no negative example e claims that a set of $\text{ImpliedCF}(\mathbb{S}_{E^+})$ should not be conflict-free, i.e., $S_e \notin \text{ImpliedCF}(\mathbb{S}_{E^+})$, then this implies that F has cost 0.

Now consider admissible sets. Similarly as for conflict-free sets, we define the following consequence operator. For a set of sets \mathbb{S} , let $\text{ImpliedADM}(\mathbb{S}) = \{X \mid X = \bigcup_{S \in \mathbb{S}'} S, \mathbb{S}' \subseteq \mathbb{S}, X \in \text{ImpliedCF}(\mathbb{S})\}$. Briefly put, if we assume \mathbb{S} to be a collection of admissible sets, then each union of sets in \mathbb{S} that is conflict-free, i.e., in $\text{ImpliedCF}(\mathbb{S})$, is also an admissible set. By this definition, $\emptyset \in \text{ImpliedADM}(\mathbb{S})$ for any \mathbb{S} .

Lemma 3. Let $F = (A, R)$ be an AF and $\mathbb{S} \subseteq 2^A$. If $\mathbb{S} \subseteq \text{adm}(F)$, then $\text{ImpliedADM}(\mathbb{S}) \subseteq \text{adm}(F)$.

Example 3. Consider $\mathbb{S}_{E^+} = \{\{a, b\}, \{a, c\}, \{b, c\}\}$ from Example 1. Then $\text{ImpliedADM}(\mathbb{S}_{E^+}) = \mathbb{S}_{E^+} \cup \{\{a, b, c\}, \emptyset\}$. Regarding the set $\mathbb{S}' = \{\{b, c\}\}$ which is the set of positive examples satisfied by F_{adm} , we have $\text{ImpliedADM}(\mathbb{S}') = \mathbb{S}' \cup \{\emptyset\}$.

Consider $P'_{adm} = (\{a, b, c\}, E_1^+, E_1^-, adm)$ with $E_1^+ = \{(\{a, c\}, 1), (\{b, c\}, 1)\}$ and $E_1^- = \{(\{a\}, 1)\}$. We have $\text{ImpliedADM}(\mathbb{S}_{E_1^+}) = \mathbb{S}_{E_1^+} \cup \{\emptyset\}$ and $\text{ImpliedADM}(\mathbb{S}_{E_1^+}) \cap \mathbb{S}_{E_1^-} = \emptyset$. Unlike for the conflict-free semantics, this condition for the admissible semantics does not imply existence of a 0-cost solution AF for P'_{adm} . In fact, a 0-cost solution AF does not exist for P'_{adm} . A 0-cost solution is possible, however, if the set of arguments A includes more arguments. For instance, take $F' = (\{a, b, c, d\}, \{(b, a), (a, b), (c, d), (d, a), (d, d)\})$. We have $\text{adm}(F') = \{\emptyset, \{b\}, \{c\}, \{b, c\}, \{a, c\}\}$ and, for $P''_{adm} = (\{a, b, c, d\}, E_1^+, E_1^-, adm)$, we have $\text{cost}(P''_{adm}, F') = 0$. Such “auxiliary” arguments, i.e., arguments not present in the examples, are not always required for 0-cost solutions under the admissible semantics. For instance, given only two positive examples $(\{a, c\}, 1)$ and $(\{b, c\}, 1)$, and negative example $(\{a, b, c\}, 1)$, one can synthesize a 0-cost AF with a mutual attack between a and b .

Similarly as for conflict-free semantics, each 0-cost solution under the admissible semantics implies that for no negative examples e we

have $S_e \in \text{ImpliedADM}(\mathbb{S}_{E^+})$. For existence of an AF F with $\text{ImpliedADM}(\mathbb{S}_{E^+}) = \text{adm}(F)$, we make use of results from [15] which requires auxiliary arguments, i.e., arguments not present in \mathbb{S}_{E^+} . We use the abstract function $\text{AuxArgs}(adm, \mathbb{S}_{E^+})$ that returns the number of auxiliary arguments needed to construct F as specified in [15, Definitions 13 and 14].

Proposition 4. Let $P = (A, E^+, E^-, adm)$ be an instance of the AF synthesis problem. Consider the following conditions.

1. $\text{ImpliedADM}(\mathbb{S}_{E^+}) \cap \mathbb{S}_{E^-} = \emptyset$.
2. $|A \setminus (\bigcup_{S \in \mathbb{S}_{E^+}} S)| > \text{AuxArgs}(adm, \mathbb{S}_{E^+})$.

If there is a solution AF F to P with $\text{cost}(P, F) = 0$, then condition 1 holds. If both conditions 1 and 2 hold, then there is a solution AF F to P with $\text{cost}(P, F) = 0$.

We move on to the stable semantics, under which the picture is more complex. Existence of a 0-cost solution for an AF synthesis instance implies that the set of positive examples \mathbb{S}_{E^+} is \subseteq -incomparable, does not include \emptyset , is disjoint from the negative sets \mathbb{S}_{E^-} , and no positive set $S \in \mathbb{S}_{E^+}$ is a proper subset of an implied conflict-free set in $\text{ImpliedCF}(\mathbb{S}_{E^+})$. These conditions are quite intuitive, since, e.g., a violation of the last condition violates the fact that stable extensions attack all arguments outside the set.

These conditions imply the existence of 0-cost solutions if a certain number of auxiliary arguments is available in A , i.e., arguments not present in \mathbb{S}_{E^+} . For achieving this result, we use again results from [15], providing a construction in this case that utilizes such auxiliary arguments to synthesize the AF. We provide here a rough bound for auxiliary arguments from [15, Definition 12]. More concretely, we use the function $\text{AuxArgs}(stb, \mathbb{S}_{E^+})$ that is equal to the maximum number of stable extensions for any AF with $|\mathbb{S}_{E^+}|$ many arguments (for more details, see [4, Theorem 1]).

Proposition 5. Let $P = (A, E^+, E^-, stb)$ be an instance of the AF synthesis problem. Consider the following conditions.

1. Any two distinct $S, S' \in \mathbb{S}_{E^+}$ are incomparable w.r.t. \subseteq .
2. $\forall S \in \mathbb{S}_{E^+}$ we have $S \not\subseteq S'$ for all $S' \in \text{ImpliedCF}(\mathbb{S}_{E^+})$.
3. $\emptyset \notin \mathbb{S}_{E^+}$.
4. $\mathbb{S}_{E^+} \cap \mathbb{S}_{E^-} = \emptyset$.
5. $|A \setminus (\bigcup_{S \in \mathbb{S}_{E^+}} S)| > \text{AuxArgs}(stb, \mathbb{S}_{E^+})$.

If there is a solution AF F to P with $\text{cost}(P, F) = 0$, then conditions 1-4 hold. If conditions 1-5 hold, then there is a solution AF F to P with $\text{cost}(P, F) = 0$.

Interestingly, the negative examples play a relatively minor role in 0-cost solutions under the stable semantics (see condition 4 of Proposition 5). In contrast to conflict-free or admissible sets, existence of stable extensions does not directly imply existence of further stable extensions for an unrestricted set of arguments A (this observation is also implicitly stated in [15, Lemma 2 and Proposition 1]).

Example 4. The AF F_{stb} from Example 1 has $stb(F_{stb}) = \{\{a, c\}, \{b, c\}\} = \mathbb{S}'$. Conditions 1-4 from Proposition 5 hold for \mathbb{S}' . One can synthesize an AF, e.g., F_{stb} , as a 0-cost solution to $P'_{stb} = (\{a, b, c\}, \{\{a, c\}, 1\}, \{\{b, c\}, 1\}, \emptyset, stb)$ without auxiliary arguments. An example where auxiliary arguments are required to synthesize an AF with 0-cost can be found in [3].

4 COMPLEXITY OF AF SYNTHESIS

We continue by analyzing the computational complexity of the AF synthesis problem. As the main results of this section, we show that AF synthesis is NP-complete in the unrestricted case under the conflict-free, admissible, and stable semantics. Furthermore, we show that while restricting either E^+ or E^- to be empty yields fragments of the problem where a trivial AF solves the problem optimally, NP-completeness persists even for $E^- = \emptyset$ under the stable semantics. The results are summarized in Table 1.

Table 1. Complexity of AF synthesis

	no restrictions	$E^+ = \emptyset$	$E^- = \emptyset$
Conflict-free	NP-c	trivial	trivial
Admissible	NP-c	trivial	trivial
Stable	NP-c	trivial	NP-c

We first outline special cases of AF synthesis in which a trivial solution AF is guaranteed to be optimal. In particular, if no positive examples are present, then the complete digraph $F = (A, 2^A \times 2^A)$ satisfies all negative examples (F has no stable extensions, and the only conflict-free and admissible set is \emptyset). If the set of negative examples E^- is empty, then AF synthesis under the conflict-free and admissible semantics is trivial by constructing the AF $F = (A, \emptyset)$ (every subset of A is conflict-free and admissible).

Proposition 6 (Trivial solutions). *An optimal solution AF F^* can be computed in polynomial time for an AF synthesis instance $P = (A, E^+, E^-, \sigma)$ if one of the following conditions holds.*

1. $\sigma \in \{cf, adm, stb\}$ and $E^+ = \emptyset$.
2. $\sigma \in \{cf, adm\}$ and $E^- = \emptyset$.

We now turn our attention to the NP-hard cases of the AF synthesis problem under the conflict-free, admissible, and stable semantics. Formally, the decision problem corresponding to AF synthesis consists of an AF synthesis instance $P = (A, E^+, E^-, \sigma)$ and an integer $k \geq 0$, and asks whether there is an AF $F = (A, R)$ with $cost(P, F) \leq k$.

Intuitively, the main source of NP-hardness for the AF synthesis problem for the considered semantics lies in finding an optimal subset of examples from which to synthesize an AF. We start with the conflict-free semantics and prove NP-hardness by a reduction from the Boolean satisfiability problem; recall that all formal proofs are detailed in Appendix A. For intuition on the reduction, “choosing” a truth assignment can be simulated by a set of examples similarly as in Example 1 (shown in Figure 1). In other words, for positive examples containing the sets $\{a, b\}$, $\{a, c\}$, and $\{b, c\}$, and negative example $\{a, b, c\}$, one can attach high weights (beyond the bound k) to the last two examples and unit weights to the first two. In this way any solution AF of cost at most k does not satisfy both of the unit-weighted examples, thus mimicking a truth assignment, i.e., one has to choose which of these two examples to satisfy. The reduction is completed by additional examples that simulate satisfaction

of clauses of a Boolean formula, by ensuring that attacks have to be present via negative examples.

Proposition 7. *AF synthesis is NP-complete under the conflict-free semantics.*

The reduction we use for establishing NP-hardness under the admissible semantics follows essentially the same idea.

Proposition 8. *AF synthesis is NP-complete under the admissible semantics.*

For the stable semantics, we establish NP-completeness as well; however, surprisingly, in contrast to the conflict-free and admissible semantics, AF synthesis under the stable semantics is NP-complete even when E^- is empty. The reduction is technically more involved. Intuitively, presence of attacks can be simulated via arguments outside the set of a positive example, since if the set is stable, it has to attack all arguments outside the set. This is also a reason why hardness holds even if E^- is empty.

Proposition 9. *AF synthesis is NP-complete for stable semantics, even if the set of negative examples is empty.*

Finally we observe that AF synthesis under the grounded semantics is trivial, since exactly one grounded extension is present in an AF.

Proposition 10. *Let $P = (A, E^+, E^-, grd)$ be an instance of the AF synthesis problem. An optimal solution AF F^* to P can be constructed in polynomial time.*

Proof. For each $e \in E^+$ an AF F with $grd(F) = \{S_e\}$ and with $cost(P, F)$ equal to the sum of all weights of $E^+ \setminus \{e\}$ plus $w_{e'}$ if $e' \in E^-$ and $S_e = S_{e'}$, can be constructed in polynomial time by adding a self-attack to all $A \setminus S_e$. Further, if $2^A \setminus \mathbb{S}_{E^+}$ is non-empty, pick $S \in 2^A \setminus \mathbb{S}_{E^+}$ with minimum-weighted $e'' \in E^- \cup \{e \mid S_e \in 2^A \setminus \mathbb{S}_{E^-}, w_e = 0\}$ s.t. $S = S_{e''}$ (best solution for synthesizing set of arguments not among positive examples). One can compute e'' in polynomial time. Computing weights for all elements in E^+ and e'' yields an optimal solution AF. \square

5 CONSTRAINT-BASED SYNTHESIS OF AFs

We continue by presenting MaxSAT encodings of AF synthesis. For background on MaxSAT, recall that for a Boolean variable x , there are two literals, x and $\neg x$. A clause is a disjunction (\vee) of literals. A truth assignment τ is a function from variables to true (1) and false (0). Satisfaction is defined as usual. A weighted partial MaxSAT (or simply MaxSAT) instance consists of hard clauses φ_h , soft clauses φ_s , and a weight function w associating to each soft clause $C \in \varphi_s$ a positive weight $w(C)$. An assignment τ is a solution to a MaxSAT instance $(\varphi_h, \varphi_s, w)$ if τ satisfies φ_h . The cost of τ , $c(\tau)$, is the sum of weights of the soft clauses not satisfied by τ . A solution τ to MaxSAT instance φ is optimal if $c(\tau) \leq c(\tau')$ for any solution τ' to φ .

Let $P = (A, E^+, E^-, \sigma)$ be an AF synthesis instance with $A = \{a_1, \dots, a_n\}$ the set of arguments, E^+ the set of positive examples, E^- the set of negative examples, and $\sigma \in \{cf, adm, stb, com\}$ a semantics. In order to synthesize an optimal solution AF $F = (A, R)$ for P , we declare propositional variables $Ext_{\sigma}^{S_e}$ for each $e \in E^+ \cup E^-$, and $r_{a,b}$ for each $a, b \in A$. Now $\tau(Ext_{\sigma}^{S_e}) = 1$ indicates $S_e \in \sigma(F)$, and $\tau(r_{a,b}) = 1$ indicates $(a, b) \in R$. The hard clauses are for each $e \in E^+ \cup E^-$ equivalences of the form

$$Ext_{\sigma}^{S_e} \leftrightarrow \varphi_{\sigma}(S_e),$$

where $\varphi_\sigma(S_e)$ encodes the fact that S_e is a σ -extension. In other words, for conflict-free sets we have

$$\varphi_{cf}(S_e) = \bigwedge_{a,b \in S_e} \neg r_{a,b},$$

stating that no attacks should occur between arguments in the example. Admissible sets are encoded as

$$\varphi_{adm}(S_e) = \varphi_{cf}(S_e) \wedge \bigwedge_{a \in S_e} \bigwedge_{b \in A \setminus S_e} \left(r_{b,a} \rightarrow \bigvee_{c \in S_e} r_{c,b} \right),$$

that is, the extension is conflict-free and every argument in the set is defended. Likewise, if an example is a stable extension, it is conflict-free and its range is the whole set of arguments, encoded as

$$\varphi_{stb}(S_e) = \varphi_{cf}(S_e) \wedge \bigwedge_{a \in A \setminus S_e} \left(\bigvee_{b \in S_e} r_{b,a} \right).$$

Finally, we note that some further semantics can be covered in a similar fashion; for instance, an NP encoding for complete semantics is

$$\varphi_{com}(S_e) = \varphi_{adm}(S_e) \wedge \bigwedge_{a \in A \setminus S_e} \left(\bigvee_{b \in A} \left(r_{b,a} \wedge \bigwedge_{c \in S_e} \neg r_{c,b} \right) \right),$$

ensuring that every argument that is defended is included.

The soft clauses, on the other hand, encode the objective function of AF synthesis under minimization. For each $e \in E^+$, we have a soft clause $\text{Ext}_\sigma^{S_e}$, and for each $e \in E^-$, a soft clause $\neg \text{Ext}_\sigma^{S_e}$, with corresponding weights. An optimal solution to an AF synthesis instance is directly extracted from an optimal solution τ to the MaxSAT instance by including (a, b) to the attack structure iff $\tau(r_{a,b}) = 1$.

MaxSAT also allows for declaring additional constraints on the solution AFs of interest by encoding such constraints as hard (or soft) clauses. For instance, any particular attack (a, b) can be fixed by including the hard clause $(r_{a,b})$. Furthermore, one can for instance also synthesize an AF with the minimum number of attacks satisfying the maximum number of examples by adding soft clauses which state that the secondary preference is minimizing the number of attacks in the style of multi-level Boolean optimization [1]; in this case, in order to still guarantee that the primary objective of satisfying the maximum number of examples is met, the weights of the examples can be adjusted to be larger than the sum of the weights imposed on adding individual attacks to the solution AFs.

6 EXPERIMENTS

We implemented our MaxSAT encodings; the resulting system and benchmarks are available at <http://www.cs.helsinki.fi/group/coreo/afsynth/>. Here we present results from a first empirical evaluation of the scalability of the approach. The experiments were run on 2.83-GHz Intel Xeon E5440 quad-core machines with 32-GB memory and Debian GNU/Linux 8 using a per-instance timeout of 900 seconds. For the experiments, we used the state-of-the-art MaxSAT solver MSCG [21].

We used two different approaches to construct AF synthesis instances. The first set of benchmarks was generated based on the benchmark AFs used in the ICCMA'15 competition [26] as follows. We selected all AFs among the benchmarks that have at least five stable extensions. The number of arguments in these 17 AFs ranges from 141 to 964. For each AF, we picked uniformly at random 5

positive examples from the set of extensions. To obtain negative examples, we selected 10, 20, ..., 150 subsets of $\bigcup S_{E^+}$ uniformly at random, using $p_{\text{arg}} = \frac{\sum_{e \in E^+} |S_e| / |E^+|}{|\bigcup S_{E^+}|}$ as the probability of including an argument in a negative example. For intuition, this choice of p_{arg} makes the sizes of positive and negative examples approximately the same. Letting $A = \bigcup S_{E^+}$ resulted in instances containing 54 to 370 arguments. Further, we introduced noise by swapping each of the initial positive and negative examples with a probability $p_{\text{noise}} \in \{0, 0.25, 0.5\}$. Weights were associated to each example by picking uniformly at random integers from the interval $[1, 10]$.

The second set of benchmarks was generated using the following random model. We picked 5, 10, ..., 80 positive examples from a set of 100 arguments uniformly at random with probability $p_{\text{arg}}^+ = 0.25$. Then $|E^-| = 20, 40, \dots, 200$ negative examples were sampled from the set $A = \bigcup S_{E^+}$, and each argument was included with probability $p_{\text{arg}}^- = \frac{\sum_{e \in E^+} |S_e| / |E^+|}{|\bigcup S_{E^+}|}$. Again, each example was assigned as weight a random integer from the interval $[1, 10]$. For each choice of parameters, this procedure was repeated 10 times to obtain a representative set of benchmarks.

A summary of the results for the admissible and stable semantics is shown in Figure 2. We exclude results for ICCMA instances under admissible, as these turned out to be very easy for our approach until running out of memory due to the increasing size of the encoding. For the ICCMA instances under the stable semantics (Figure 2 left), almost every instance can be solved within the timeout limit for up to 100 examples, with a median running time of only ≈ 10 seconds at 100 examples. Increasing the noise probability clearly increases hardness; we hypothesize this to be due to the fact that by increasing noise we are increasing the number of positive examples. On the random instances the number of negative examples under the admissible semantics (Figure 2 right) clearly correlates with runtimes. Under the stable semantics (Figure 2 middle), the number of negative examples does not appear to have a noticeable effect on the runtimes. This is inline with our complexity analysis (recall Section 4), as under the stable semantics AF synthesis remains NP-complete even without any negative examples, unlike under admissible.

7 VARIANTS AND EXTENSIONS

We discuss further variants of AF synthesis: synthesis from multiple sources, synthesis under multiple semantics, and synthesis from symbolically represented examples.

Multiple Sources. The problem of AF synthesis is in a natural way applicable when the examples originate from multiple sources, e.g., collections of extensions of several source AFs, and resulting in an AF that optimally solves the task of synthesizing the union of semantical collection or examples of the different sources. This use of the AF synthesis problem shares resemblance with aggregation or merging of multiple AFs studied in [12, 11].

Multiple Semantics. So far for each AF synthesis problem we required that all examples are given w.r.t. a specific semantics. A natural generalization is to let each example individually be linked to a semantics. Formally, an example $e = (S, w, \sigma)$ is then a triple with a semantics σ , denoted by σ_e . The cost of an AF F is given by

$$\sum_{e \in E^+} w_e \cdot I(S_e \notin \sigma_e(F)) + \sum_{e \in E^-} w_e \cdot I(S_e \in \sigma_e(F)).$$

Example 5. Consider $E^+ = \{(\{a, c\}, 1, cf), (\{b, c\}, 1, stb)\}$ and $E^- = \{(\{a\}, 1, adm)\}$. This defines a unique 0-cost AF $F =$

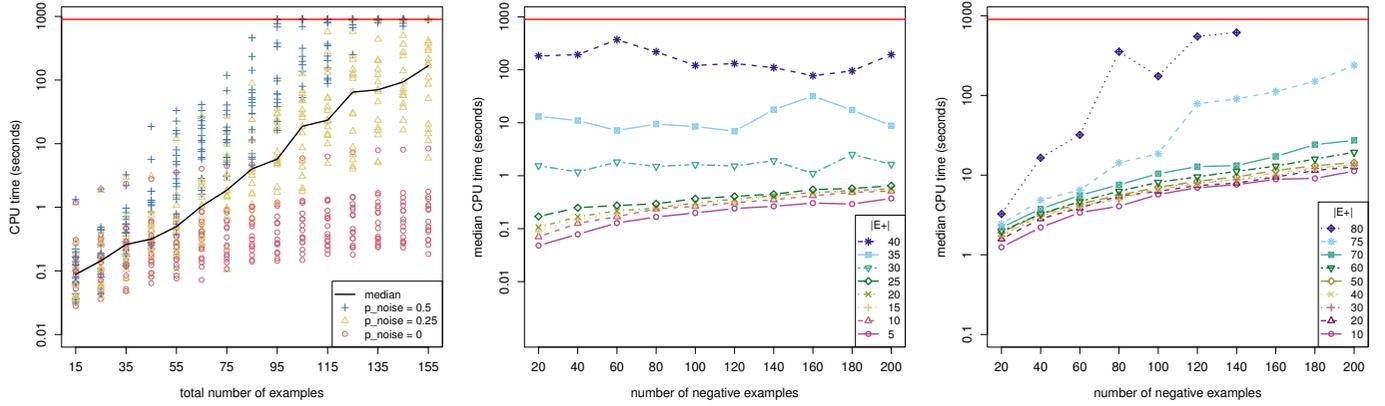


Figure 2. ICCMA instances for stable semantics (left); random instances for stable (middle) and admissible (right).

(A, R) with $A = \{a, b, c\}$ for the AF synthesis instance with multiple semantics $P = (A, E^+, E^-)$ by $R = \{(b, a)\}$.

The corresponding decision problem, i.e., for a given AF synthesis problem with multiple semantics $P = (A, E^+, E^-)$, is there an AF F with $\text{cost}(P, F) \leq k$ for an integer $k \geq 0$, does not exhibit higher computational complexity among the conflict-free, admissible, and stable semantics.

Corollary 11. *AF synthesis with multiple semantics among the conflict-free, admissible, and stable semantics is NP-complete.*

For solving AF synthesis with multiple semantics we can make use of our encodings of Section 5. In particular, we can conjoin the corresponding formulas for the different semantics and sharing the variables for attacks.

Symbolic Representation of Examples. For a set A of arguments, there can be in general up to $2^{|A|}$ positive or negative examples. This exponentiality in the input can be restrictive for large number of examples. Following ideas from [15], we note that, instead of explicit representation, examples could also be represented symbolically by encoding them succinctly in Boolean logic. This gives rise to the problem variant of AF synthesis with symbolic representation, with instances of the form $P = (A, \phi^+, \phi^-, \sigma)$, where ϕ^+ and ϕ^- are Boolean formulas. Let $\text{Mod}(\phi)$ be the set of models (satisfying assignments) of a Boolean formula ϕ , represented as sets themselves (variables assigned to true). For a given AF F , its associated cost $\text{cost}(P, F)$ is

$$\sum_{m \in \text{Mod}(\phi^+)} I(m \notin \sigma(F)) + \sum_{m \in \text{Mod}(\phi^-)} I(m \in \sigma(F)),$$

that is, unit weight is applied when a model of ϕ^+ is not a σ -extension of F and when a model of ϕ^- is a σ -extension of F .

Lemma 12. *Let ϕ be a Boolean formula, A the vocabulary of ϕ , F an AF, and σ a semantics. It holds that $|\text{Mod}(\phi)| = \text{cost}(P, F)$ for $P = (A, \phi, \phi, \sigma)$.*

Proof. Any AF F satisfies exactly $|\text{Mod}(\phi)|$ examples encoded in the formulas, since if $m \in \text{Mod}(\phi)$, then either $m \in \sigma(F)$ or $m \notin \sigma(F)$ (each implies unit weight). If $m \notin \text{Mod}(\phi)$, then both $m \in \sigma(F)$ and $m \notin \sigma(F)$ imply no weight. \square

Based on this lemma, determining the cost of a given AF for an AF synthesis instance with symbolic representation is presumably

very complex. In particular, we show $\#P$ -hardness, $\#P$ being the class of counting problems where the task is to count the number of accepting paths of a given non-deterministic polynomial-time Turing machine (see [27, 28]). As a prominent example, counting the number of models of a Boolean formula is $\#P$ -complete.

Corollary 13. *Counting the number of examples from a given AF synthesis instance with symbolic representation that are not satisfied by a given AF is $\#P$ -complete under the conflict-free, admissible, complete, and stable semantics.*

8 RELATED WORK

Before conclusions, we briefly discuss some of the most related work to ours within and beyond argumentation.

As we generalize the notion of realizability, the most closely related work to ours are recent articles focusing on realizability of AFs, and most recently, of abstract dialectical frameworks (ADFs) [8, 23, 19]. The central question studied in these works, as initiated in [15], is whether a given set of sets (of arguments) \mathbb{S} can be realized by an AF, i.e., whether an AF with $\sigma(F) = \mathbb{S}$ for a semantics σ exists. In [3, 20] realizability was studied under the restriction that the set of arguments of the constructed AF has to match exactly the set of arguments occurring in the input, i.e., is in \mathbb{S} . In [16] the authors give a construction for an AF using additional arguments in the three-valued labeling setting under the preferred and semi-stable semantics. We study the problem of synthesizing an AF that optimally matches a given set of examples semantically, even when an exact realization (a 0-cost solution) is not possible. Also, we analyze the complexity of AF synthesis, showing that, in contrast to polynomial-time results for checking realizability [15], AF synthesis is in general NP-complete. To our best knowledge, no previous systems for solving realizability have been empirically evaluated. Most recently, in [23, 19] a declarative encoding in answer set programming (ASP) for realizability was presented but not empirically evaluated. Our MaxSAT-based implementation for the AF synthesis problem also covers realizability.

In related work that incorporates AF construction from examples, [22] formally studies a logical characterization of inductive concept learning and AF learning in a multi-agent setting. In contrast to our work, they induce a rule-based theory and construct an AF based on conflicting rules. Very recently, [24] study probabilistic AF [17, 18, 25] learning with non-exact methods; we tackle the exact optimization problem of AF synthesis.

Finally, going beyond argumentation, there is a long line of research of constructing, inducing, or learning logical structures from examples, from [29] to, e.g. work for constraint acquisition [7] as well as inductive logic programming [10, 6].

9 CONCLUSIONS

We proposed AF synthesis as a generalization of the important problem of realizability in abstract argumentation, relaxing in a natural way the stringent requirements for realizability to accommodate incomplete and noisy information. From the theoretical perspective, we related AF synthesis to realizability, and analyzed the complexity of AF synthesis both in the general case and in restricted settings. Motivated by the NP-completeness of AF synthesis in general under three key AF semantics, we proposed Boolean optimization based algorithmic solutions for the problem, and empirically studied this first approach to AF synthesis using a state-of-the-art MaxSAT solver on different types of AF synthesis instances. In terms of further work, we hope to establish the computational complexity of AF synthesis under further central AF semantics, and thereafter extend the MaxSAT-based approach to cover additional semantics.

ACKNOWLEDGEMENTS

Work funded by Academy of Finland, grants 251170 COIN, 276412, and 284591; and Research Funds of the University of Helsinki.

A FORMAL PROOFS

We provide formal proofs for the results presented in Sections 3 and 4. We start by restating definitions from [15].

Definition 4. ([15, Definitions 6, 7, and 8]) *Let A be a set of arguments and $\mathbb{S} \subseteq 2^A$ a set of sets of arguments. Set \mathbb{S} is*

- *incomparable if $S \not\subseteq S'$ holds for all $S, S' \in \mathbb{S}$ with $S \neq S'$;*
- *tight if for all $S \in \mathbb{S}$ and all $a \in A$ it holds that $S \cup \{a\} \notin \mathbb{S}$ implies that there exists a $b \in S$ s.t. $\{a, b\} \not\subseteq S'$ for all $S' \in \mathbb{S}$;*
- *conflict-sensitive if for each $S, S' \in \mathbb{S}$ s.t. $S \cup S' \notin \mathbb{S}$ it holds that $\exists a, b \in S \cup S'$ s.t. $\{a, b\} \not\subseteq S''$ for all $S'' \in \mathbb{S}$; and*
- *downward closed if $\mathbb{S} = \{S' \mid S \in \mathbb{S}, S' \subseteq S\}$.*

Proof of Lemma 1. Assume that $\mathbb{S} \subseteq cf(F)$ holds. Let $S \in ImpliedCF(\mathbb{S})$. By definition it follows that for each $a, b \in S$ we have $\exists S' \in \mathbb{S}$ with $\{a, b\} \subseteq S'$. If $\mathbb{S} \subseteq cf(F)$ then $S' \in cf(F)$ and thus $\{a, b\} \in cf(F)$. This implies that $S \in cf(F)$ (each pair in S is conflict-free). \square

For proving Proposition 2 we use the following auxiliary lemma.

Lemma 14. *Let $\mathbb{S} \subseteq 2^A$ for a set A . It holds that $ImpliedCF(\mathbb{S})$ (i) contains \emptyset , (ii) is downward closed, (iii) is tight, and (iv) $\mathbb{S} \subseteq ImpliedCF(\mathbb{S})$.*

Proof. From the definition it directly follows that \emptyset is contained in $ImpliedCF(\mathbb{S})$ for any \mathbb{S} . Let $S \in ImpliedCF(\mathbb{S})$. Then for all $a, b \in S$ it holds that $\exists S' \in ImpliedCF(\mathbb{S})$ s.t. $\{a, b\} \subseteq S'$. It follows that for any $S'' \subseteq S$ it holds that $S'' \in ImpliedCF(\mathbb{S})$. To show (iii), suppose that the set is not tight, i.e., $\exists S \in ImpliedCF(\mathbb{S})$ and $a \in A$ s.t. $S \cup \{a\} \notin ImpliedCF(\mathbb{S})$ and for all $b \in S$ there exists an $S' \in ImpliedCF(\mathbb{S})$ s.t. $\{a, b\} \subseteq S'$. This implies that for all $x, y \in S \cup \{a\}$ we have $\exists S'' \in ImpliedCF(\mathbb{S})$ s.t. $\{x, y\} \subseteq S''$ and thus $S \cup \{a\} \in ImpliedCF(\mathbb{S})$ which contradicts the assumption that $ImpliedCF(\mathbb{S})$ is not tight. Finally, if $S \in \mathbb{S}$ then it follows that $S \in ImpliedCF(\mathbb{S})$ (iv). \square

Proof of Proposition 2. For the “only-if” direction, assume AF F is an optimal solution to P of cost 0, i.e., $cost(P, F) = 0$. It follows that $\mathbb{S}_{E^+} \subseteq cf(F)$. By Lemma 1 we have $ImpliedCF(\mathbb{S}_{E^+}) \subseteq cf(F)$. Thus $ImpliedCF(\mathbb{S}_{E^+}) \cap \mathbb{S}_{E^-} = \emptyset$, since $cf(F) \cap \mathbb{S}_{E^-} = \emptyset$. For the “if” direction, assume $ImpliedCF(\mathbb{S}_{E^+}) \cap \mathbb{S}_{E^-} = \emptyset$. By Lemma 14 it follows that $ImpliedCF(\mathbb{S}_{E^+})$ is tight, downward closed, and contains \emptyset . Due to [15, Proposition 5] it immediately follows that there exists an AF $F = (A', R')$ s.t. $cf(F) = ImpliedCF(\mathbb{S}_{E^+})$. Since $ImpliedCF(\mathbb{S}_{E^+}) \cap \mathbb{S}_{E^-} = \emptyset$, it follows that $cost(P, F) = 0$. In [15, Proposition 5] the set A' is specified as all arguments occurring in $ImpliedCF(\mathbb{S}_{E^+})$. If A contains more arguments, then we construct $F = (A, R)$ by extending R' with self-attacks for each $A \setminus A'$. \square

Proof of Lemma 3. Assume $\mathbb{S} \subseteq adm(F)$ holds and let $S \in ImpliedADM(\mathbb{S})$. Then $S \in ImpliedCF(\mathbb{S})$ and thus $S \in cf(F)$ (Lemma 1). Finally, every union of admissible sets which is conflict-free is again an admissible set (see [9, Lemma 1]). \square

For proving Proposition 4 we utilize the following lemma.

Lemma 15. *Let $\mathbb{S} \subseteq 2^A$ for a set A . It holds that $ImpliedADM(\mathbb{S})$ (i) contains \emptyset and (ii) is conflict-sensitive.*

Proof. Claim (i) follows from definition. Suppose (ii) does not hold, i.e., there exist $S, S' \in ImpliedADM(\mathbb{S})$ s.t. $S \cup S' \notin ImpliedADM(\mathbb{S})$ and for all $a, b \in S \cup S'$ there exists an $S'' \in ImpliedADM(\mathbb{S})$ with $\{a, b\} \subseteq S''$. It follows that $S \cup S' \in ImpliedCF(\mathbb{S})$ and $S \cup S' \in ImpliedADM(\mathbb{S})$. This contradicts the assumption that $ImpliedADM(\mathbb{S})$ is not conflict-sensitive. \square

Proof of Proposition 4. For the first claim, assume that there exists an AF F with $cost(P, F) = 0$. Then $\mathbb{S}_{E^+} \subseteq adm(F)$ and thus $ImpliedADM(\mathbb{S}_{E^+}) \subseteq adm(F)$, due to Lemma 3. For the second claim, assume $ImpliedADM(\mathbb{S}_{E^+}) \cap \mathbb{S}_{E^-} = \emptyset$ and condition 2 holds. By Lemma 15, $ImpliedADM(\mathbb{S}_{E^+})$ is conflict-sensitive and contains \emptyset . By [15, Proposition 8], there exists an AF $F' = (A', R')$ s.t. $A' \subseteq A$ and $adm(F') = ImpliedADM(\mathbb{S}_{E^+})$. Define $F = (A, R)$ by extending R' to have self-attacks for each argument in $A \setminus A'$. It follows that $adm(F) = ImpliedADM(\mathbb{S}_{E^+})$. Assuming conditions 1-2, we have $cost(P, F) = 0$. \square

Proof of Proposition 5. The claims of the proposition follow directly for the special case with $E^+ = \emptyset$. We proceed with the case that E^+ is non-empty. For the first claim, assume that F is a 0-cost solution to P . Conditions 1, 3, and 4 follow immediately. For condition 2, since $\mathbb{S}_{E^+} \subseteq stb(F)$ it follows that $\mathbb{S}_{E^+} \subseteq cf(F)$ and thus $ImpliedCF(\mathbb{S}_{E^+}) \subseteq cf(F)$. Supposing condition 2 does not hold directly violates that $\mathbb{S}_{E^+} \subseteq stb(F)$ (stable extensions are subset-maximal conflict-free sets). For the second claim, assume that conditions 1-5 hold. Then \mathbb{S}_{E^+} is a subset of the \subseteq -maximal elements of $ImpliedCF(\mathbb{S}_{E^+})$, since $\mathbb{S}_{E^+} \subseteq ImpliedCF(\mathbb{S}_{E^+})$ (Lemma 14), and by assumption of condition 2. By [15, Lemma 2] it follows that \mathbb{S}_{E^+} is tight. Further, by [15, Proposition 7] and conditions 1-5, it follows that there exists an AF $F' = (A', R')$ with $A' \subseteq A$ s.t. $stb(F') = \mathbb{S}_{E^+}$. Construct $F = (A, R)$ by extending R' to contain self-attacks for each argument in $A \setminus A'$ and attacks from each argument in \mathbb{S}_{E^+} to each $A \setminus A'$. \square

We continue with proofs for the special cases of empty E^+ or E^- .

Proof of Proposition 6. If the first condition is met, AF $F = (A, R)$ with $R = (A \times A)$ satisfies $cf(F) = adm(F) = \{\emptyset\}$ and $stb(F) =$

\emptyset . If the second condition is met, $AF F' = (A, \emptyset)$ satisfies $cf(F') = adm(F') = 2^A$. \square

We move on to proofs of complexity results.

Proof of Proposition 7. For an AF synthesis instance $P = (A, E^+, E^-, cf)$ membership in NP follows from a guess of an AF $F = (A, R)$, since $cost(P, F)$ can be computed in polynomial time.

For hardness, we provide a reduction from the satisfiability problem of conjunctive normal form (CNF) Boolean formulas. Let ϕ be a propositional formula in 3-CNF over variables $X = \{x_1, \dots, x_n\}$, $|X| = n$ and set of clauses C . Let $b = n + 1$.

$$E^+ = \{(\{x_i, x_i^T\}, 1) \mid x_i \in X\} \cup \quad (1)$$

$$\{(\{x_i, x_i^F\}, 1) \mid x_i \in X\} \cup \quad (2)$$

$$\{(\{x_i^T, x_i^F\}, b) \mid x_i \in X\} \cup \quad (3)$$

$$\{(\{y, z\}, b) \mid x_i, x_j \in X, i \neq j, \quad (4)$$

$$y \in \{x_i, x_i^T, x_i^F\}, z \in \{x_j, x_j^T, x_j^F\}\}$$

$$E^- = \{(\{x_i, x_i^T, x_i^F\}, b) \mid x_i \in X\} \cup \quad (5)$$

$$\{(\{x_i, x_i^T \mid x_i \in c\} \cup \{x_i, x_i^F \mid \neg x_i \in c\}, b) \mid c \in C\} \quad (6)$$

Let $P = (A, E^+, E^-, cf)$ be the constructed instance for AF synthesis with bound $k = n$. Intuitively, one cannot satisfy all examples of forms (1), (2), (3), and (5) simultaneously, and due to the chosen weights and cost limit, one has to violate either (1) or (2) for a given $x_i \in X$, thus “choosing” a truth assignment over X (true iff an attack between x_i and x_i^F is synthesized). We now claim that there exists an AF $F = (A, R)$ with $cost(P, F) \leq n$ iff ϕ is satisfiable.

“Only-if” direction: assume that $F = (A, R)$ has $cost(P, F) \leq n$. Then all examples with weight $n + 1$ are satisfied by F . It is immediate that for each $x_i \in X$ we have either $\{x_i, x_i^T\} \in cf(F)$ or $\{x_i, x_i^F\} \in cf(F)$ but not both (if both would be conflict-free then together with $\{x_i^T, x_i^F\}$ being conflict-free in F implies that $\{x_i, x_i^T, x_i^F\}$ if conflict-free which contradicts that cost of F is lower than $n + 1$ (5); if none of the sets with weight 1 are conflict-free for $x_i \in X$, then overall cost would be over n as well). This straightforwardly defines a truth assignment $\tau(x_i) = 0$ iff $\{x_i, x_i^T\} \in cf(F)$ and 1 otherwise. Suppose τ does not satisfy ϕ . Then there exists a $c \in C$ s.t. $\tau \not\models c$ and τ does not satisfy any literal l in c .

$$\tau(l) = 0, \forall l \in c$$

iff $\forall l \in c$

$$l = x_i \text{ implies } \tau(x_i) = 0 \text{ and}$$

$$l = \neg x_i \text{ implies } \tau(x_i) = 1$$

iff $\forall l \in c$

$$l = x_i \text{ implies } \{x_i, x_i^T\} \in cf(F) \text{ and}$$

$$l = \neg x_i \text{ implies } \{x_i, x_i^F\} \in cf(F)$$

iff $\{x_i, x_i^T \mid x_i \in c\} \cup \{x_i, x_i^F \mid \neg x_i \in c\} \in cf(F)$ (*)

only-if $cost(P, F) \geq n + 1$

Conclusion (*) follows from (4): for each $x_i, x_j \in X$ with $x_i \neq x_j$ no attacks are in between sets $\{x_i, x_i^T, x_i^F\}$ and $\{x_j, x_j^T, x_j^F\}$. “If” direction: assume ϕ is satisfiable. Construct AF $F = (A, R)$ with $R = \{(x_i, x_i^T) \mid \tau(x) = 0\} \cup \{(x_i, x_i^F) \mid \tau(x) = 1\}$. It is immediate that F satisfies all non-unit weighted examples except for (6), which follows from similar consideration as in the only-if direction. Finally, cost of F is n . \square

Proof sketch of Proposition 8. NP-completeness for admissible semantics follows the same reasoning as proof of Proposition 7. For the “only-if” direction, just note that the same unit-weighted examples are mutually exclusive for a solution AF as in the conflict-free case. For “if” direction, the constructed AF has mutual attacks instead of uni-directional ones (conflict-free sets are then admissible). \square

Proof sketch of Proposition 9. Membership follows from a non-deterministic guess of an AF F and checking each example individually whether it is satisfied (which can be done in polynomial time).

For hardness, we provide a reduction from the Boolean satisfiability problem. Let ϕ be a Boolean formula in CNF, with vocabulary X , with $|X| = n$, and set of clauses C . Let $b = n + 1$.

$$A = X \cup \{x^T, x^F, d_x \mid x \in X\} \cup \{d'_c, d''_c \mid c \in C\} \cup \{d\} \quad (7)$$

$$E^+ = \{(\{d'_c\} \cup \{x^T \mid x \in c\} \cup \{x^F \mid \neg x \in c\}, b) \mid c \in C\} \cup \quad (8)$$

$$\{(\{d'_c, d''_c, d\}, b) \mid c \in C\} \cup \quad (9)$$

$$\{(\{x^T, x^F, d_x\}, b) \mid x \in X\} \cup \quad (10)$$

$$\{(\{x^T, d_x, d\}, 1) \mid x \in X\} \cup \quad (11)$$

$$\{(\{x^F, d_x, d\}, 1) \mid x \in X\} \quad (12)$$

Let $P = (A, E^+, \emptyset, stb)$ and bound $k = n$. We claim that ϕ satisfiable iff there exists an AF F s.t. $cost(P, F) \leq n$.

Assume such an AF F exists. It is immediate that all examples with weight $n + 1$ are satisfied by F . For each $x \in X$ it holds that exactly one example of $\{(\{x^T, d_x, x\}, 1), (\{x^F, d_x, x\}, 1)\}$ is satisfied. If none of them is satisfied the cost of F would be higher than n . If both are satisfied, then there is no attack between x^T, x^F, d_x , and d , thus $\{x^T, x^F, d_x\}$ cannot be stable (does not attack d). This defines a truth assignment $\tau(x) = 0$ iff $(\{x^T, d_x, x\}, 1)$ is satisfied by F . We claim that $\tau \models \phi$. Suppose the contrary, then $\exists c \in C$ with $\tau \not\models c$ and all literals in c are not satisfied by τ . Consider the set $\{d'_c\} \cup \{x^T \mid x \in c\} \cup \{x^F \mid \neg x \in c\}$, which must be stable in F by assumption. By construction of τ and example (9), it follows that no argument in that set attacks d , thus it cannot be stable, which contradicts the assumption that τ does not satisfy ϕ .

Assume that ϕ is satisfiable. Construct AF $F = (A, R)$ with

$$R = \{(d''_c, d_x), (d_x, d''_c) \mid c \in C, x \in X\} \cup \quad (13)$$

$$\{(d''_c, x^T), (d''_c, x^F), (x^T, d''_c), (x^F, d''_c) \mid c \in C, x \in X\} \cup \quad (14)$$

$$\{(a, b), (b, a) \mid a \in \{d'_c, d''_c\}, b \in \{d'_c, d''_c\}, c, c' \in C, c \neq c'\} \cup \quad (15)$$

$$\{(d_x, d_y), (d_x, z), (z, d_x) \mid x, y \in X, x \neq y, z \in \{y^T, y^F\}\} \cup \quad (16)$$

$$\{(d_x, d'_c), (d'_c, d_x) \mid c \in C, x \in X\} \cup \quad (17)$$

$$\{(d'_c, x^T), (x^T, d'_c) \mid c \in C, x \in X, x \notin c\} \cup \quad (18)$$

$$\{(d'_c, x^F), (x^F, d'_c) \mid c \in C, x \in X, \neg x \notin c\} \cup \quad (19)$$

$$\{(x^T, d), (d, x^T) \mid \tau(x) = 1\} \cup \quad (20)$$

$$\{(x^F, d), (d, x^F) \mid \tau(x) = 0\}. \quad (21)$$

Briefly put, this involved construction adds mutual attacks between arguments to ensure that all the examples (9) and (10) are stable. A mutual attack is added between x^T and d (x^F and d) based on the truth assignment τ , violating one of the unit weighted examples. Finally, examples of form (8) are satisfied, since one of the arguments in these sets (except d'_c) attacks d due to assumption that τ satisfies ϕ . Intuitively, the examples encoding the clauses (8) are satisfied since one of the arguments corresponding to the chosen truth assignment that satisfies one of the literals attacks d . \square

REFERENCES

- [1] Josep Argelich, Inês Lynce, and João P. Marques-Silva, 'On solving Boolean multilevel optimization problems', in *Proc. IJCAI*, ed., Craig Boutilier, pp. 393–398. AAAI Press, (2009).
- [2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin, 'An introduction to argumentation semantics', *Knowledge Engineering Review*, **26**(4), 365–410, (2011).
- [3] Ringo Baumann, Wolfgang Dvořák, Thomas Linsbichler, Hannes Strass, and Stefan Woltran, 'Compact argumentation frameworks', in *Proc. ECAI*, eds., Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 69–74. IOS Press, (2014).
- [4] Ringo Baumann and Hannes Strass, 'On the maximal and average numbers of stable extensions', in *Proc. TFAA*, eds., Elizabeth Black, Sanjay Modgil, and Nir Oren, volume 8306 of *Lecture Notes in Computer Science*, pp. 111–126. Springer, (2013).
- [5] T.J.M. Bench-Capon and Paul E. Dunne, 'Argumentation in artificial intelligence', *Artificial Intelligence*, **171**(10-15), 619–641, (2007).
- [6] Francesco Bergadano and Daniele Gunetti, *Inductive logic programming - from machine learning to software engineering*, MIT Press, 1996.
- [7] Christian Bessiere, Frédéric Koriche, Nadjib Lazaar, and Barry O'Sullivan, 'Constraint acquisition', *Artificial Intelligence*, (2015).
- [8] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes P. Wallner, and Stefan Woltran, 'Abstract dialectical frameworks revisited', in *Proc. IJCAI*, ed., Francesca Rossi, pp. 803–809. AAAI Press / IJCAI, (2013).
- [9] Martin Caminada, 'Comparing two unique extension semantics for formal argumentation: Ideal and eager', in *Proc. BNAIC*, eds., Mehdi Dastani and Edwin de Jong, pp. 81–87, (2007).
- [10] Jesse Davis and Jan Ramon, eds. *Inductive Logic Programming - 24th International Conference, ILP 2014, Nancy, France, September 14-16, 2014, Revised Selected Papers*, volume 9046 of *Lecture Notes in Computer Science*. Springer, 2015.
- [11] Jérôme Delobelle, Adrian Haret, Sébastien Konieczny, Jean-Guy Mailly, Julien Rossit, and Stefan Woltran, 'Merging of abstract argumentation frameworks', in *Proc. KR*, eds., Chitta Baral, James P. Delgrande, and Frank Wolter, pp. 33–42. AAAI Press, (2016).
- [12] Jérôme Delobelle, Sébastien Konieczny, and Srdjan Vesic, 'On the aggregation of argumentation frameworks', in *Proc. IJCAI*, eds., Qiang Yang and Michael Wooldridge, pp. 2911–2917. AAAI Press, (2015).
- [13] Martin Diller, Adrian Haret, Thomas Linsbichler, Stefan Rümmele, and Stefan Woltran, 'An extension-based approach to belief revision in abstract argumentation', in *Proc. IJCAI*, eds., Qiang Yang and Michael Wooldridge, pp. 2926–2932. AAAI Press, (2015).
- [14] Phan Minh Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games', *Artificial Intelligence*, **77**(2), 321–358, (1995).
- [15] Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran, 'Characteristics of multiple viewpoints in abstract argumentation', *Artificial Intelligence*, **228**, 153–178, (2015).
- [16] Sjur Kristoffer Dyrkolbotn, 'How to argue for anything: Enforcing arbitrary sets of labellings using AFs', in *Proc. KR*, eds., Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, pp. 626–629. AAAI Press, (2014).
- [17] Anthony Hunter, 'Probabilistic qualification of attack in abstract argumentation', *International Journal of Approximate Reasoning*, **55**(2), 607–638, (2014).
- [18] Hengfei Li, Nir Oren, and Timothy J. Norman, 'Probabilistic argumentation frameworks', in *Proc. TFAA*, eds., Sanjay Modgil, Nir Oren, and Francesca Toni, volume 7132 of *Lecture Notes in Computer Science*, pp. 1–16. Springer, (2011).
- [19] Thomas Linsbichler, Jörg Pührer, and Hannes Strass, 'Characterizing realizability in abstract argumentation', in *Proc. NMR*, eds., Gabriele Kern-Isberner and Renata Wassermann, pp. 85–94, (2016).
- [20] Thomas Linsbichler, Christof Spanring, and Stefan Woltran, 'The hidden power of abstract argumentation semantics', in *Proc. TFAA*, eds., Elizabeth Black, Sanjay Modgil, and Nir Oren, volume 9524 of *Lecture Notes in Computer Science*, pp. 146–162. Springer, (2015).
- [21] Antonio Morgado, Alexey Ignatiev, and Joao Marques-Silva, 'MSCG: Robust core-guided MaxSAT solving', *Journal on Satisfiability, Boolean Modeling and Computation*, **9**, 129–134, (2015).
- [22] Santiago Ontañón, Pilar Dellunde, Lluís Godo, and Enric Plaza, 'A de-
feasible reasoning model of inductive concept learning from examples and communication', *Artificial Intelligence*, **193**, 129–148, (2012).
- [23] Jörg Pührer, 'Realizability of three-valued semantics for abstract dialectical frameworks', in *Proc. IJCAI*, eds., Qiang Yang and Michael Wooldridge, pp. 3171–3177. AAAI Press, (2015).
- [24] Régis Riveret and Guido Governatori, 'On learning attacks in probabilistic abstract argumentation', in *Proc. AAMAS*, eds., Catholijn M. Jonker, Stacy Marsella, John Thangarajah, and Karl Tuyls, pp. 653–661. ACM, (2016).
- [25] Régis Riveret, Dimitrios Korkinof, Moez Draief, and Jeremy V. Pitt, 'Probabilistic abstract argumentation: an investigation with boltzmann machines', *Argument & Computation*, **6**(2), 178–218, (2015).
- [26] Matthias Thimm, Serena Villata, Federico Cerutti, Nir Oren, Hannes Strass, and Mauro Vallati, 'Summary report of the first international competition on computational models of argumentation', *AI Magazine*, **37**(1), 102–104, (2016).
- [27] Leslie G. Valiant, 'The complexity of computing the permanent', *Theoretical Computer Science*, **8**, 189–201, (1979).
- [28] Leslie G. Valiant, 'The complexity of enumeration and reliability problems', *SIAM Journal on Computing*, **8**(3), 410–421, (1979).
- [29] Leslie G. Valiant, 'A theory of the learnable', *Communications of the ACM*, **27**(11), 1134–1142, (1984).

Budgeted Multi-Armed Bandit in Continuous Action Space

Francesco Trovò, Stefano Paladino, Marcello Restelli, Nicola Gatti¹

Abstract. Multi-Armed Bandits (MABs) have been widely considered in the last decade to model settings in which an agent wants to learn the action providing the highest expected reward among a fixed set of available actions during the operational life of a system. Classical techniques provide solutions that minimize the regret due to learning in settings where selecting an arm has no cost. Though, in many real world applications the learner has to pay some cost for pulling each arm and the learning process is constrained by a fixed budget B . This problem is addressed in the literature as the Budgeted MAB (BMAB). In this paper, for the first time, we study the problem of Budgeted Continuous-Armed Bandit (BCAB), where the set of the possible actions consists in a continuous set (e.g., a range of prices) and the learner suffers from a random reward and cost at each round. We provide a novel algorithm, named B-Zoom, which suffers a regret of $\tilde{O}(B^{\frac{d+1}{d+2}})$, where d is the Zooming dimension of the problem. Finally, we provide an empirical analysis showing that, despite a lower average performance, the proposed approach is more robust to adverse settings as compared to existing algorithms designed for BMAB.

1 Introduction

In a Multi-Armed Bandit (MAB) problem [3], an agent, called *learner*, is allowed to select a single option, called *arm*, from a finite number of available options and to observe the corresponding stochastic reward. The techniques developed for a MAB problem minimize the loss, called *regret*, incurred during the learning process and provide theoretical guarantees about convergence to the optimal arm. Most regret-minimization algorithms available in the literature provide solutions to the case in which there is a constraint over the maximum number of rounds the agent is allowed to pull arms. However, in many applications, an agent is subject to different constraints. A very common case is when the learner has a fixed budget which she uses to pay a stochastic cost associated with the pulling of a specific arm. Simply, the constraint over the budget reduces to the constraint over the maximum number of rounds when each arm has a fixed unitary cost.

In this setting, known as *Budgeted MAB* (BMAB) [10], the learner is given a fixed budget in advance and she is allowed to pull arms until the budget has been totally spent. The BMAB is able to model a wide range of concrete applications. For instance, bidding in Sponsored Search Auctions (SSA) [6] when an advertiser has no information neither about the probability of being clicked (usually called click-through rate) nor about the cost of being clicked is a BMAB

problem. In the same field, the problem of optimizing an advertising campaign presents a similar model. Another application that can be modeled by means of a BMAB problem consists in determining the optimal sensor to interrogate in a wireless sensor network scenario [16, 18]. More precisely, when we retrieve information from a sensor, we gain information about the monitored process and, at the same time, we spend budget in terms of energetic costs. Also the problem of a service provider trying to balance the costs of the employed resources and the revenues gained by the provided services fits the BMAB model [2].

In many applications, the use of a finite set of arms provides an extremely raw model of the situation one studies, potentially forcing the learner to pull only suboptimal arms and thus to suffer a linear regret over time (or budget). Natural examples of spaces of continuous arms are prices and costs. In this paper, to the best of our knowledge, we study the first generalization of the BMAB to continuous arm spaces, named the *Budgeted Continuous-Armed Bandit* (BCAB). In order to cope efficiently with continuous space, we need additional assumptions over the regularity of the average reward and cost functions. In particular, as customary in the literature on Continuous-Armed Bandits (CAB), we assume Lipschitz continuity over the expected reward and cost functions. Such an assumption is largely supported by real-world scenarios, e.g., in SSAs similar bids have similar expected rewards and payments.

Related works A number of recent results on sequential learning settings whose stopping time depends on a fixed budget can be found in the literature. Some of them consider fixed costs [1, 7, 11, 17], while others assume to have stochastic ones [10, 19, 20]. There is a wide literature studying settings in which exploration and exploitation phases are separate and only the exploration phase is subject to costs [1, 7, 11]. Only few works consider settings with deterministic costs without separating exploration and exploitation phases. In particular, in [17], the authors tackle the problem by relying on an approximated optimization technique of the unbounded knapsack problem, which hardly generalizes to the setting with stochastic costs. [5, 10, 19, 20] consider the BMAB problem with stochastic rewards and costs over a discrete space of arms. In [10], the authors propose a frequentist approach that relies on UCB-like bounds [3] achieving an $O(\log B)$ regret for a generic instance of the BMAB having budget B . This approach has been extended to consider also linear bandits in [19].² [20] considers the Thompson sampling algorithm to solve the budgeted MAB in the same setting having the same theoretical upper bound by relying on a Bayesian framework. Finally in [5], an algorithm providing a distribution-free bound of $\tilde{O}(\sqrt{B})$ has been

¹ Politecnico di Milano, Italy, email: {francesco1.trovo, stefano.paladino, marcello.restelli, nicola.gatti}@polimi.it.

² In linear bandit problems, the reward function is forced to be linear.

proposed.³

The literature provides a large number of results analyzing the CAB setting without costs and budget [4, 8, 9, 12, 13, 14, 15]. The CAB problem is arbitrarily hard in the general setting in which the reward function can be arbitrary, presenting $\Theta(T)$ regret over a horizon of T rounds. Positive results can be obtained when the reward function exhibits some structure. Under the assumption of Lipschitzianity of the expected reward functions, the lower bound over the regret is $\Omega(T^{2/3})$ for the one dimensional version of the problem [14]. The former techniques for the CAB problem are based, initially, on the discretization of the action space by exploiting the structure of the problem and, subsequently, on the adoption of MAB techniques over the discretized problem [4, 9, 15] (let us notice that the application of MAB algorithms to any “blind” discretization may lead to a regret $\Omega(T)$). Recently, new techniques adopting a different, more efficient, approach which changes the set of arms during time on the basis of the observed performance of the arms previously chosen have been developed. One of the most promising techniques for this setting is the Zooming algorithm [12, 13]. This algorithm is designed for CAB problems in metric spaces and, differently from most of the previous works, that consider a uniform discretization of the space which is fixed in advance, it starts from a single arm and, if needed, automatically adds arms over time in the domain. Moreover, it provides an upper bound on the regret of $\tilde{O}(T^{\frac{d+1}{d+2}})$, where d is the *Zooming dimension* associated with the reward function (in the single dimension version $d = 1$ and therefore the Zooming algorithm matches the lower bound). Another algorithm designed for the same setting is called HOO [8]. It is based on the idea of using a search tree to find the best arm. Although it assures an upper bound comparable to the Zooming one of $\tilde{O}(T^{\frac{p+1}{p+2}})$, where p is the packing dimension, it may have higher computational complexity.

Original contributions Our original contributions are as follow. We design the first algorithm, named B-Zoom, able to work in the BCAB setting; the B-Zoom algorithm extends the Zooming algorithm to the case with budget. We provide a theoretical regret analysis of the B-Zoom algorithm, showing that it suffers a regret $\tilde{O}(B^{\frac{d+1}{d+2}})$ matching the regret of the Zooming algorithm in the case in which the BCAB setting reduces to the CAB one (i.e., $B = T$ and unitary cost for all the arms). We experimentally evaluate B-Zoom comparing its performance w.r.t. that of a number of frequentist algorithms.

2 Problem formulation

We denote by $\mathcal{A} \subseteq [0, 1]$ the space of the available actions, also called arms, and by x a generic arm. In the BCAB setting, at each round $t \in \mathbb{N}^+$, a learner is allowed to choose an arm $x_t \in \mathcal{A}$. She receives a reward $r_t(x_t)$ and incurs a cost $c_t(x_t)$. Rewards $r_t(x)$ are realizations of i.i.d. random variables $R_t(x) \sim \mathcal{D}_r([0, 1])$, where $\mathcal{D}_r([0, 1])$ is a generic probability density function (pdf) over support $[0, 1]$, and expected value $\mathbb{E}[R_t(x)] = \mu_r(x)$ with $\mu_r : \mathcal{A} \rightarrow [0, 1]$. Costs $c_t(x)$ are realizations of i.i.d. random variables $C_t(x) \sim \mathcal{D}_c([0, 1])$, where $\mathcal{D}_c([0, 1])$ is a generic pdf over support $[0, 1]$, and expected value $\mathbb{E}[C_t(x)] = \mu_c(x)$ with $\mu_c : \mathcal{A} \rightarrow [\lambda, 1]$. Here $\lambda > 0$ is a known lower bound on the average cost of an action, needed to exclude the case with costless actions.⁴

³ We write $u_n = \tilde{O}(v_n)$ when $u_n = O(v_n)$ up to a logarithmic factor.

⁴ Without loss of generality, we considered from now on the setting in which the arms are selected in $\mathcal{A} \equiv [0, 1]$ and average reward $\mu_r(x)$ and cost functions $\mu_c(x)$ have images in $[0, 1]$ for each $x \in \mathcal{A}$. In the case that

A fixed budget $B > 0$ is available to the learner at the beginning of the learning process. We denote by $B(t) := B - \sum_{i=1}^{t-1} c_i(x_i)$ the residual budget available at round t due to the costs incurred in having pulled the arms during the previous $t - 1$ rounds. As customary in the previous works on budget, in the case the learner is not able to pay at t for the cost of the chosen arm x_t , she is forced to stop and does not gain any reward due to x_t . Moreover, we assume that the reward function $\mu_r(x)$ and cost function $\mu_c(x)$ are Lipschitz with known constant L_r and L_c , respectively. These assumptions are usual in Lipschitz bandits and here required to solve our problem.

A generic policy \mathcal{U} for a BCAB problem is an algorithm able to decide the arm x_t to pull at round t , on the basis of the history in terms of previous realizations of the rewards $\{r_1(x_1), \dots, r_{t-1}(x_{t-1})\}$, costs $\{c_1(x_1), \dots, c_{t-1}(x_{t-1})\}$ and pulled arms $\{x_1, \dots, x_{t-1}\}$. We define the *stopping time* t_a of a generic policy \mathcal{U} which chooses arm x_t at round t the longest t such that $B(t) \geq 0$. Notice that t_a is a random variable depending on the costs $C_t(x)$ and the initial budget B .

In a BCAB problem, a policy should be able to select a sequence of arms that minimizes the amount of budget spent and maximizes the reward collected during the process. The loss of a generic policy \mathcal{U} in a BCAB problem with budget B is represented by the *pseudo-regret* $\mathcal{R}(B)$:

$$\mathcal{R}(B) = \mathcal{R}^*(B) - \mathbb{E}_{r,c} \left[\sum_{t=1}^{t_a} r_t(x_t) \right], \quad (1)$$

where, $\mathcal{R}^*(B)$ is the optimal expected total reward when the distribution of rewards $\mathcal{D}_r([0, 1])$ and costs $\mathcal{D}_c([0, 1])$ are known, i.e., the one which solves the following stochastic optimization problem:

$$\max_{\mathcal{U}} \mathbb{E} \left[\sum_{t=1}^{t_a} r_t(x_t) \right], \text{ s.t. } \sum_{t=1}^{t_a} c_t(x_t) \leq B,$$

where the expected value $\mathbb{E}[\cdot]$ is taken w.r.t. the randomness associated to the policy \mathcal{U} , the rewards, and the costs.

3 The proposed method

In what follows, we introduce our algorithm named B-Zoom to tackle the BCAB problem. The B-Zoom algorithm is based on the idea of the Zooming algorithm and is its extension to the case where a fixed budget B is available and the learner incurs a stochastic cost $C_t(x)$ in pulling arm x at round t . After a brief description of its main features, we provide its theoretical analysis, giving an upper bound over the pseudo-regret $\mathcal{R}(B)$.

3.1 The B-Zoom algorithm

Initially, we introduce the following function on which our algorithm is based:

Definition 1. Given an average reward function $\mu_r(x)$ and an average cost function $\mu_c(x)$, we define the expected reward-to-cost ratio function $\mu : \mathcal{A} \rightarrow [0, \frac{1}{\lambda}]$ as:

$$\mu(x) = \frac{\mu_r(x)}{\mu_c(x)}.$$

the space and the average functions are over different domains, a rescaling procedure should be performed so they have values and images in $[0, 1]$.

Algorithm 1 The B-Zoom Algorithm

```

1: Input: Budget  $B$ , Minimum average cost  $\lambda$ , Arm support set  $\mathcal{A}$ 
2:  $i_{ph} = 0$ 
3:  $B(0) \leftarrow B$ 
4:  $t \leftarrow 0$ 
5: while  $B(t) > 0$  do
6:    $i_{ph} \leftarrow i_{ph} + 1$ 
7:    $X(i_{ph}) \leftarrow \emptyset$ 
8:   for  $t \in \{2^{i_{ph}-1}, \dots, 2^{i_{ph}} - 1\}$  do
9:     if  $B(t-1) > 0$  then
10:       $\mathcal{C} \leftarrow \cup_{x \in X(i_{ph})} \mathcal{B}(E_{t-1}(x), x)$ 
11:       $\mathcal{N}\mathcal{C} \leftarrow \mathcal{A} \setminus \mathcal{C}$ 
12:      if  $\mathcal{N}\mathcal{C} \neq \emptyset$  then
13:        Randomly pick  $x \in \mathcal{N}\mathcal{C}$ 
14:         $X(i_{ph}) \leftarrow X(i_{ph}) \cup \{x\}$ 
15:         $\bar{r}_t(x) \leftarrow 0$ 
16:         $\bar{c}_t(x) \leftarrow 0$ 
17:         $n_t(x) \leftarrow 0$ 
18:         $u_t(x) \leftarrow +\infty$ 
19:         $E_t(x) \leftarrow +\infty$ 
20:      Play arm  $x_t$  s.t.:  $x_t = \arg \max_{x \in X(i_{ph})} u_t(x)$ 
21:      Suffer cost  $c_t(x_t)$ 
22:       $B(t) \leftarrow B(t-1) - c_t(x_t)$ 
23:      if  $B(t) \geq 0$  then
24:        Gain reward  $r_t(x_t)$ 
25:         $n_t(x_t) \leftarrow n_{t-1}(x_t) + 1$ 
26:         $\bar{r}_t(x_t) \leftarrow \frac{(n_t(x_t)-1)\bar{r}_{t-1}(x_t) + r_t(x_t)}{n_t(x_t)}$ 
27:         $\bar{c}_t(x_t) \leftarrow \frac{(n_t(x_t)-1)\bar{c}_{t-1}(x_t) + c_t(x_t)}{n_t(x_t)}$ 
28:         $E_t(x_t) \leftarrow \frac{1}{\lambda} \left(1 + \frac{1}{\lambda}\right) \sqrt{\frac{8i_{ph} + \ln(4)}{n_t(x_t)}}$ 
29:         $u_t(x) \leftarrow \frac{\bar{r}_t(x)}{\max\{\lambda, \bar{c}_t(x)\}} + 2E_t(x)$ 

```

We can show that function $\mu(x)$ is Lipschitz when both $\mu_r(x)$ and $\mu_c(x)$ are Lipschitz.

Lemma 1. *Given an average reward function $\mu_r : \mathcal{A} \rightarrow [0, 1]$, L_r -Lipschitz, and an average cost function $\mu_c : \mathcal{A} \rightarrow [\lambda, 1]$, $\lambda > 0$, L_c -Lipschitz, the average reward-to-cost ratio function $\mu(x)$ is L' -Lipschitz with $L' \leq \frac{L_c + L_r}{\lambda^2}$.*

Proof. Thanks to the Lipschitz assumption over functions $\mu_r(x)$ and $\mu_c(x)$, we have:

$$\begin{aligned} |\mu_r(x_1) - \mu_r(x_2)| &\leq L_r |x_1 - x_2| & \forall x_1, x_2 \in [0, 1] \\ |\mu_c(x_1) - \mu_c(x_2)| &\leq L_c |x_1 - x_2| & \forall x_1, x_2 \in [0, 1] \end{aligned}$$

thus:

$$\begin{aligned} |\mu(x_1) - \mu(x_2)| &= \\ &= \left| \frac{\mu_r(x_1)}{\mu_c(x_1)} - \frac{\mu_r(x_2)}{\mu_c(x_2)} \right| = \frac{|\mu_r(x_1)\mu_c(x_2) - \mu_r(x_2)\mu_c(x_1)|}{\mu_c(x_1)\mu_c(x_2)} \\ &\leq \frac{1}{\lambda^2} |\mu_r(x_1)\mu_c(x_2) - \mu_r(x_1)\mu_c(x_1) + \\ &\quad + \mu_r(x_1)\mu_c(x_1) - \mu_r(x_2)\mu_c(x_1)| \\ &\leq \frac{1}{\lambda^2} (\mu_r(x_1)|\mu_c(x_2) - \mu_c(x_1)| + |\mu_r(x_1) - \mu_r(x_2)|\mu_c(x_1)) \\ &\leq \frac{L_c + L_r}{\lambda^2} |x_1 - x_2| \leq L' |x_1 - x_2|. \end{aligned}$$

□

From now on, without loss of generality, we assume $L' = 1$ Lipschitz constant for the function $\mu(\cdot)$. Indeed, a scaling procedure can be always performed to obtain $L' = 1$.

The B-Zoom algorithm pseudo-code is presented in Algorithm 1. The functioning of the algorithm is split into temporal phases, where the i -th phase is denoted by i_{ph} and the length of phase i_{ph} is $2^{i_{ph}-1}$

rounds. Each phase i_{ph} is associated with a (potentially different) subset of arms $X(i_{ph}) \subset \mathcal{A}$ named *active arms*, which is initially empty and is incrementally populated over the phase i_{ph} . Furthermore, each active arm $x \in X(i_{ph})$ is associated with an open ball $\mathcal{B}(E_t(x), x)$ with radius $E_t(x)$ and centered in x , where $E_t(x)$ is a confidence radius defined as follows:

$$E_t(x) = \begin{cases} \frac{1}{\lambda} \left(1 + \frac{1}{\lambda}\right) \sqrt{\frac{8i_{ph} + \ln(4)}{n_t(x)}} & \text{if } n_t(x) > 0 \\ +\infty & \text{otherwise} \end{cases},$$

where $n_t(x) = \sum_{i=1}^t I\{x_i = x\}$ is the number of rounds an arm x has been pulled up to round t and $I\{\cdot\}$ is the indicator function. The confidence radius $E_t(x)$ varies over time, reducing as the number of rounds an arm x has been pulled increases. Notice that, if an active arm x has never been pulled, its ball $\mathcal{B}(E_t(x), x)$ contains entirely \mathcal{A} , the radius $E_t(x)$ being infinite independently of t .

At time t , we define the *covering set* of the active arms $\mathcal{C} = \cup_{x \in X(i_{ph})} \mathcal{B}(E_{t-1}(x), x)$ as the union of the balls of all the active arms. We say that a set \mathcal{A} is *covered* by \mathcal{C} if and only if $\mathcal{C} \supseteq \mathcal{A}$. The covering of a set \mathcal{A} by \mathcal{C} can be easily checked by means of a *covering oracle* (as we discuss below for the sake of presentation); we denote by $\mathcal{N}\mathcal{C} := \mathcal{A} \setminus \mathcal{C}$ the subset of \mathcal{A} that is not covered by \mathcal{C} .

At each round t , the first task accomplished by the B-Zoom algorithm is to decide whether or not to add new active arms. The *rationale* whereby such a decision is taken follows. If at round t the arm space \mathcal{A} is not covered by the covering set \mathcal{C} of active arms $X(i_{ph})$, the algorithm randomly draws an arm $x \in \mathcal{N}\mathcal{C}$ with an arbitrary probability distribution and add it to the active arm set $X(i_{ph})$. Notice that, independently of the shape of $\mathcal{N}\mathcal{C}$, no more than one active arm is added at each round t . Indeed, once an active arm has been added, the radius of its ball is, by definition, infinite and therefore the new covering set \mathcal{C} covers the whole arm space \mathcal{A} .

At each round t , once the coverage of \mathcal{A} by \mathcal{C} has been evaluated and, potentially, a new active arm has been introduced in $X(i_{ph})$, the B-Zoom algorithm plays the arm $x_t \in X(i_{ph})$ having the maximum upper bound $u_t(x_t)$ defined as:

$$u_t(x) = \frac{\bar{r}_t(x)}{\max\{\lambda, \bar{c}_t(x)\}} + 2E_t(x), \quad (2)$$

where $\bar{r}_t(x) = \frac{\sum_{i=1}^t r_i(x) I\{x_i=x\}}{n_t(x)}$ and $\bar{c}_t(x) = \frac{\sum_{i=1}^t c_i(x) I\{x_i=x\}}{n_t(x)}$ are the estimated average reward and cost for arm x , respectively. The idea behind the computation of $u_t(x)$ is that we want to upper bound (in high probability) the average reward-to-cost ratio function $\mu(x)$ with the first term in the r.h.s. of Equation 2 plus a radius $E_t(x)$ and we consider another radius $E_t(x)$ to be able to bound $\mu(\tilde{x})$ for all arms $\tilde{x} \in \mathcal{B}(E_t(x), x)$ by relying on the Lipschitzianity of the function $\mu(x)$. Once the arm x_t has been played, the B-Zoom algorithm pays a cost $c_t(x_t)$ and, in the case there is still enough budget remaining $B(t) > 0$, it gains reward $r_t(x_t)$ and updates the necessary statistics $\bar{r}_t(x_t)$, $\bar{c}_t(x_t)$ and $n_t(x_t)$ corresponding to arm x_t , otherwise the algorithm stops. Notice that, in $u_t(x)$, we use $\max\{\lambda, \bar{c}_t(x)\}$ in place of the unbiased estimator $\bar{c}_t(x)$ since for some realizations it could happen that $\bar{c}_t(x) < \lambda$, but we *a priori* know that $\mu_c(x) \geq \lambda$.

The B-Zoom algorithm does not require the setting of any parameter, but it requires information about the Lipschitz constant L' (or equivalently of the constants L_r and L_c related to the average rewards and costs, respectively) and of the minimum average cost λ . Moreover, it requires also a covering oracle. In the case the arm space \mathcal{A} is one dimensional, we can state the following (the complexity in higher dimensional spaces might be higher [8]).

Theorem 1. A covering oracle for the B-Zoom algorithm over $\mathcal{A} \subset \mathbb{R}$ has computational complexity $O(n)$, where $n = |X(i_{ph})|$ and $|\cdot|$ is the cardinality operator.

Proof. Let us suppose that at a given round t we are storing in a list s for each arm $x \in X(i_{ph})$ an interval $[p_t(x), q_t(x)] = [x - E_{t-1}(x), x + E_{t-1}(x)]$ and we ordered them w.r.t. ascending values of $p_t(x)$. At each new round, we have either to insert a new arm or modify the confidence radius of an existing one. In the case we introduce a new arm x_j in the set of active arms $X(i_{ph})$, we need to insert the interval $[p_t(x_j), q_t(x_j)]$ in the ordered list s . This operation requires a computational cost of $O(\log_2(n))$ (binary search). Otherwise, if an existing arm x is selected, we have to delete the old interval $[p_{t-1}(x), q_{t-1}(x)]$ from the list s and insert the new one $[p_t(x), q_t(x)]$, which has a total computational cost of $O(\log_2(n))$.

After that, we need to form the covering set \mathcal{C} . Let us assume that the list of intervals is $s = \{I_1, \dots, I_n\} = \{[p_t(x_1), q_t(x_1)], \dots, [p_t(x_n), q_t(x_n)]\}$ (with $p_t(x_1) \leq \dots \leq p_t(x_n)$). At first, we have $C_1 = I_1$. For each $i \in \{2, \dots, n\}$ we perform $C_i = C_{i-1} \cup I_i$. If C_i is still an interval, i.e., if we have $c_M \geq p_t(x_i)$ with $C_{i-1} = [c_m, c_M]$, we continue the procedure, otherwise we can say that the set \mathcal{A} is not covered by \mathcal{C} . If we reached the n -th interval and $C_n = \mathcal{C} \supseteq \mathcal{A}$, then \mathcal{A} is covered by \mathcal{C} . This procedure consists in a maximum of n interval union operations, whose cost is constant. Thus, the computation of the covering set has requires a computational cost of $O(n)$. By considering an empty list s at the first round and by using an inductive argument, we complete the proof. \square

At each round t , the B-Zoom algorithm has a computational complexity of $O(n)$ with $n \leq t$ due to the complexity of the covering oracle. Moreover, notice that n is upper bounded by $O\left(\frac{\lambda^4}{(\lambda+1)^2} \frac{t}{\ln(t)}\right)$ and therefore in practice $n \ll t$. For comparison, the HOO algorithm [8], in its general formulation requires $O(t)$ at turn t .

3.2 Theoretical analysis

Considering the problem formulation described in Section 2, we can show that:

Theorem 2. The regret $\mathcal{R}(B)$ over a generic BCAB problem of the B-Zoom algorithm is:

$$\mathcal{R}(B) \leq \tilde{C} \cdot (\ln(B))^{\frac{1}{d+2}} \cdot B^{\frac{d+1}{d+2}},$$

where d is the Zooming dimension of the Lipschitz MAB problem (\mathcal{A}, l, μ) , with $l(x, y) = L'|x - y|$, and \tilde{C} is an appropriately defined constant.

Proof. At first, by defining the arm with largest expected reward-to-cost ratio $x^* \in \mathcal{A}$ as:

$$x^* := \arg \max_{x \in \mathcal{A}} \mu(x) = \arg \max_{x \in \mathcal{A}} \frac{\mu_r(x)}{\mu_c(x)},$$

we are able to decompose regret $\mathcal{R}(B)$ defined in Equation (1) into two parts:

$$\mathcal{R}(B) = \underbrace{\mathcal{R}^*(B) - \mathbb{E}_{r,c} \left[\sum_{t=1}^{t_a^*} r_t(x^*) \right]}_{\mathcal{R}_1} +$$

$$+ \underbrace{\mathbb{E}_{r,c} \left[\sum_{t=1}^{t_a^*} r_t(x^*) \right] - \mathbb{E}_{r,c} \left[\sum_{t=1}^{t_a} r_t(x_t) \right]}_{\mathcal{R}_2},$$

where \mathcal{R}_1 is the component considering that the best possible strategy is not the one choosing always x^* until t_a^* (the stopping round of action x^*), and \mathcal{R}_2 is the component considering the loss due to the process of finding the arm x^* .

Regret \mathcal{R}_1 can be bounded by trivially extending the result discussed in [20] for BMAB to the case of BCAB:

Lemma 2. Given any instance of the BCAB problem we have:

$$\mathcal{R}_1 \leq 2\mu(x^*) = 2 \frac{\mu_r(x^*)}{\mu_c(x^*)} \leq \frac{2}{\lambda}.$$

Instead, bounding \mathcal{R}_2 is not trivial. For sake of clarity, we divide the proof into three steps. In the first step, we define an auxiliary Lipschitz CAB problem (\mathcal{A}, l, μ) , i.e., a CAB problem without budget or, equivalently, in which each arm has unitary cost and the budget corresponds to a temporal deadline. We show that the execution of the B-Zoom algorithm up to $t = t_a$ to problem (\mathcal{A}, l, μ) is equivalent to the execution of a modified version of the Zooming algorithm. We use this relation to bound the regret of this problem with $\mathcal{R}_\Delta(t)$ over a generic horizon $t \leq t_a$. In the second step, we show that \mathcal{R}_2 is bounded by $\mathcal{R}_\Delta(t_a)$. In the third step, we derive the relationship between the stopping round t_a and the budget B of a BCAB problem and use it to formulate the bound over \mathcal{R}_2 in terms of the budget B .

Step 1. Since Lemma 1 holds, the instance of the CAB problem (\mathcal{A}, l, μ) , with $l(x, y) = L'|x - y|$, is a Lipschitz MAB problem [13]. The regret of the B-Zoom algorithm executed over the Lipschitz problem (\mathcal{A}, l, μ) at round $t \leq t_a$ is defined as:

$$\begin{aligned} \mathcal{R}_\Delta(t) &:= \sum_{i_{ph}=1}^{\log_2(t)} \sum_{x \in X(i_{ph})} \left(\frac{\mu_r(x^*)}{\mu_c(x^*)} - \frac{\mu_r(x)}{\mu_c(x)} \right) n_t(x) \\ &= \sum_{i_{ph}=1}^{\log_2(t)} \sum_{x \in X(i_{ph})} (\mu(x^*) - \mu(x)) n_t(x). \end{aligned}$$

By verifying that the bounds used in the B-Zoom algorithm satisfy the properties required in Lemma 4.15 in [13] we are able to resort on the regret bound results presented in the same work. More specifically we require that the following two properties are satisfied by the B-Zoom algorithm:

Property 1. Consider an instance of the Lipschitz CAB problem (\mathcal{A}, l, μ) and a generic algorithm \mathfrak{A} considering estimates $\hat{\mu}(x)$ and confidence radius $b_t(x)$ for the arm x in phase i_{ph} . A phase i_{ph} is clean with probability δ if for each t s.t. $2^{i_{ph}} \leq t \leq 2^{i_{ph}+1} - 1$ and for each arm $x \in \mathcal{A}$:

$$|\hat{\mu}(x) - \mu(x)| < r_t(x)$$

holds with probability at least $1 - \delta$.

Property 2. Consider the instance of the Lipschitz CAB problem (\mathcal{A}, l, μ) and a generic algorithm \mathfrak{A} considering estimates $\hat{\mu}(x)$ and confidence radius $b_t(x)$ for the arm x in phase i_{ph} . The radius $b_t(x)$ is (c_0, β) -good if there exist $c_0 > 0$ and $\beta > 0$, at a given phase i_{ph} s.t. for all $x \in X(i_{ph})$ if $\mu(x^*) - \mu(x) < E_t(x)$ then $n_t(x) \leq c_0(\mu(x^*) - \mu(x))^{-\beta} i_{ph}$.

At first, we want to show that both these properties are satisfied by the B-Zoom algorithm when applied to problem (\mathcal{A}, l, μ) .

Lemma 3. *Each phase i_{ph} of the B-Zoom algorithm applied to Lipschitz CAB problem (\mathcal{A}, l, μ) is clean with probability at least $1 - 4^{-i_{ph}}$.*

Proof. The proof will show that the probability of the phase i_{ph} of not being clean is smaller than t^{-4} . Since the B-Zoom algorithm considers estimates $\hat{\mu}(x) := \frac{\bar{r}_t(x)}{\bar{c}_t(x)}$ and radius $b_t(x) := E_t(x)$, we have that:

$$\begin{aligned} & \mathbb{P} \left[\left| \frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \mu(x) \right| > E_t(x) \middle| x = x_j \right] \\ &= \mathbb{P} \left[\left| \frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{\mu_r(x)}{\mu_c(x)} \right| > E_t(x) \middle| x = x_j \right] \\ &= \mathbb{P} \left[\underbrace{\left| \frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{\mu_r(x)}{\mu_c(x)} \right| > E_t(x)}_{e_1} \middle| x = x_j \right] + \\ & \quad + \mathbb{P} \left[\underbrace{\left| \frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{\mu_r(x)}{\mu_c(x)} \right| < -E_t(x)}_{e_2} \middle| x = x_j \right]. \end{aligned}$$

The event e_1 implies that at least one of the following two inequalities holds:

- $\bar{r}_t(x) \geq \mu_r(x) + \varepsilon_t(x)$,
- $\bar{c}_t(x) \leq \mu_c(x) - \varepsilon_t(x)$,

where $\varepsilon_t(x) = \sqrt{\frac{8i_{ph} + \ln 4}{n_t(x)}}$. In fact, if $\bar{r}_t(x) \leq \mu_r(x) + \varepsilon_t(x) \wedge \bar{c}_t(x) \geq \mu_c(x) - \varepsilon_t(x)$ and since $\bar{c}_t(x) \geq \lambda$, $\forall x \in \mathcal{A}$, we have:

$$\begin{aligned} \frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{\mu_r(x)}{\mu_c(x)} &= \frac{\bar{r}_t(x)\mu_c(x) - \mu_r(x)\bar{c}_t(x)}{\bar{c}_t(x)\mu_c(x)} \\ &\pm \frac{\pm \mu_c(x)\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} \frac{[\bar{r}_t(x) - \mu_r(x)]\mu_c(x) + [\mu_c(x) - \bar{c}_t(x)]\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} \\ &\leq \frac{\varepsilon_t(x)\mu_c(x) + \varepsilon_t(x)\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} = \\ &= \frac{\varepsilon_t(x)}{\bar{c}_t(x)} + \frac{\varepsilon_t(x)\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} \leq \frac{\varepsilon_t(x)}{\lambda} + \frac{\varepsilon_t(x)}{\lambda^2} \\ &= \frac{1}{\lambda} \left(1 + \frac{1}{\lambda} \right) \varepsilon_t(x) = E_t(x). \end{aligned}$$

The event e_2 implies that at least one of the following two inequalities holds:

- $\bar{r}_t(x) \leq \mu_r(x) - \varepsilon_t(x)$,
- $\bar{c}_t(x) \geq \mu_c(x) + \varepsilon_t(x)$.

In fact, if $\bar{r}_t(x) \geq \mu_r(x) - \varepsilon_t(x) \wedge \bar{c}_t(x) \leq \mu_c(x) + \varepsilon_t(x)$ we have:

$$\begin{aligned} \frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{\mu_r(x)}{\mu_c(x)} &= \frac{\bar{r}_t(x)\mu_c(x) - \mu_r(x)\bar{c}_t(x)}{\bar{c}_t(x)\mu_c(x)} \\ &\pm \frac{\pm \mu_c(x)\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} \frac{[\bar{r}_t(x) - \mu_r(x)]\mu_c(x) + [\mu_c(x) - \bar{c}_t(x)]\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} \\ &\geq \frac{-\varepsilon_t(x)\mu_c(x) - \varepsilon_t(x)\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} = \end{aligned}$$

$$\begin{aligned} &= -\frac{\varepsilon_t(x)}{\bar{c}_t(x)} - \frac{\varepsilon_t(x)\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} \geq -\frac{\varepsilon_t(x)}{\lambda} - \frac{\varepsilon_t(x)}{\lambda^2} \\ &= -\frac{1}{\lambda} \left(1 + \frac{1}{\lambda} \right) \varepsilon_t(x) = -E_t(x) \end{aligned}$$

Thus, we can write:

$$\begin{aligned} & \mathbb{P} \left[\left| \frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \mu(x) \right| > E_t(x) \middle| x = x_j \right] \leq \\ & \mathbb{P} [\bar{r}_t(x) \geq \mu_r(x) + \varepsilon_t(x)] + \mathbb{P} [\bar{c}_t(x) \leq \mu_c(x) - \varepsilon_t(x)] + \\ & \quad + \mathbb{P} [\bar{r}_t(x) \leq \mu_r(x) - \varepsilon_t(x)] + \mathbb{P} [\bar{c}_t(x) \geq \mu_c(x) + \varepsilon_t(x)]. \end{aligned}$$

We can provide a bound to each single term in the r.h.s. of the previous inequality by means of the Hoeffding's bound:

$$\begin{aligned} & \mathbb{P} \left[\left| \frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{r_t(x)}{c_t(x)} \right| > E_t(x) \middle| x = x_j \right] \\ & \leq \frac{t^{-4}}{4} + \frac{t^{-4}}{4} + \frac{t^{-4}}{4} + \frac{t^{-4}}{4} = t^{-4} \end{aligned}$$

with $\varepsilon_t(x) = \sqrt{\frac{8i_{ph} + \ln 4}{n_t(x)}}$.

Taking the union bound over all the $n_t(x) < t$, integrating over $x_j \in [0, 1]$ and taking the union bound over $i \in [0, t]$ concludes the proof. \square

Lemma 4. *The radius $E_t(x)$ of the B-Zoom algorithm applied to Lipschitz CAB problem (\mathcal{A}, l, μ) is (c_0, β) -good with $c_0 = \frac{10(1+\lambda)^2}{\lambda^4}$ and $\beta = 2$.*

Proof. By using the definition of $E_t(x)$ in the B-Zoom algorithm and by defining $\Delta := \mu(x^*) - \mu(x)$:

$$\begin{aligned} \Delta &< E_t(x) \\ \Delta &< \frac{1}{\lambda} \left(1 + \frac{1}{\lambda} \right) \sqrt{\frac{8i_{ph} + \ln 4}{n_t(x)}} \\ \sqrt{\frac{8i_{ph} + \ln 4}{n_t(x)}} &> \frac{\lambda^2 \Delta}{1 + \lambda} \\ \frac{10i_{ph}}{n_t(x)} &> \frac{\lambda^4 \Delta^2}{(1 + \lambda)^2} \\ n_t(x) &< \frac{10(1 + \lambda)^2}{\lambda^4} \Delta^{-2} i_{ph} \end{aligned}$$

thus, taking $c_0 = \frac{10(1+\lambda)^2}{\lambda^4}$ and $\beta = 2$ concludes the proof. \square

Since both Lemmas 3 and 4 hold, it is possible to use Lemma 4.15 in [13] to bound the regret $\mathcal{R}_\Delta(t)$ of the B-Zoom algorithm applied to the Lipschitz CAB problem (\mathcal{A}, l, μ) .

Theorem 3. *Consider the instance of the Lipschitz MAB problem (\mathcal{A}, l, μ) . Fix any $c > 0$ and let d be the Zooming dimension with multiplier c [13]. The regret $\mathcal{R}_\Delta(t)$ of the B-Zoom algorithm satisfies:*

$$\mathcal{R}_\Delta(t) \leq \bar{C}(\ln(t))^{\frac{1}{d+2}} \cdot t^{\frac{d+1}{d+2}},$$

for any $t > 0$, where \bar{C} is an appropriate constant (depending on c).

Step 2. In what follows, we bound \mathcal{R}_2 in terms of $\mathcal{R}_\Delta(t_a)$. It can be observed that, by considering the arm x_t selected by the B-Zoom algorithm at round t , we have the guarantee that at each round t it holds $\mu(x^*) - \mu(x_t) \leq 3E_t(x_t)$. Notice that this inequality does not represent a bound on the instantaneous regret $\mu_r(x^*) - \mu_r(x_t)$. Indeed, the limit of the difference $\mu_r(x^*) - \mu_r(x_t)$ as $3E_t(x_t)$ goes to zero may be a constant $1 - \lambda$ (e.g., consider the case: $\mu_r(x^*) = 1$, $\mu_c(x^*) = 1$, $\mu_r(x_t) = \lambda - \varepsilon$ and $\mu_c(x_t) = \lambda$ with $\varepsilon \ll \lambda$; we have $\mu(x^*) - \mu(x_t) = \frac{\varepsilon}{\lambda}$ while $\mu_r(x^*) - \mu_r(x_t) = 1 - \lambda + \varepsilon$), and therefore the results described in [13] cannot be directly applied to bound \mathcal{R}_2 . Instead, to bound \mathcal{R}_2 , we restate \mathcal{R}_2 as:

$$\begin{aligned} \mathcal{R}_2 &= \mathbb{E}_{r,c} \left[\sum_{t=1}^{t_a^*} r_t(x^*) \right] - \mathbb{E}_{r,c} \left[\sum_{t=1}^{t_a} r_t(x_t) \right] = \\ &= \mathbb{E}_c \left[\mathbb{E}_r \left[\sum_{t=1}^{t_a^*} r_t(x^*) \right] - \mathbb{E}_r \left[\sum_{t=1}^{t_a} r_t(x_t) \right] \right] = \\ &= \mathbb{E}_c \left[\underbrace{\mu_r(x^*)t_a^* - \sum_{i_{ph}}^{\log_2(t_a)} \sum_{x \in X(i_{ph})} \mu_r(x)n_{i_{ph}}(x)}_{\bar{\mathcal{R}}_2} \right], \end{aligned}$$

where $X(i_{ph})$ is the set of active arms in phase i_{ph} and $n_{i_{ph}}(x)$ is the number of rounds we pull x during phase i_{ph} . Let us consider a generic round $t \leq t_a$, with residual budget $B(t) = B - \bar{B} \geq 0$. With notation overload, we denote by $B(x, i_{ph})$ the amount of budget spent by arm x in phase i_{ph} . Each arm $x \in X(i_{ph})$ spent $B(x, i_{ph}) = \mu_c(x)n_{i_{ph}}(x)$ in the phase i_{ph} (in expectation w.r.t. the reward) and $\sum_{i_{ph}=1}^{\log_2(t)} \sum_{x \in X(i_{ph})} B(x, i_{ph}) = \bar{B}$. Let us define $t^*(x) := \frac{\mu_c(x)n_{i_{ph}}(x)}{\mu_c(x^*)}$ for every $x \in X(i_{ph})$ and for every i_{ph} , i.e., the amount of rounds the arm x^* should be pulled to spend a budget of $B(x, i_{ph}) = \mu_c(x)n_{i_{ph}}(x)$. To bound \mathcal{R}_2 , we need to consider $t = t_a$. It is easy to show that $\sum_{i_{ph}=1}^{\log_2(t_a)} \sum_{x \in X(i_{ph})} t^*(x) = t^*$. Thus, we have:

$$\begin{aligned} \bar{\mathcal{R}}_2 &= \sum_{i_{ph}=1}^{\log_2(t_a)} \sum_{x \in X(i_{ph})} \left(\mu_r(x^*)t^*(x) - \mu_r(x)n_{i_{ph}}(x) \right) \\ &= \sum_{i_{ph}=1}^{\log_2(t_a)} \sum_{x \in X(i_{ph})} \left(\mu_r(x^*) \frac{\mu_c(x)}{\mu_c(x^*)} n_{i_{ph}}(x) - \mu_r(x) \frac{\mu_c(x)}{\mu_c(x)} n_{i_{ph}}(x) \right) \\ &= \sum_{i_{ph}=1}^{\log_2(t_a)} \sum_{x \in X(i_{ph})} \left(\frac{\mu_r(x^*)}{\mu_c(x^*)} - \frac{\mu_r(x)}{\mu_c(x)} \right) n_{i_{ph}}(x) \mu_c(x) \\ &\leq \sum_{i_{ph}=1}^{\log_2(t_a)} \sum_{x \in X(i_{ph})} \left(\frac{\mu_r(x^*)}{\mu_c(x^*)} - \frac{\mu_r(x)}{\mu_c(x)} \right) n_{i_{ph}}(x) = \mathcal{R}_\Delta(t_a), \end{aligned}$$

where the last inequality holds since $\mu_c(x) \leq 1$ for every $x \in \mathcal{A}$.

Step 3. In this step we formulate the regret for the Lipschitz CAB problem (\mathcal{A}, l, μ) , previously defined on the basis of t_a , as depending on B . Initially, we state:

Lemma 5. For each $\delta \in (0, 1)$, with probability at least $1 - \delta$, the number of rounds t used to spend a budget of \bar{B} is:

$$t \leq \frac{\bar{B}}{\lambda} + \frac{2(1-\lambda)}{\lambda\sqrt{\lambda}} \sqrt{\bar{B} \ln(1/\delta)} + \left(\frac{1-\lambda}{\lambda} \right)^2 \ln(1/\delta).$$

Proof. Consider the unbiased estimator of the average cost

$\frac{\sum_{i=1}^t c_i(x_i)}{t}$ until round t , we have:

$$\begin{aligned} &\mathbb{P} \left(\frac{\sum_{i=1}^t c_i(x_i)}{t} \leq \lambda - \varepsilon \right) \\ &\leq \mathbb{P} \left(\frac{\sum_{i=1}^t c_i(x_i)}{t} \leq \frac{\sum_{i=1}^t \mu_c(x_i)}{t} - \varepsilon \right) \end{aligned}$$

since $\mu_c(x) > \lambda$ for every $x \in \mathcal{A}$. Thus, we can bound the r.h.s. of the previous equation by using the Hoeffding's bound:

$$\mathbb{P} \left(\frac{\sum_{i=1}^t c_i(x_i)}{t} \leq \lambda - \varepsilon \right) \leq \delta \rightarrow \varepsilon = \sqrt{\frac{\ln(1/\delta)(1-\lambda)^2}{2t}}$$

At round t and with probability at least $1 - \delta$ the budget spent is:

$$\begin{aligned} \bar{B} &\geq t \left(\lambda - \sqrt{\frac{\ln(1/\delta)(1-\lambda)^2}{2t}} \right) \\ 2\lambda t - \sqrt{2 \ln(1/\delta)(1-\lambda)^2} t^{1/2} - 2\bar{B} &\leq 0 \\ t^{1/2} &\leq \frac{\sqrt{2 \ln(1/\delta)(1-\lambda)^2} + \sqrt{2 \ln(1/\delta)(1-\lambda)^2 + 16\lambda\bar{B}}}{4\lambda} \\ t^{1/2} &\leq \frac{\sqrt{2 \ln(1/\delta)(1-\lambda)^2} + \sqrt{2 \ln(1/\delta)(1-\lambda)^2} + \sqrt{16\lambda\bar{B}}}{4\lambda} \\ t &\leq \left(\frac{4(1-\lambda)\sqrt{\ln(1/\delta)} + 4\sqrt{\lambda\bar{B}}}{4\lambda} \right)^2 \\ t &\leq \left(\sqrt{\frac{\bar{B}}{\lambda}} + \frac{(1-\lambda)\sqrt{\ln(1/\delta)}}{\lambda} \right)^2 \\ t &\leq \frac{\bar{B}}{\lambda} + \frac{2(1-\lambda)}{\lambda\sqrt{\lambda}} \sqrt{\bar{B} \ln(1/\delta)} + \left(\frac{1-\lambda}{\lambda} \right)^2 \ln(1/\delta) \end{aligned}$$

which concludes the proof. \square

Since the bound in Lemma 5 holds also for the stopping round t_a (when the budget spent is B), by considering $\delta = B^{-\frac{1}{d+2}}$ and $\ln(B) \leq B$ we have:

$$\begin{aligned} t_a &\leq \frac{B}{\lambda} + \frac{2(1-\lambda)}{\lambda\sqrt{\lambda}} \sqrt{B \ln(1/\delta)} + \left(\frac{1-\lambda}{\lambda} \right)^2 \ln(1/\delta) \\ t_a &\leq \frac{B}{\lambda} + \frac{2(1-\lambda)}{\lambda\sqrt{\lambda}\sqrt{d+2}} B + \left(\frac{1-\lambda}{\lambda} \right)^2 \frac{B}{d+2} \\ t_a &\leq \frac{6}{\lambda^2(d+2)} B \end{aligned}$$

Finally, by taking the expectation over time (or over costs equivalently), we have that there exists constant \tilde{C} (depending on λ, d and c) s.t.:

$$\begin{aligned} \mathcal{R}(B) &\leq \mathcal{R}_1 + (1-\delta)\bar{\mathcal{R}}_2 + \delta B \\ &\leq \mathcal{R}_1 + (1-\delta)\mathcal{R}_\Delta(t_a) + \delta B \\ &\leq \frac{2}{\lambda} + (1-\delta)\tilde{C}(\ln(t_a))^{\frac{1}{d+2}} \cdot t_a^{\frac{d+1}{d+2}} + \delta B \\ &\leq \frac{2}{\lambda} + \tilde{C}(\ln(B))^{\frac{1}{d+2}} \cdot B^{\frac{d+1}{d+2}} + \delta B \\ &\leq \tilde{C}(\ln(B))^{\frac{1}{d+2}} \cdot B^{\frac{d+1}{d+2}} \end{aligned}$$

which concludes the proof. \square

4 Experimental analysis

In this section, we evaluate the empirical performance of the B-Zoom algorithm. Our evaluation is twofold and it is based on the comparison with other frequentist algorithms, ours being of this class. In particular, we compare the performance of the B-Zoom algorithm (denoted BZ for short) w.r.t. the one of the Zooming algorithm [12] (denoted Z for short) suited for CAB problems, analyzing empirically the impact of taking into account explicitly information about budget and costs. We recall indeed that, in the worst-case analysis, the Zooming algorithm performs arbitrarily worse than the B-Zoom one if a budget constraint is present, since it is assured to find the arm maximizing the expected reward, which in general is different from the optimal reward-to-cost ratio optimal arm x^* . Furthermore, we empirically analyze the impact of exploiting information about the continuous structure of the arm space by comparing the performance of the B-Zoom algorithm w.r.t. the UCBBV1 algorithm [10], designed for BMAB problems, and the UCB1 [3] one, designed for MAB problems, both applied to a finite set of arms obtained by some discretization of the arm space \mathcal{A} and kept fixed for all the rounds. We recall that in the worst-case analysis the UCBBV1 and UCB1 algorithms applied to a finite set of arms randomly drawn from the arm space perform arbitrarily worse than the B-Zoom algorithm when expected reward and costs are Lipschitz. Indeed, consider the case the reward-to-cost ratio $\mu(x)$ is flat except for a small-supported peak in which there is the optimum. The UCBBV1 algorithm might perform as good as a random choice when the finite set of arms is such that all the arms are positioned in the flat part of $\mu(x)$.

In what follows, we consider the *cumulative profit* of a policy \mathfrak{A} over a fixed budget B as figure of merit, defined as:

$$P_{\mathfrak{A}}(B) = \sum_{t=1}^{t_a} r_t(x_t).$$

Experimental setting To provide a thorough experimental evaluation of each algorithm, we consider different settings with budget $B \in \{50,000; 100,000; 500,000\}$ and with $\lambda \in \{0.05; 0.10; 0.25; 0.50\}$. We select the average reward-to-cost ratio functions $\mu(x)$ s.t. the average reward $\mu_r(x)$ and cost $\mu_c(x)$ are:

$$\begin{aligned} \mu_r(x) &= 0.05x + 0.95, \\ \mu_c(x) &= \lambda + \frac{1}{5} \left(1 - e^{-500(x-\tilde{x})^2} \right), \end{aligned}$$

where \tilde{x} is the arm with the lowest expected cost. The value for the arm \tilde{x} is sampled for each of the experiments from a uniform distribution over $[0, 1]$. The instantaneous reward $r_t(x)$ and cost $c_t(x)$ for pulling an arm x are sampled from Bernoulli distributions, i.e., $R_t(x) \sim Be(\mu_r(x))$ and $C_t(x) \sim Be(\mu_c(x))$, respectively. The above functions are modeling a setting where the reward is linearly increasing in the value $x \in \mathcal{A}$ of the arms and the cost is constant except for a small area around \tilde{x} , where it is sensibly lower, which generates a unimodal reward-to-cost ratio function. For each pair of values of B and λ , we generate 100 different instances characterized by potentially different reward-to-cost ratio functions. Given the empirical profit obtained by an algorithm over the 100 instances, we compute the empirical mean \bar{P} , the minimum m , the 25-th percentile Q_1 (first quartile) and the 50-th percentile Q_2 (second quartile or median value).

For the UCBBV1 and UCB1 algorithms we use different numbers of arms $K \in \{5, 10, 15\}$ randomly placed over the arm space \mathcal{A} . For sake of comparison, we here adopt a version of the B-Zoom

Table 1 Results for the cumulative profit provided in thousand of reward units. The highest cumulative profit for each row is highlighted in bold.

K		P_{BZ}	P_Z	P_{UCBBV1}			P_{UCB1}		
		-	-	5	10	15	5	10	15
$\lambda = 0.05$	m	203	199	198	199	199	198	199	198
	Q_1	212	200	200	203	204	201	207	203
	Q_2	215	201	204	215	213	206	216	209
	\bar{P}	217	212	266	236	216	215	216	211
$\lambda = 0.1$	m	170	165	165	165	166	165	166	166
	Q_1	175	166	167	168	172	167	168	169
	Q_2	180	167	173	184	192	171	174	174
	\bar{P}	181	173	230	224	219	178	176	176
$\lambda = 0.25$	m	113	110	110	111	110	110	110	111
	Q_1	117	111	111	112	118	111	112	112
	Q_2	119	111	114	129	144	112	114	115
	\bar{P}	119	115	134	139	142	115	115	115
$\lambda = 0.5$	m	72	71	71	71	71	71	71	71
	Q_1	73	71	71	72	74	71	72	72
	Q_2	74	71	72	77	85	72	73	73
	\bar{P}	74	74	78	80	83	73	73	73
$B = 50,000$									
$\lambda = 0.05$	m	413	397	398	398	400	397	397	398
	Q_1	428	400	400	405	414	400	410	411
	Q_2	442	401	408	429	459	413	432	423
	\bar{P}	442	426	542	554	520	427	436	426
$\lambda = 0.1$	m	347	331	330	332	332	332	332	332
	Q_1	362	333	334	336	342	334	337	341
	Q_2	372	334	345	386	430	341	353	348
	\bar{P}	374	351	463	476	494	353	353	352
$\lambda = 0.25$	m	232	221	221	221	221	221	222	221
	Q_1	242	222	223	224	236	223	225	226
	Q_2	246	222	231	263	311	225	231	230
	\bar{P}	247	230	275	284	301	230	231	231
$\lambda = 0.5$	m	145	142	142	143	143	142	142	142
	Q_1	150	143	143	143	147	143	143	144
	Q_2	152	143	144	161	171	144	146	145
	\bar{P}	152	147	156	164	168	146	146	146
$B = 100,000$									
$\lambda = 0.05$	m	2129	1995	1995	1997	1998	1996	1994	1995
	Q_1	2249	2001	2002	2037	2083	2001	2042	2060
	Q_2	2368	2005	2050	2999	2654	2088	2155	2123
	\bar{P}	2376	2111	3383	4372	4022	2171	2162	2146
$\lambda = 0.1$	m	1923	1661	1662	1664	1666	1662	1664	1664
	Q_1	2120	1666	1667	1682	1788	1667	1687	1707
	Q_2	2308	1669	1700	2201	3076	1675	1757	1769
	\bar{P}	2267	1725	2457	2758	3082	1756	1769	1768
$\lambda = 0.25$	m	1360	1107	1109	1110	1109	1108	1110	1109
	Q_1	1432	1111	1111	1187	1248	1112	1115	1124
	Q_2	1471	1112	1142	1599	1743	1123	1145	1142
	\bar{P}	1461	1142	1343	1554	1608	1153	1152	1149
$\lambda = 0.5$	m	784	712	713	713	714	713	713	712
	Q_1	807	714	715	718	800	715	716	719
	Q_2	821	715	730	874	928	722	729	729
	\bar{P}	816	724	808	845	890	735	730	731
$B = 500,000$									

and Zooming algorithms that do not consider exponentially long phases i_{ph} , but we use a single phase and a confidence radius equal to $E_t(x) = \frac{1}{\lambda} \left(1 + \frac{1}{\lambda} \right) \sqrt{\frac{8(\ln B - \ln \lambda) + \ln 4}{n_t(x)}}$, where the term $\frac{B}{\lambda}$ is a rough estimation of the average stopping time of the learning process. The experiments here reported have been performed in MATLAB.

Results We report in Table 1 the results of our experiments. Initially, we focus on the empirical mean \bar{P} . The B-Zoom algorithm outperforms the Zooming one for all the configurations, providing a larger profit up to about 30%. This was expected since the B-Zoom algorithm exploits more information than the Zooming algorithm. Similar results can be observed when compared with the UCB1. Unexpectedly, the UCBBV1 applied to a randomly gener-

ated set of arms outperforms the B-Zoom algorithm for all the configurations, providing a larger profit up to about 40%. In order to understand the reasons behind such a behavior, we need to focus on the other indices: m , Q_1 , and Q_2 . The B-Zoom algorithm in most of the cases outperforms the other algorithms for the indices m and Q_1 . More specifically, with $B = 50,000$ all the minimum values m provided by the B-Zoom algorithm are higher than the one achieved by other algorithms and the difference in terms of cumulative profit increases as the minimum average cost λ decreases. In this setting, when $\lambda = 0.05$ and $\lambda = 0.10$ even the first quartile Q_1 has higher values. Conversely, in terms of median (Q_2), the UCBBV1 algorithm with $K = 15$ arms provides the largest profit in most of the settings, despite being less robust to unfavorable cases. Similar results also hold for the settings with $B = 100,000$ and $B = 500,000$. These results suggest that the B-Zoom algorithm is the most robust algorithm in the worst case, assuring the best performance for m and Q_1 , but, in order to be robust, it must pay a cost in the average case (and in some situations in the median case). This suggests also that, in our experiments, the worst case occurs with low probability, otherwise the B-Zoom algorithm would provide good performance also for the empirical mean \bar{P} . This is clear by observing Figure 1, where we report the boxplot for the case with $B = 100,000$ and $\lambda = 0.10$: the B-Zoom algorithm provides the best performance at m and Q_1 , but it also presents a compact distribution with an extremely low variance. Instead, the performance of the UCBBV1 algorithm presents a very large variance that allows it to have both poor and excellent profits.

By analyzing how different values of initial budgets B affect the performance of the algorithms, we can observe how the B-Zoom algorithm is able to improve the minimum cumulative profit from approximately 1% in the case $B = 50,000$ scenario to more than 10% in the case $B = 500,000$. This behavior was expected since we have assurance of convergence to the optimal solution for the B-Zoom algorithm, while an algorithm relying on a fixed discretization of the space or considering a different minimization objective function (loss of cumulative reward) might not converge to the optimal arm.

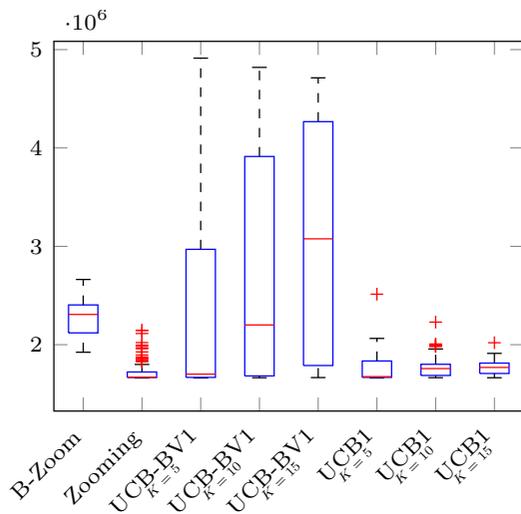


Figure 1. Boxplots of $P_1(500,000)$ for different algorithms with $\lambda = 0.1$

Summarily, the theoretical guarantee over the regret minimization of the B-Zoom algorithm represents an intrinsic limit to outperform

Table 2 Results for the cumulative profit with fixed cost $\lambda = 0.5$, provided in thousands of reward units. The highest cumulative profit for each budget and row is highlighted in bold.

K	P_Z				P_{UCB1}				P_Z				P_{UCB1}			
	-	5	10	15	-	5	10	15	-	5	10	15	-	5	10	15
m	0.7	0.0	0.0	0.0	2.6	0.0	0.0	0.0	43.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Q_1	1.4	0.0	0.0	1.5	4.0	0.0	0.4	2.4	48.6	0.0	8.0	27.2	0.0	0.0	0.0	0.0
Q_2	2.0	0.1	2.9	5.7	4.7	0.3	5.7	11.7	53.2	14.7	46.3	70.7	0.0	0.0	0.0	0.0
\bar{P}	1.9	2.2	3.4	4.7	5.0	4.6	7.2	9.3	51.9	32.3	48.6	60.1	0.0	0.0	0.0	0.0
	B = 50,000				B = 100,000				B = 500,000							

other algorithms that do not have any theoretical guarantee in terms of mean empiric profit. However, the B-Zoom algorithm is more robust w.r.t. the other algorithms in terms of the minimum m and first quartile Q_1 of cumulative profit. In principle, this makes the B-Zoom algorithm more suitable for situations in which the learner is risk averse.

Additional results On the basis of the results described above, we investigate whether the poor performance in terms of empiric mean profit of the B-Zoom algorithm is intrinsic in the need for being robust to the worst case with continuous arm space independently of the presence of budget constraints or it is due exclusively to the presence of budget constraints in continuous arm space. To evaluate this issue, we compare the performance of an algorithm suited for the CAB case, i.e., the Zooming algorithm, versus the one provided by a discrete MAB, i.e., the UCB1, once a random discretization of the space is applied reducing the budget constraint to a time horizon constraint. We consider a setting with fixed costs for all the arms $C_t(x) = \lambda, \forall x, t$ and rewards $R_t(x)$ drawn from Bernoulli distributions with expected value $\mu_r(x) = \frac{1}{5}e^{-500(x-\tilde{x})^2}$ and \tilde{x} is uniformly drawn from $[0, 1]$, where the optimal arm \tilde{x} for the reward-to-cost ratio function coincides with the optimal arm for the reward function. In this way, the BCAB problem reduces to a CAB problem with $T = B/\lambda$. We repeat the experiments for 100 independent runs.

We report our experimental results in Table 2. Even in these experiments the continuous approach is able to provide a risk-averse alternative to the discretized ones, at the expense of loss in terms of average performance. Again, the values for the minimum m is always higher and the first quartile Q_1 is higher in the case we have a larger budget. This behaviour is explained by the fact that the Zooming algorithm always adds arms over the whole space to cope with possible worst-case settings, which decreases its average performance. The loss due to the introduction of such arms is balanced by the convergence to the optimal arm, which asymptotically provides higher profits and at the same time is able not to reduce the losses in unfavorable settings.

5 Conclusions and future works

In this paper, we present a new problem, the Budgeted Continuous-Armed Bandit (BCAB), and an algorithm, the B-Zoom, specifically suited for this setting. We study the proposed algorithm both in terms of theoretical properties and empirical performances. While it suffers a regret of $\tilde{O}(B^{\frac{d+1}{d+2}})$, it is able to provide empirical evidence that it is more risk averse than the algorithms present in the literature of BMAB.

Some of the most promising works for future research are: introducing a vector of costs and a stopping round dependent on a combination of these costs. Moreover, we may explore the problem of having a search space \mathcal{A} with more than one dimension. Finally, it could be interesting to extend other existing algorithms for the CAB setting to the BCAB problem.

REFERENCES

- [1] András Antos, Varun Grover, and Csaba Szepesvári, ‘Active learning in multi-armed bandits’, in *Proceedings of the International Conference on Algorithmic Learning Theory, ALT*, pp. 287–302. Springer, (2008).
- [2] Danilo Ardagna, Barbara Panicucci, and Mauro Passacantando, ‘A game theoretic formulation of the service provisioning problem in cloud systems’, in *Proceedings of the International Conference on World Wide Web, WWW*, pp. 177–186, (2011).
- [3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer, ‘Finite-time analysis of the multiarmed bandit problem’, *Machine learning*, **47**(2-3), 235–256, (2002).
- [4] Peter Auer, Ronald Ortner, and Csaba Szepesvári, ‘Improved rates for the stochastic continuum-armed bandit problem’, in *Proceedings of the Annual Conference on Learning Theory, COLT*, pp. 454–468, (2007).
- [5] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins, ‘Bandits with knapsacks’, in *Proceedings of the Annual Symposium on Foundations of Computer Science, FOCS*, pp. 207–216, (2013).
- [6] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Kamal Jain, Omid Etesami, and Mohammad Mahdian, ‘Dynamics of bid optimization in online advertisement auctions’, in *Proceedings of the International Conference on World Wide Web, WWW*, pp. 531–540, (2007).
- [7] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz, ‘Pure exploration in multi-armed bandits problems’, in *Proceedings of the International Conference on Algorithmic Learning Theory, ALT*, pp. 23–37, (2009).
- [8] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári, ‘X-armed bandits’, *The Journal of Machine Learning Research*, **12**, 1655–1695, (2011).
- [9] Sébastien Bubeck, Gilles Stoltz, and Jia Yuan Yu, ‘Lipschitz bandits without the lipschitz constant’, in *Proceedings of the International Conference on Algorithmic Learning Theory, ALT*, pp. 144–158, (2011).
- [10] Wenkui Ding, Tao Qin, Xu-Dong Zhang, and Tie-Yan Liu, ‘Multi-armed bandit with budget constraint and variable costs’, in *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI*, pp. 232–238, (2013).
- [11] Sudipto Guha and Kamesh Munagala, ‘Approximation algorithms for budgeted learning problems’, in *Proceedings of the Symposium on Theory of Computing, STOC*, pp. 104–113. ACM, (2007).
- [12] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal, ‘Multi-armed bandits in metric spaces’, in *Proceedings of the Symposium on Theory of Computing, STOC*, pp. 681–690, (2008).
- [13] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal, ‘Bandits and experts in metric spaces’, *arXiv preprint arXiv:1312.1277*, (2013).
- [14] Robert D. Kleinberg, ‘Nearly tight bounds for the continuum-armed bandit problem’, in *Proceedings of Neural Information Processing Systems, NIPS*, pp. 697–704, (2004).
- [15] Stefan Magureanu, Richard Combes, and Alexandre Proutiere, ‘Lipschitz bandits: Regret lower bound and optimal algorithms’, in *Proceedings of the Conference on Learning Theory, COLT*, pp. 975–999, (2014).
- [16] Paritosh Padhy, Rajdeep K Dash, Kirk Martinez, and Nicholas R Jennings, ‘A utility-based adaptive sensing and multihop communication protocol for wireless sensor networks’, *Transactions on Sensor Networks*, **6**(3), 27:1–27:39, (2010).
- [17] Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas R Jennings, ‘Knapsack based optimal policies for budget-limited multi-armed bandits’, in *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI*, pp. 1134–1140, (2012).
- [18] Long Tran-Thanh, Alex Rogers, and Nicholas R Jennings, ‘Long-term information collection with energy harvesting wireless sensors: a multi-armed bandit based approach’, in *Proceedings of the Autonomous Agents and Multi-Agent Systems, AAMAS*, volume 25, pp. 352–394, (2012).
- [19] Yingce Xia, Wenkui Ding, Xu-Dong Zhang, Nenghai Yu, and Tao Qin, ‘Budgeted bandit problems with continuous random costs’, in *Proceedings of the Asian Conference on Machine Learning, ACML*, pp. 317–332, (2015).
- [20] Yingce Xia, Haifang Li, Tao Qin, Nenghai Yu, and Tie-Yan Liu, ‘Thompson sampling for budgeted multi-armed bandits’, in *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*, pp. 3960–3966, (2015).

A Framework for Automatic Debugging of Functional and Degradation Failures

Nuno Cardoso¹ and Rui Abreu² and Alexander Feldman³ and Johan de Kleer⁴

Abstract. Software diagnosis is a particularly challenging problem for modern systems, which may consist of dozens, if not hundreds, of components computing on concurrent and potentially distributed platforms, and using infrastructure and services built by many organizations. We propose a framework that generalizes state-of-the-art classical reasoning-based fault diagnosis which tolerates observation uncertainty and addresses degradation of quality of service. Empirical evaluation involving 27 000 highly realistic synthetic scenarios demonstrates an average accuracy improvement of 20% (with 99% statistical significance) which is considerable in the domain of Software Fault Localization (SFL). We measure the improvement in accuracy on well-established SFL performance metrics.

Introduction

One of the most important way to improve the trustworthiness of software systems is to increase their robustness in the face of (run-time) failures. While design-time methods are useful in improving confidence in software (e.g., [3, 13, 16, 19, 22, 31]), they cannot by themselves eliminate the possibility of run-time failures, which are induced by a variety of factors largely outside the control of the organization producing that software: faults in runtime infrastructure and components provided by third-parties, unpredictable loads, variable resources, and malicious attempts to break a system. Moreover, as mentioned in [12], the distinction between “healthy” and “broken” is often indistinct and fuzzy, and there is a gradual transition, over time, between these two states [12]. Consequently, stakeholders must take increasing responsibility for improving the trustworthiness of their systems through building automatic runtime problem detection and repair [12, 23].

Diagnosis for today’s complex systems, however, is particularly challenging. First, the presence of concurrency makes it difficult to identify which computation might have caused a problem. Second, reliance on middleware for distributed communication, and more generally the use of components and infrastructure produced by many organizations, means that in many cases neither specifications nor code is available for all parts of the system. Third, in many systems, problems may be intermittent, caused by transient faults or variability in loads. Fourth, many of the “faults” that we care about are reflected indirectly by violation of a systems quality of service goals, such as degradation of response latency, rather than by a direct failure such as a server or system crash. Such “soft” faults may

be difficult to detect and diagnose [12]. Consequently, although fault diagnosis has been studied extensively for both hardware and software systems as a development time activity, the ability to do this at run-time (i.e., while the system is operational) in a systematic way for complex systems has remained an elusive goal [8].

As no behavioral models are typically available, current approaches to software diagnosis abstract the system under analysis in terms of component activity and correct/incorrect behavior, notably lacking mechanisms to encode soft faults.

We propose a framework that generalizes state-of-the-art classical reasoning-based fault diagnosis (such as, Spectrum-based Fault Localization (SFL) [4], GDE [19]) to accommodate functional and degradation failures. In particular, the framework is capable of reasoning under uncertainty (there is a variety of sources of uncertainty as the ability to observe the behavior of a system may be limited by the kinds of monitoring infrastructure available) and handle soft faults. In many cases the existence of a fault is linked to degradation of quality of service. For example, high latencies of responses to queries may indicate that servers are overloaded, that a network connection is faulty, or both.

Our framework improves the classical reasoning approach in 65% of the cases and achieved at least equal performance in 94% of the cases. The overall relative improvement in the diagnostic quality was of 20% on average, with a 99% confidence interval.

This paper makes the following contributions:

- We discuss the limitations imposed by the classical reasoning-based fault diagnosis;
- We propose a generalization of the classical reasoning-based diagnostic framework aimed at improving its accuracy when diagnosing soft faults;
- We compare the accuracies of the classical and our novel approach using a simulation-based setup, which has been shown to be able to generate realistic scenarios [9].

Reasoning-Based Diagnosis

In this section we introduce concepts and definitions used throughout the paper, as well as the reasoning-based SFL approach to diagnosis.

Definition 1 (Diagnostic System). *A diagnostic system DS is a set of components $COMPS = \{c_1, c_2, \dots, c_m\}$.*

The type of systems we consider typically consists of hundreds to thousands of components. These components can be code-blocks, assembly-level instructions or whole state machines. The systems can be distributed, hybrid, and can contain network components such as routers and balancers.

¹ University of Porto and HASLab / INESC TEC, Portugal. email: nunopcardoso@gmail.com

² Palo Alto Research Center, Inc, USA. email: rui@parc.com

³ Palo Alto Research Center, Inc, USA. email: afeldman@parc.com

⁴ Palo Alto Research Center, Inc, USA. email: dekleer@parc.com

Definition 2 (Transaction). A transaction $\langle A, e \rangle$ is a pair containing $A \subseteq \text{COMPS}$ and the transaction outcome $e \in \{0, 1\}$.

Transactions are typically computed by executing programs or program tasks and recording success or failure e . The program components that participate in A are instrumented by using debugger-like methods [14]. The convention is that $e = 1$ means failure and $e = 0$ means success. Transactions where $e = 1$ are also known as conflicts [19]. A conflict represents a set of components that cannot be simultaneously healthy to explain the observed erroneous behavior.

Definition 3 (Hit Spectrum). A hit spectrum \mathbf{A} is a set of transactions $\mathbf{A} = \{\langle A_1, e_1 \rangle, \langle A_2, e_2 \rangle, \dots, \langle A_n, e_n \rangle\}$.

We assume that an external diagnostic engine [21, 28, 11, 1, 5] computes a set of diagnoses. These diagnoses are used as an input to our algorithm.

Definition 4 (Diagnosis). A diagnosis $\langle d, Pr(d) \rangle$ is defined by a set of components $d \subseteq \text{COMPS}$ and a prior probability $Pr(d)$.

The prior probability $Pr(d)$ estimates to what extent a candidate, without further evidence, is responsible for the system's malfunction. To define $Pr(d)$, let p_j denote the prior probability that a component c_j is at fault. The value of p_j is application dependent. In the context of development-time fault localization, p_j is often approximated as $p_j = 1/1000$, i.e., 1 fault for each 1000 lines of code [7].

Assuming that components fail independently, the prior probability for a particular diagnosis d is given by

$$Pr(d) = \prod_{j \in d} p_j \cdot \prod_{j \in \text{COMPS} \setminus d} (1 - p_j) \quad (1)$$

When the p_j are equal the larger the candidate the smaller its *a priori* probability will be.

This leads us to our main goal which is to apply a Bayes conditioning rule.

Definition 5 (Bayes Conditioning). Given a diagnosis d and a hit spectrum \mathbf{A} , the *a posteriori* probability $Pr(d|\mathbf{A})$ is:

$$Pr(d|\mathbf{A}) = Pr(d) \cdot \prod_{i=1,2,\dots,N} \frac{Pr(A_i, e_i | d)}{Pr(A_i)} \quad (2)$$

In order to characterize the optimality of our algorithm we need to compute posteriori probability (given \mathbf{A}) for a whole set of diagnoses and to rank them. This gives us our main computational problem:

Problem 1 (Diagnostic Ordering). Given a set of diagnoses $D = \{\langle d_k, Pr(d_k) \rangle\}$ for $k \in \{1, 2, \dots, K\}$ and a hit spectrum \mathbf{A} , compute $Pr(d_k|\mathbf{A})$ and order D such that:

$$\forall d_k \in D : Pr(d_k|\mathbf{A}) \geq Pr(d_{k+1}|\mathbf{A}) \quad (3)$$

In what follows we describe the relevant aspects of the classical reasoning-based SFL approach to address the ranking problem [3, 19].

To simplify computation we assume conditional independence throughout the process, i.e., our Bayes classifier is naïve.

The denominator $Pr(A_i)$ is a normalizing term that is identical for all $d \in D$ and needs not to be calculated for ranking purposes as it does not alter the rank order.

To bias the prior probability taking run-time information (i.e., observations) into account, $Pr(A_i, e_i | d)$ (referred to as likelihood) is defined as

$$Pr(A_i, e_i | d) = \begin{cases} G(d, A_i) & \text{if } e_i = 0 \\ 1 - G(d, A_i) & \text{otherwise} \end{cases} \quad (4)$$

$G(d, A_i)$ (referred to as transaction goodness) is used to account for the fact that components may fail intermittently, estimating the probability of nominal system behavior under an activation pattern A_i and a diagnostic candidate d .

Let g_j (referred to as component goodness) denote the probability that a component c_j performs nominally. Considering that all components must perform nominally to observe a nominal system behavior, $G(d, A_i)$ is defined as

$$G(d, A_i) = \prod_{j \in (d \cap A_i)} g_j \quad (5)$$

In scenarios where the values for g_j are not otherwise available, those values can be estimated by maximizing $Pr(A, e | d)$ (Maximum Likelihood Estimation (MLE) for naïve Bayes classifier) under parameters $\{g_j | j \in d \wedge 0 \leq g_j \leq 1\}$ [3].

Approach

In this section we discuss how degradation failures can be more accurately detected/represented and how the diagnostic framework presented in the previous section can be enhanced to more accurately diagnose such kind of errors.

Fuzzy Error Detection

The first challenge in diagnosing soft failures related to their detection. Existent approaches to error detection (e.g., [8], SFL [4], and GDE [19]) make use of first-order logic descriptions of the correct behavior of the system (weak-fault models) to assign transactions to one of two possible sets: the pass set and the fail set (P and F respectively, where $F = \overline{P}$). A consequence of such fault models is the *crisp* distinction between correct and incorrect system states. While this crisp logic description enables an accurate representation of hard failures, it is unable to accurately represent a large variety of soft failures. Take for instance a type of soft failure that, informally, can be described by the statement "The system is slow." Even though we can easily relate the slowness of the system to an appropriate metric (e.g., response time), it is not easy to define a crisp boundary in this same metric to distinguish acceptable and slow transactions. By setting a crisp boundary at, for instance, 1 second, a response time of 0.9999 seconds would be considered to be correct whereas a marginally superior response time would be considered incorrect. Also, a response time of 0.9999 seconds would result in the same type of error information (pass) as a smaller response time even though the larger response time may represent an error symptom.

To overcome the expressiveness limitation of crisp logic error detection mechanisms, we propose a generalization using fuzzy logic [29]. Fuzzy logic extends the notion of binary set membership by introducing the concept of membership functions, denoted μ_ω (membership function for set ω – in the context of this paper $\omega \in \{F, P\}$), that map a particular domain on the real continuous interval $[0, 1]$, where the endpoints of 0 and 1 conform to no membership and full membership, respectively. Let x be an arbitrary event. In the context of error detection, the concept of fuzzy membership enables the representation of 3 types of system states:

correct: $\mu_F(x) = 0$,
incorrect: $\mu_F(x) = 1$, and
degraded: $0 < \mu_F(x) < 1$

As a consequence of the new fuzzy error model, a degraded transaction exhibits both correct and incorrect behaviors simultaneously, however with different degrees. As $F = \bar{P}$, it follows that

$$\mu_P(x) = 1 - \mu_F(x) \quad (6)$$

As an example, consider the crisp fail set containing all response times (rt) above 1 second. This same set could be represented in terms of a membership function as

$$e_{\text{crisp}}(rt) = \mu_F(rt) = \begin{cases} 0 & , rt \leq 1 \\ 1 & , rt > 1 \end{cases} \quad (7)$$

To achieve the goal of representing soft failures (and implicitly the degraded state), consider that all response times below 0.5 seconds could be considered correct and all times above 1 second incorrect. Furthermore, consider that the amount of degradation follows a linear pattern between those two thresholds. The fuzzy fail set representing this particular type of error could be defined as

$$e_{\text{fuzzy}}(rt) = \mu_F(rt) = \begin{cases} 0 & , rt < 0.5 \\ 2 \cdot rt - 1 & , 0.5 \leq rt \leq 1 \\ 1 & , rt > 1 \end{cases} \quad (8)$$

Both membership functions are presented in Figure 1. It is important to point out that the assumption of linear degradation introduced in Equation 8 was only for simplicity. In real-world scenarios, the membership functions are application dependent and can exhibit arbitrary patterns. We treat the membership functions as black-boxes. Figure 2 shows a set of alternative membership functions for the error previously described. Despite their odd shapes they are acceptable membership functions, provided that they correctly describe the error state of the transaction.

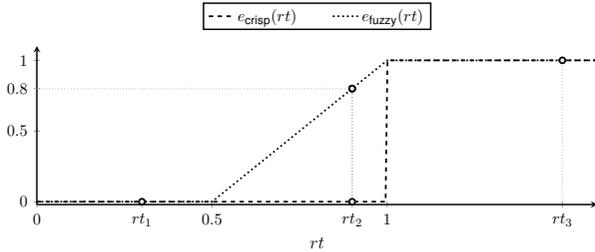


Figure 1: Crisp vs. fuzzy sets

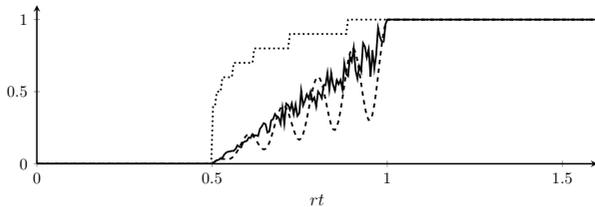


Figure 2: Arbitrary membership functions

Furthermore, it is possible to change the error detection sensitivity by raising e_{fuzzy} to an exponent, as depicted in Figure 3 using μ_F . We can see that by raising e_{fuzzy} to an exponent in the interval $[1, +\infty]$, the error detection becomes less sensitive to errors in the *fuzzy* zone (i.e., the error value in the fuzzy zone becomes smaller than the original). In contrast, by raising e_{fuzzy} to an exponent in the interval $[0, 1]$,

the error detection becomes more sensitive to errors. In fact, when the exponent tends to either $+\infty$ or 0 , the fuzzy membership becomes a binary membership and errors in the fuzzy zone become passes and fails, respectively.

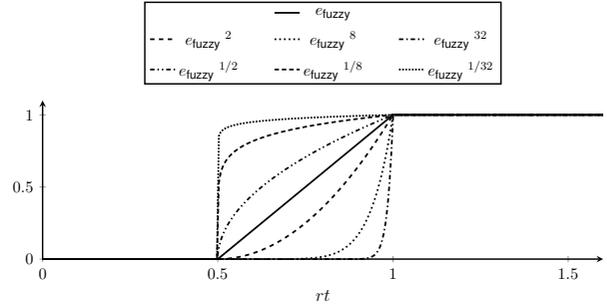


Figure 3: Error detection sensitivity intuition

t	rt	A	e	
			$e_{\text{crisp}}(rt)$	$e_{\text{fuzzy}}(rt)$
1	0.3	$\{c_2\}$	0	0
2	0.9	$\{c_1\}$	0	0.8
3	1.5	$\{c_1, c_2\}$	1	1

Table 1: Fuzzy error hit spectrum example

To illustrate the fuzzy error detection process, consider the spectrum presented in Table 1, which also contains the run-times for each transaction (marked in Figure 1). From this spectrum we can see that, in particular for t_2 , the crisp error vector neglected an error symptom whereas the fuzzy error vector categorized that same transaction as being 80% degraded.

Fuzzy Error Diagnosis

Using fuzzy logic to detect errors, it is possible to assert that a particular transaction is 80% degraded (i.e., $\mu_{\bar{P}} = 0.8$ and consequently $\mu_P = 0.2$). The remaining challenge consists in integrating this additional knowledge in the diagnostic process.

As an example, consider again the spectrum depicted in Table 1. Using the approach explained in the previous section, more concretely Barinel [3], with $e \in \{0, 1\}$, it follows that the diagnosis candidates⁵ $d_1 = \{c_1\}$ and $d_2 = \{c_2\}$ are ranked equally. This is because the components in both candidates are involved in the same number of passed and failed transactions. However, intuitively we would expect d_1 to be ranked ahead of d_2 since transaction t_2 , in which component c_1 was involved, shows error symptoms whereas t_1 does not.

To solve this limitation we make use of the concept of probability of a fuzzy event in Equation 4. From [30], it follows that the probability of a fuzzy event is defined as

$$\Pr(A_i, e_i|d) = \sum_{\omega \in \Omega} \mu_{\omega}(x) \cdot \Pr(A_i, e_i|d) \quad (9)$$

where $\Omega \in \{F, P\}$ and x is an arbitrary event (an observation; rt in

⁵ The candidates for the fuzzy approach are calculated by setting a threshold for e to discretize transactions in terms of pass/fail. In this example we use the threshold $e = 1$.

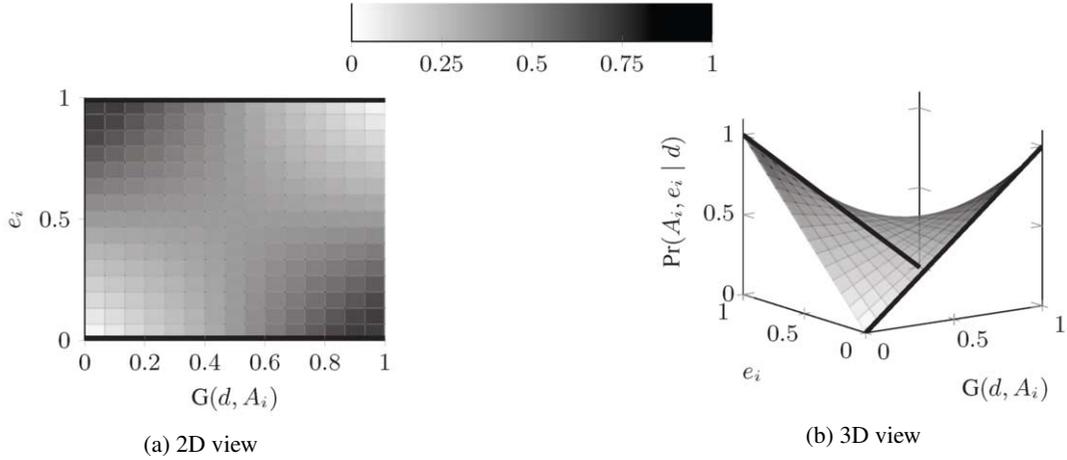


Figure 4: Likelihood function plot

our working example). From Equation 9, Equation 4 generalizes to

$$\Pr(A_i, e_i | d) = \underbrace{e_i \cdot (1 - G(d, A_i))}_{\omega=F} + \underbrace{(1 - e_i) \cdot G(d, A_i)}_{\omega=P} \quad (10)$$

where $e_i = \mu_F(\omega)$ and $1 - e_i = \mu_P(\omega)$. The term $(1 - G(d, A_i))$ accounts for Equation 4's first branch, and $G(d, A_i)$ for the second one. In contrast to Equation 4, this generalization is valid for fuzzy error values (i.e., $e = e_{\text{fuzzy}}$). Figure 4 shows the plot of the Equation 10 with respect to e_i and $G(d, A_i)$. For comparison, we also plot Equation 4 with thick black lines.

We next illustrate the framework for our running example. The probabilities of the two candidates (d_1, d_2) are calculated as follows

$$\Pr(d|\mathbf{A}) = \Pr(d) \cdot \prod_{i \in \{1, 2, \dots, N\}} \frac{\Pr(A_i, e_i, rt|d)}{\Pr(A_i)} \quad (11)$$

where

$$\Pr(d_1) = \Pr(d_2) = \frac{1}{1000} \cdot (1 - \frac{1}{1000}) = 9.99 \times 10^{-4} \quad (12)$$

$$\Pr(A, e, rt|d_1) = \underbrace{(0.8 \cdot (1 - g_1) + (1 - 0.8) \cdot g_1)}_{t_2} \times \underbrace{(1 \cdot (1 - g_1) + (1 - 1) \cdot g_1)}_{t_3} \quad (13)$$

$$\Pr(A, e, rt|d_2) = \underbrace{(0 \cdot (1 - g_2) + (1 - 0) \cdot g_2)}_{t_1} \times \underbrace{(1 \cdot (1 - g_2) + (1 - 1) \cdot g_2)}_{t_3} \quad (14)$$

As an example, setting g_1 and g_2 as 0.1 yields the diagnostic ranking $\langle d_1, d_2 \rangle$, where the true faulty explanation is the first place of the ranking. However, as component goodnesses are assumed not to be available, g_1 and g_2 are estimated using maximum likelihood estimation in both symbolic expressions: $\Pr(A, e | d_1)$ and $\Pr(A, e | d_2)$.

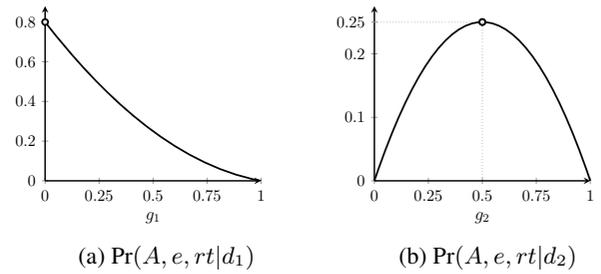


Figure 5: Likelihood plots

As can be seen directly from Figure 5, the Prs are maximized when $g_1 \approx 0$ and $g_2 = 0.5$.

Applying the maximizing values to both expressions, it follows that

$$\Pr(A, e, rt|d_1) = 0.8$$

$$\Pr(A, e, rt|d_2) = 0.25$$

Hence,

$$\Pr(d_1|\mathbf{A}) = 0.76$$

$$\Pr(d_2|\mathbf{A}) = 0.24$$

The diagnostic ranking is therefore $\langle d_1, d_2 \rangle$, which breaks the ambiguity between d_1 and d_2 , thus improving the diagnostic accuracy.

Benchmark

In this section we describe our benchmark approach and discuss results.

Simulator

Performing benchmarks on real applications requires extensive adaptation and is therefore very time consuming. Furthermore, the use of a limited set of applications limits the generalization of the conclusions.

To overcome such issues, we make use of a simulator as proposed in [9]⁶. The simulator provides functions to describe and execute a probabilistic model of an arbitrary system, thereby gathering the required spectra. The authors of the simulator showed [9] that the benchmark results for both real and synthetic data are comparable. This is mainly due to the fact that, since systems are highly abstracted, the spectra generated by real and simulated systems is similar.

The simulator consists of a stack automaton which takes as input a probabilistic model of the system (such as, for instance, the one depicted in Figure 6) and, through a Monte Carlo process, generates spectra. The probabilistic model describes the systems topology, the interaction between components, and the systems faults. To build the topology portion of the model, two primitives exist: components and links (depicted in white and light gray, respectively).

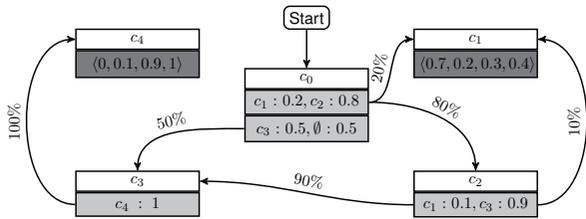


Figure 6: Probabilistic topology model

Concretely, a component (identified by its numeric ID) contains a list of links. A link consists of a list of component IDs with associated transition probabilities (\emptyset corresponds to no transition). Whenever a component is activated all the links belonging to that particular component are sequentially activated. A link is an abstraction to the components' interaction that contains a set of component IDs with their respective call probabilities. With the activation of a link, the current component and link list position are pushed onto a call stack and a component is randomly selected to continue the execution. At the end of the component's execution, an element is popped from the call stack, returning the control to the caller component. Using this model, a transaction can be generated by pushing a component marked as an entry point onto the call stack.

To emulate the error behavior, components may be injected with faults (depicted in dark gray), which are parameterized over 4 variables (p_c , p_d , p_i , and p_f). p_c , p_d , and p_i correspond to the probabilities of correct, degraded, and incorrect behavior. During the simulation, whenever a faulty component is activated, the outcome of such activation (in terms of correct, degraded, or incorrect) is randomly determined using such probabilities. In the event of a component performing erroneously, it has an associated probability p_f of failure which, whenever it occurs, it results in a premature end of the transaction (in Figure 6, an error in component c_4 always results in a failure whereas an error in component c_1 only has a 40% chance of resulting in a failure). To determine the transaction's fuzzy error value, we apply the following rules:

$$e = \begin{cases} 1, & \text{if at least one component performed erroneously} \\ 0, & \text{if all components performed correctly} \\ \text{rand}(0, 1), & \text{otherwise} \end{cases} \quad (15)$$

Setup

We generate the spectra required for our benchmark in two phases. In the first stage, we randomly generate a set of system models, while in the second, we use such models to generate the required spectra.

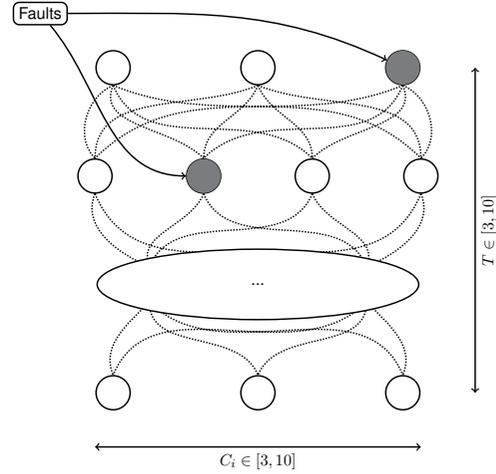


Figure 7: N-tier service architecture

We generated system models that comply with a N-tier service architecture (Figure 7). Systems were created by randomly selecting the number of tiers ($T \in [3, 10]$) as well as the number of components in each tier ($C_i \in [3, 10]$, $1 \leq i \leq T$). Every component is connected to all the components of the next tier with random transition probabilities. To exhibit erroneous behavior, a number of faults (F) was randomly injected (in terms of position) in the systems.

In our setup, we generated 100 systems for each value $F \in [2, 4]$, totaling 300 systems. The injected faults had 90% and 10% probabilities of degraded, and erroneous behavior, respectively.

The spectra generation is parameterized over a single variable E , representing the number of errors at the end of which the simulation stops. For each generated system, we ran 10 simulations for each value $E \in [1, 9]$, totaling 90 spectra per system. Overall, our benchmark is composed of $300 \times 90 = 27000$ test cases.

Metrics

The wasted effort metric evaluates how many components need to be inspected before all faulty components are found [25]. To calculate this metric one must undergo an iterative process. Starting with the first candidate, all of the candidate's components are inspected to determine whether or not that particular set of components was responsible for the erroneous behavior. Depending on the result of such inspection two outcomes may occur. On the one hand, if the component is found to be faulty, that particular component is removed from all other candidates in the ranking. On the other hand, if the component is found to be healthy, all candidates in the ranking containing that particular component are removed. This process is repeated until all faulty components are found. In the case of the last inspected candidate being tied with other candidates, it is assumed that, on average, half of the healthy components are examined.

During this iterative process, we keep track of two counters: inspected components (I) and faulty components (C). Using these two counters, the wasted effort metric is calculated as

$$W = I - C \quad (16)$$

⁶ <https://github.com/SERG-Delft/sfl-simulator>

	d	Rank	I	C
1	$\{c_1, c_4\}$	1	2	1
2	$\{c_2, c_3, c_4\}$	2		
3	$\{c_3, c_4, c_5\}$	3		
4	$\{c_1, c_2\}$	4	4	2
5	$\{c_3, c_5\}$	4		

Table 2: Example diagnostic report

As an example consider the diagnostic report presented in Table 2 for which the correct diagnostic candidate is $d = \{c_1, c_2\}$. In order to calculate the wasted effort, we start by examining c_1 and c_4 finding that c_1 is faulty while c_4 is healthy. Due to c_4 being healthy, candidates d_2 and d_3 are not examined. Examining d_4 we observe that the only unexplored component (c_2) is faulty. Additionally, we see that both system's faults were discovered. However, as d_5 is tied with d_4 , we must inspect half of the healthy components. The wasted effort of this diagnosis is therefore $W = 4 - 2 = 2$, meaning that 2 healthy components (c_4 and c_3/c_5) were examined in the process of finding the root cause of the system errors.

A normalized version of the wasted effort is called diagnostic quality and is defined as

$$Q = 1 - W/(M - C) \quad (17)$$

where M is the number of system components. The diagnostic quality value is contained between zero and one and estimates the fraction of system's healthy components that need to be examined before all faulty components are found.

In this paper we refine the diagnostic quality metric to take into account the fact that, for a specific spectrum, not all components of the system can be at fault. As an example consider a system with 1000 components with a spectrum consisting of a single failing transaction activating 2 components. Assuming the diagnostic algorithm only proposes plausible⁷ candidates, the quality is contained in the interval between 1 and $\frac{999}{1000}$. Instead of calculating the diagnostic quality using the M components of the system, we use M_s , the number of "suspicious" components to calculate the new metric, which shall be referred to as "fair quality" (Q_f). A component is said to be suspicious if it was activated in a failing transaction. A consequence of using Q_f is that the diagnostic qualities of all possible permutations of the ranking always have a lower bound quality of 0.

Results

We compare the performance of the crisp, state-of-the-art, diagnostic approach, presented in the Reasoning-based Diagnosis Section, with our fuzzy approach for the generated spectra. For more information regarding this approach, see [3].

In Figure 8, we compare the average Q_f for each test scenario. From the analysis of the plot we can see that the fuzzy approach always (on average) outperforms by the crisp approach. This is due to the fact that the fuzzy approach is able to successfully take advantage of the extra information to break the ties in the ranking (as shown in the example from Table 1) that occur when dealing with small numbers of erroneous transactions.

A more detailed analysis of the data (Figure 9) shows that our approach outperformed the crisp approach in 65% of the test cases. Moreover, in 94% of the cases our approach was at least as accurate as the classical approach. In the remaining 6% of the test cases the accuracy loss was due to (1) lack of observations, and (2) marginal

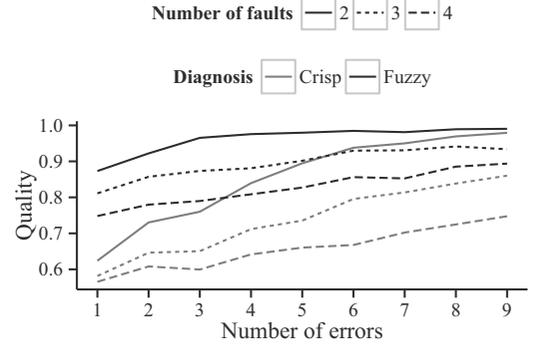


Figure 8: Benchmark Averages

variations in the posterior probability, still enough to make the relative ranking change. The overall average improvement of quality introduced by our algorithm was of $\Delta Q_f = 0.153$, representing a relative improvement of 21%. By performing a paired one-tailed T-test, we can ascertain that our approach introduced a relative improvement of 20%, with a 99% confidence interval.

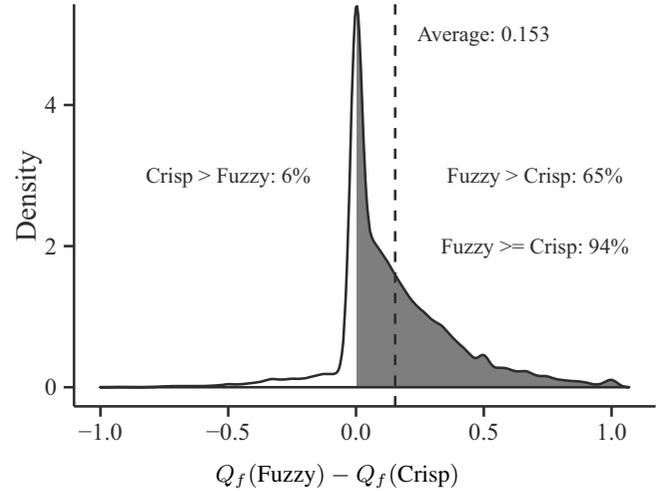


Figure 9: Quality improvement density plot

In Figure 10, we present a set of boxplots⁸ comparing the quality distributions of both approaches for each test scenario. From the analysis of the plots we can see that not only the fuzzy approach has a better performance than the crisp approach, but also that the fuzzy approach distribution is much more skewed towards better quality results than the crisp approach. Additionally, we can see that the fuzzy approach exhibits a higher consistency (i.e., smaller inter-quartile range) than the crisp approach.

A final remark is that, with the increase of erroneous transactions, it appears that the crisp approach quality seems to converge towards to the same average quality as the fuzzy approach. This happens due to the fact that the information introduced by the occurrence of errors eventually compensates the limitations imposed by the crisp error abstraction.

⁸ For each test scenario, the box corresponds to 2nd and 3rd quartiles (i.e., 50% of the cases), the vertical lines correspond to the 1st and 4th quartiles, and the small dashes correspond to test cases categorized as outliers. A test case is considered to be an outlier if its distance from the box is greater than $1.5 * IQR$ (inter-quartile range, i.e., the height of the box).

⁷ By plausible we mean that all the candidate's components were at least activated once in an erroneous transaction.

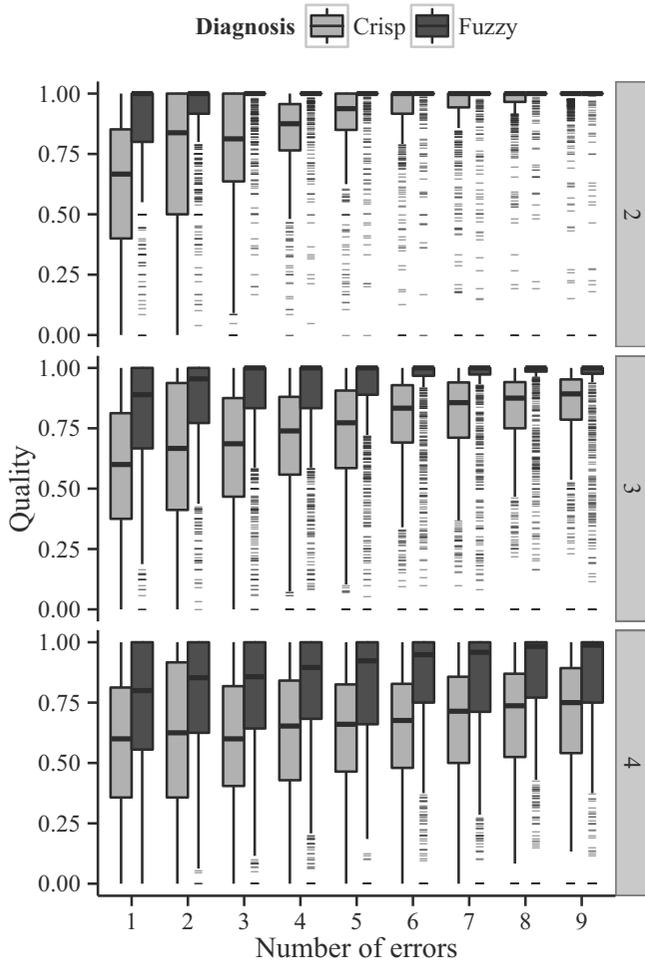


Figure 10: Benchmark Boxplots

Related Work

SFL and the improvement described in this paper occupy the ground between machine learning and model-based diagnosis [20]. Machine learning techniques [27] require a lot of training data which in the case of SFL is hit spectra. Hit spectra may look like a lot of data, however, the “per component” information is limited (we get n labels per component). Further, the per component pass/no pass information is Boolean and machine learning has to be trained for exponential number of multiple-faults, something that is not required in our approach. Last, if a lot of learning data is available, then machine learning easily overfits due to the Boolean nature of the features.

Model-based diagnosis is an attractive diagnostic approach because some frameworks are complete, others fast [11], and the complete ones use all available information to reason from symptoms to diagnoses. There are two fundamental problems with MBD approaches, however: (1) the exponential reasoning cost for complete algorithms and (2) the modeling problem for all MBD-based algorithms. Complex logical systems like computer programs cannot be mathematically modeled without encountering problems with decidability and halting.

Technique similar to ours, but applied to a different problem domain is the one of Kahuna [26]. There, the authors apply peer similarity to diagnose performance problems in map-reduce systems. In the black-box approach, the authors use learning from successfully passed tests. The Kahuna [26] black-box approach can be improved

by using non-Boolean classification similar to the one proposed in this paper.

In [18], the authors apply the same concept to the diagnosis of failures in distributed file systems, such as PVFS or Lustre.

Computer programs nowadays fail mostly due to combination of faults. Single-faults are ruled-out with the help of classical debugging and unit-testing. Reasoning about multiple faults comes at steep computational price. Efficient lightweight fault localization techniques to diagnose software systems are PINPOINT [10], TARANTULA [17], and OCHIAI [2]. While extremely efficient, they do not reason in terms of multiple faults.

The approach in this paper improves state-of-the-art multiple-fault health estimation. This gives increased accuracy compared to simple similarity-based SFL which is optimal only in the case of single-faults. The improved ranking is shown and analyzed in [3], [19], and [6].

The algorithm described in this paper is very suitable for troubleshooting of large and complex, network systems. There are various latencies and degradations in these systems that make the pass/fail qualification of a test inadequate. Discrete event systems [24] are currently used for monitoring and diagnosis of networked systems but they have enormous computational cost and, unlike our approach, are more difficult to randomize and sample.

The approach in this paper is related to hybrid automata [15], an approach also originating in the control community. Hybrid automata merge Hidden-Markov Models and dynamic systems.

Conclusions

We presented a generalization to the classical reasoning-based approach that not only guarantees equal diagnostic quality when diagnosing functional failures but also improves the diagnostic quality when diagnosing performance degradation failures (soft failures).

The conducted synthetic benchmark showed that, for our setup with 27000 test cases, our approach improved the diagnostic quality in 65% of the cases and performed at least as good as the classical approach in 94% of the test cases. On average, the relative improvement introduced by our approach was of 20%, with a 99% confidence interval.

Future work includes the extension of existent error detection frameworks to include the fuzzy error abstraction proposed in this paper. Comparison with other approaches than the crisp approach (Barinel) has remained for future work, because our goal was to improve current crisp-based SFL approach (relevant due to its low-cost modeling and diagnostic effort, thus scaling to large, real software/hardware systems) in order to deal gracefully with non-functional errors. The existence of such a framework would enable a real-world validation of the proposed approach. Furthermore, the broader impact of this paper is that the approach paves the way to automatic oracles in the context of automatic test generation and self-healing systems.

Acknowledgements

We would like to thank Ion Matei, Lígia Massena, André Silva, and Alexandre Perez for the useful discussions about this work. This material is based upon work supported by the National Science Foundation under Grant No. CNS 1116848, and by the scholarship number SFRH/BD/79368/2011 from Fundação para a Ciência e Tecnologia (FCT).

REFERENCES

- [1] Rui Abreu and Arjan J. C. Van Gemund. A low-cost approximate minimal hitting set algorithm and its application to model-based diagnosis. In *Proceedings of the 8th Symposium on Abstraction, Reformulation, and Approximation*, SARA'09, 2009.
- [2] Rui Abreu, Peter Zoetewij, and Arjan J. C. Van Gemund. On the accuracy of spectrum-based fault localization. In *Proceedings of the 2nd International Academic And Industrial Conference on Testing – Practice And Research Techniques*, TAICPART'07, pages 89–98, 2007.
- [3] Rui Abreu, Peter Zoetewij, and Arjan J. C. Van Gemund. A new bayesian approach to multiple intermittent fault diagnosis. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, IJCAI'09, pages 653–658, 2009.
- [4] Rui Abreu, Peter Zoetewij, and Arjan J. C. Van Gemund. Spectrum-based multiple fault localization. In *Proceedings of the 2009 International Conference on Automated Software Engineering*, ASE'09, pages 88–99, 2009.
- [5] Nuno Cardoso and Rui Abreu. MHS²: A map-reduce heuristic-driven minimal hitting set search algorithm. In *Proceedings of the 2013 International Conference on Multicore Software Engineering, Performance, and Tools*, MUSEPAT'13, pages 25–36, 2013.
- [6] Nuno Cardoso and Rui Abreu. A kernel density estimate-based approach to component goodness modeling. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, AAAI'13, 2013.
- [7] John Carey, Neil Gross, Marcia Stepanek, and Otis Port. Software hell. In *Business Week*, pages 391–411, 1999.
- [8] Paulo Casanova, David Garlan, Bradley Schmerl, and Rui Abreu. Diagnosing architectural run-time failures. In *Proceedings of the 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS'13, pages 103–112, 2013.
- [9] Cuiting Chen, Hans-Gerhard Gross, and Andy Zaidman. Improving service diagnosis through increased monitoring granularity. In *Proceedings of the 7th International Conference on Software Security and Reliability*, SERE'13, pages 129–138, 2013.
- [10] Mike Y. Chen, Emre Kiciman, Eugene Fratkin, Armando Fox, O Fox, and Eric Brewer. Pinpoint: Problem determination in large, dynamic internet services. In *Proceedings of the 2002 International Conference on Dependable Systems and Networks*, DSN'2002, pages 595–604, 2002.
- [11] Alexander Feldman, Gregory Provan, and Arjan J. C. Van Gemund. Computing minimal diagnoses by greedy stochastic search. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, AAAI'08, pages 911–918, 2008.
- [12] Debanjan Ghosh, Raj Sharman, H. Raghav Rao, and Shambhu Upadhyaya. Self-healing systems - survey and synthesis. *Decision Support Systems in Emerging Economies*, 42(4):2164–2185, 2007.
- [13] Alberto Gonzalez-Sanchez, Rui Abreu, Hans-Gerhard Gross, and Arjan JC van Gemund. Spectrum-based sequential diagnosis. In *AAAI'11*, 2011.
- [14] Mary Jean Harrold, Gregg Rothermel, Rui Wu, and Liu Yi. An empirical investigation of program spectra. In *Proceedings of the 1998 Workshop on Program Analysis for Software Tools and Engineering*, PASTE'98, pages 83–90, 1998.
- [15] Michael W Hofbaur and Brian C Williams. Mode estimation of probabilistic hybrid systems. In *Hybrid Systems: Computation and Control*, pages 253–266. Springer, 2002.
- [16] Birgit Hofer and Franz Wotawa. Spectrum enhanced dynamic slicing for better fault localization. In *Proceedings of the 20th European Conference on Artificial Intelligence*, ECAI'12, pages 420–425, 2012.
- [17] James A. Jones and Mary Jean Harrold. Empirical evaluation of the tarantula automatic fault-localization technique. In *Proceedings of the 20th International Conference on Automated Software Engineering*, ASE'05, pages 273–282, 2005.
- [18] Michael P. Kasick, Jiaqi Tan, Rajeev Gandhi, and Priya Narasimhan. Black-box problem diagnosis in parallel file systems. In *Proceedings of the 8th Conference on File and Storage Technologies*, FAST, pages 43–56, 2010.
- [19] Johan De Kleer. Diagnosing multiple persistent and intermittent faults. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, IJCAI'09, pages 733–738, 2009.
- [20] Johan De Kleer and Brian C Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.
- [21] Johan De Kleer and Brian C. Williams. Readings in model-based diagnosis. In *Readings in Model-Based Diagnosis*, pages 100–117. Morgan Kaufmann Publishers Inc., 1992.
- [22] W. Mayer and M. Stumptner. Model-based debugging using multiple abstract models. In *Proceedings of the 2003 International Workshop on Automated and Analysis-Driven Debugging*, AADEBUG'03, pages 55–70, 2003.
- [23] E. Piel, A. Gonzalez-Sanchez, H-G. Gross, Arjan J. C. Van Gemund, and R. Abreu. Online spectrum-based fault localization for health monitoring and fault recovery of self-adaptive systems. In *Proceedings of the 8th International Conference on Autonomic and Autonomous Systems*, ICAS'11, pages 64–73, 2012.
- [24] Meera Sampath, Raja Sengupta, Stephane Lafortune, Kasim Sinnamohideen, and Demosthenis C Teneketzis. Failure diagnosis using discrete-event models. *Control Systems Technology, IEEE Transactions on*, 4(2):105–124, 1996.
- [25] Friedrich Steimann, Marcus Frenkel, and Rui Abreu. Threats to the validity and value of empirical assessments of the accuracy of coverage-based fault locators. In *Proceedings of the 2013 International Symposium on Software Testing and Analysis*, ISSTA'2013, pages 314–324, 2013.
- [26] Jiaqi Tan, Xinghao Pan, Eugene Marinelli, Soila Kavulya, Rajeev Gandhi, and Priya Narasimhan. Kahuna: Problem diagnosis for mapreduce-based cloud computing environments. In *Proceedings of the 12th Network Operations and Management Symposium*, NOMS, pages 112–119, 2010.
- [27] W Eric Wong and Yu Qi. Bp neural network-based effective fault localization. *International Journal of Software Engineering and Knowledge Engineering*, 19(04):573–597, 2009.
- [28] Franz Wotawa. A variant of reiter's hitting-set algorithm. *Information Processing Letters*, 79(1):45–51, 2001.
- [29] Lotfi Asker Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.
- [30] Lotfi Asker Zadeh. Probability measures of fuzzy events. *Journal of Mathematical Analysis and Applications*, 23(2):421–427, 1968.
- [31] Tom Zamir, Roni Stern, and Meir Kalech. Using model-based diagnosis to improve software testing. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1135–1141. AAAI Press, 2014.

Online Adaptation of Deep Architectures with Reinforcement Learning

Thushan Ganegedara, Lionel Ott and Fabio Ramos¹

Abstract. Online learning has become crucial to many problems in machine learning. As more data is collected sequentially, quickly adapting to changes in the data distribution can offer several competitive advantages such as avoiding loss of prior knowledge and more efficient learning. However, adaptation to changes in the data distribution (also known as covariate shift) needs to be performed without compromising past knowledge already built in into the model to cope with voluminous and dynamic data. In this paper, we propose an online stacked Denoising Autoencoder whose structure is adapted through reinforcement learning. Our algorithm forces the network to exploit and explore favourable architectures employing an estimated utility function that maximises the accuracy of an unseen validation sequence. Different actions, such as *Pool*, *Increment* and *Merge* are available to modify the structure of the network. As we observe through a series of experiments, our approach is more responsive, robust, and principled than its counterparts for non-stationary as well as stationary data distributions. Experimental results indicate that our algorithm performs better at preserving gained prior knowledge and responding to changes in the data distribution.

1 Introduction

Over the past decade, Deep Architectures [5], [1] have become a widely-discussed topic in machine learning. One key reason being the ability to jointly perform feature-extraction and classification on raw data, outperforming many other techniques in various domains including object recognition [7], [2], hand-writing recognition [5] and speech recognition [4]. A deep network can be understood as a neural network consisting of many hidden layers [3]. While the interest in deep networks arose quite early, only the recent hardware and optimisation developments (e.g. Graphical Processing Units (GPUs), Greedy pre-training) sparked the practicality of deep architectures.

Despite the note-worthy learning capacity, deep architectures are still susceptible to the past-knowledge being overridden due to *Covariate Shift* [15]. Covariate shift is a common phenomenon that transpires in online settings. Covariate shift essentially refers to the difference in training and testing data distributions. Successful exploitation of adaptive capabilities of deep networks to minimise the adverse effects of the covariate shift will lead to new frontiers in data science.

While many algorithms (especially Support Vector Machines (SVM)) have been enhanced with online learning capabilities [9], [11], only few attempts of incorporating online learning for Neural Networks have been proposed in the literature, notably in [19], [13], [10], and [14]. Of these, only [14] and [19] focus on

changing the structure of the network, where the others focus on adapting a fixed architecture accordingly. [14] proposes an intriguing approach to evolve neural networks using genetic algorithm, by mutating weights and nodes in the network and crossing over existing networks to generate more fit off-springs. However, this technique is not scalable for deep networks and requires many repetitive runs through the data. [19] proposes a structural adaptation technique for deep architectures relying on simple heuristic (i.e. immediate performance convergence). [19] does not seek a long-term reward and lacks in responsiveness, as it waits for a pool of data to be filled in order to add nodes to the structure. These limitations motivate the question of how to explore the space of different architectures in an online setting in a more responsive, robust and principled manner.

In this paper, we introduce a state-of-the-art mechanism to modify deep architectures (specifically Denoising Autoencoders [18]) based on reinforcement learning. The decision making behaviour exploits and explores possible actions to discover favourable modifications to the structure (i.e. adding/removing nodes) by maximising a stipulated reward over time. Adding nodes helps to accommodate new features, while removing nodes helps to remove redundant features. An additional pooling operation fine-tunes the network with previously observed data. The method keeps track of a continuously updated utility (long-term reward) function to decide which action is best for a given state, whose estimation will improve over time. The experimental results on three datasets clearly show that our algorithm outperforms its counterparts in both stationary and non-stationary situations.

2 Background

2.1 Online Learning

By online learning we refer to the ability to accommodate new knowledge (i.e. features) without overriding previously acquired knowledge (i.e. features) [17]. This is becoming more popular due to the explosive growth of data. Online learning has the ability to learn from a continuous stream of data without a loss of past knowledge and attempts to address the non-stationary nature of data by allowing more flexibility in the model. For this reason, online algorithms perform significantly better in handling problems with covariate shift.

2.2 Deep Networks

We begin the presentation of the method by introducing the following notation:

- \mathbf{x} - Inputs
- \mathbf{y} - Input labels
- K - Number of classes

¹ University of Sydney, Australia email: tgan4199@uni.sydney.edu.au, lott4241@uni.sydney.edu.au and fabio.ramos@sydney.edu.au

- $\tilde{\mathbf{x}}$ - Noise-corrupted input
- $\hat{\mathbf{x}}$ - Reconstructed input
- W - Weights of a neuron layer
- b - Bias of a neuron layer
- b' - Reconstruction bias of a neuron layer

2.2.1 Autoencoder

An Autoencoder [6] maps a set of inputs $\mathbf{x} = \{\mathbf{x}_i \in [0, 1]^D\} \forall i = 1, \dots, N$ where $\mathbf{x}_i = \{x_i^1, x_i^2, \dots, x_i^D\}$ and D is dimensionality of data to a latent feature space H with $h_{W,b}(\mathbf{x}) = sig(W\mathbf{x}+b)$, where $W \in \mathbb{R}^{H \times D}$, $b \in \mathbb{R}^H$ and $sig(s) = \frac{1}{1+\exp-s}$. An autoencoder can reconstruct the input $\hat{\mathbf{x}}_i \forall i = 1, \dots, N$ from the latent feature space H with $\hat{\mathbf{x}} = sig(W^T \times h_{W,b}(\mathbf{x}) + b')$ where superscript T denotes transpose and $b' \in \mathbb{R}^D$. For simplicity we assume tied weights.

2.2.2 Denoising Autoencoder

The Denoising Autoencoder (DAE) is a variant of autoencoder which uses a corrupted (noisy) version of the example as the input [18]. This forces the algorithm to become more robust to noise. DAE works in the following manner.

First, the inputs are corrupted by introducing noise using a binomial distribution with probability p . Let us call the corrupted input $\tilde{\mathbf{x}}$. Next, $\tilde{\mathbf{x}}$ is mapped to a hidden representation using $h_{W,b}(\tilde{\mathbf{x}}) = sig(W\tilde{\mathbf{x}}+b)$ where $W \in \mathbb{R}^{H \times D}$, $b \in \mathbb{R}^H$ and $sig(s) = \frac{1}{1+\exp-s}$. Finally, the decoding function retrieves the reconstructed input, $\hat{\mathbf{x}} = sig(W^T \times h_{W,b}(\tilde{\mathbf{x}}) + b')$, where $b' \in \mathbb{R}^D$. In this work, we assume tied weights for encoding and decoding. *Cross entropy* is used as the cost function (Equation 1),

$$L_{gen}(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{j=1}^D x^j \log(\hat{x}^j) + (1 - x^j) \log(1 - \hat{x}^j). \quad (1)$$

The optimal values for parameters W, b, b' are found by minimising the cost function,

$$W_{opt}, b_{opt}, b'_{opt} = \operatorname{argmin}_{W,b,b'} L_{gen}(\mathbf{x}, \hat{\mathbf{x}}).$$

2.2.3 Stacked Denoising Autoencoders

A Stacked Denoising Autoencoder (SDAE) [18] is a set of connected autoencoders. A SDAE undergoes two main processes; pre-training and fine-tuning. In the pre-training process, the network is considered as a set of autoencoders AE^1, \dots, AE^L . The output of $AE^l = \{W^l, b^l, b'^l\}, h_{W,b}^l$ where l is the current layer, is calculated as follows,

$$h_{W,b}^l = \begin{cases} sig(W^l \tilde{\mathbf{x}} + b^l); & \text{if } l = 1 \\ sig(W^l h_{W,b}^{l-1} + b^l); & \text{Otherwise.} \end{cases}$$

In the fine-tuning phase, the network is treated as a single deep autoencoder and trained using labelled data \mathcal{D} . Assuming labelled data in the format $\mathcal{D} = (\mathbf{x}_i, \mathbf{y}_i), \forall i = 1, \dots, N$ where $\mathbf{y}_i \in \{0, 1\}^K$ such that if y_i^j are the elements of \mathbf{y}_i then $\sum_j y_i^j = 1$, we can use a *softmax* layer with parameters $\{W^{out}, b^{out}, b'^{out}\}$. The output of the network is defined as $\hat{\mathbf{y}} = \operatorname{softmax}(W^{out} h_{W,b}^L + b'^{out})$, where $\operatorname{softmax}(a_k) = \frac{\exp(a_k)}{\sum_{k'} \exp(a_{k'})}$. Then the cost function becomes,

$$L_{disc}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{j=1}^K (y^j \log \hat{y}^j + (1 - y^j) \log(1 - \hat{y}^j)). \quad (2)$$

Finally, from Equation 2, we can formulate the optimisation problem to learn W, b and b' as,

$$W_{opt}, b_{opt}, b'_{opt} = \operatorname{argmin}_{W,b,b'} L_{disc}(\mathbf{y}, \hat{\mathbf{y}}),$$

where $W_{opt} = (W_{opt}^1, \dots, W_{opt}^L, W_{opt}^{out})$, $b_{opt} = (b_{opt}^1, \dots, b_{opt}^L, b_{opt}^{out})$ and $b'_{opt} = (b'_{opt}^1, \dots, b'_{opt}^L, b'_{opt}^{out})$.

2.3 Incremental Feature Learning for Denoising Autoencoders

Merge-Incremental Denoising Autoencoders (MI-DAE) is an online learning stacked denoising autoencoder proposed in [19]. Initially, the network is pre-trained using a pool of data (typically first 12,000 examples). Then, for every batch of data b_t , add hard examples (i.e. \mathbf{x}_i if $L_{gen}(\mathbf{x}_i, \hat{\mathbf{x}}_i) > \frac{\sum_{\forall \mathbf{x}_j \in b_t} L_{gen}(\mathbf{x}_j, \hat{\mathbf{x}}_j)}{|b_t|}$) to a pool, B . The method then performs merging of nodes within the same layer or adds new nodes to the network. Once the number of points in B exceeds a threshold, τ , retrieve previously calculated pairs of nodes with the highest similarity (ΔMrg) and add ΔInc new nodes to the network. Next, use B to greedily train newly added features. Afterwards, update ΔMrg and ΔInc [20] and remove all data from B . Finally repeat this process for all the batches in the sequence. Pseudo-code for this algorithm is presented in Algorithm 1.

Algorithm 1 MergeInc Algorithm

```

1: procedure MERGEINC( $b_t, \Delta\text{Mrg}, \Delta\text{Inc}$ )
2:   Define:  $\mu$  - Average reconstruction error for the
3:     most recent 10,000 examples
4:   Define:  $\tau$  - Pool threshold (10,000 examples)
5:   Compute objective  $L_{disc}(\mathbf{y}_j, \hat{\mathbf{y}}_j), \forall \{\mathbf{x}_j, \mathbf{y}_j\} \in b_t$ 
6:   Add hard example  $\mathbf{x}_j$  to  $B$  if  $L_{gen}(\mathbf{x}_j, \hat{\mathbf{x}}_j) > \mu$  of  $b_t$ 
7:   if  $|B| > \tau$  then
8:     Merge  $2\Delta\text{Mrg}$  candidates to  $\Delta\text{Mrg}$ 
9:     Add  $\Delta\text{Inc}$  nodes and fine-tune  $\Delta\text{Inc}$  new nodes with
10:     $\{\mathbf{x}_j, \mathbf{y}_j\} \in B$  while keeping rest of the network constant
11:    Update  $\Delta\text{Mrg}$  and  $\Delta\text{Inc}$  (Heuristic-based [20])
12:    Set  $B = \emptyset$ 
13:   end if
14:   Fine-tune all the features (with  $\Delta\text{Mrg}$  and  $\Delta\text{Inc}$ ) with  $b_t$ 
15: end procedure

```

2.4 Reinforcement Learning and Markov Decision Processes

After describing SDAE, we now introduce notation and the basics of reinforcement learning (RL). RL enables an agent to learn a *policy*, π (a function that defines which action to take in a given state), by interacting with its environment, preferably trading-off between exploration and exploitation. A reinforcement learning task that satisfies the *Markov Property* can be formulated as a Markov decision process (MDP) [16]. Formally a Markov Decision Process can be defined using the following,

- A set of states - S
- A set of actions - A
- A transition function - $T : S \times A \times S \rightarrow [0, 1]$
- A reward function - $R : S \times A \times S \rightarrow \mathbb{R}$.

Table 1. The notations and definitions used in Section 3

Notation	Description	Notation	Description
N	Number of data points	B_r	Pool of data containing most recent τ examples
D	Dimensionality of data	B_{ft}	Pool of data containing dissimilar inputs
K	Number of classes	Λ	Distance threshold for B_{ft}
p	Number of data points in one batch	$\tilde{L}^n(m)$	Exponential Moving Average of error L in the window $n - m$ to n
n	sequence number of the current batch of data	ν_l^n	Ratio between the current count of neurons and the initial count for neuron layer l for n^{th} data batch
τ	Size of data pools	ΔInc	Number of neurons to add at a given time
\mathbf{x}_i	i^{th} data point	ΔMrg	Number of neurons to remove at a given time
\mathbf{y}_i	Vectorized label of \mathbf{x}_i s.t $\forall y_i^j \in \mathbf{y}_i, y_i^j \in \{0, 1\}$ s.t. $\sum_j y_i^j = 1$	r^n	The reward for the n^{th} batch of data
\mathcal{D}	Dataset containing $\{\{\mathbf{x}_1, \mathbf{y}_1\}, \{\mathbf{x}_2, \mathbf{y}_2\}, \dots\}$	γ	Discount rate for Q value update
\mathcal{D}^n	n^{th} batch of data ($\mathcal{D}^n \subset \mathcal{D}$)	$Q(s, a)$	Utility function
L_g^n	Generative error for n^{th} batch of data	η_1	The duration until beginning to collect state-action pairs
L_c^n	Classification error for n^{th} batch of data	η_2	The duration until beginning to exploit Q-values

In this paper, RL is used to find the policy to adapt the structure of the network, given the current network configuration or state. Therefore, at a given instance i , from state s^i an action a^i is performed and the network transits to state s^{i+1} . Actions are modifications to the network such as adding new nodes or removing existing nodes. The state is a function of the network performance and will be defined in Section 3. The reward r^i for going from state s^i to s^{i+1} by taking action a^i is calculated based on the errors produced on the learning task. State s^{i+1} depends on the current state s^i and current action a^i , and is conditionally independent of all the previous states and actions, thus satisfying the *Markov Property*. The ultimate goal is to learn an optimal policy $\pi^*(s^i, a^i)$ that recommends the best action a^i for a given state s^i .

In order to learn the policy to select the best action for a given state, Q-Learning is used. Q-Learning (a variant of Temporal difference [16]) is an off-policy model-free approach to finding the optimal policy, π^* . Q-Learning estimates the utility value in an online manner and, as an off-policy learning, it learns a value function independent of the agent's experience. This leads to exploring new tactics the agent has not tried. Furthermore, Q-Learning can be employed for MDPs with unknown transition and reward functions. Q-Learning proceeds as follows,

1. Define $Q(s^n, a^n)$, where $s^n \in S$ and $a^n \in A$.
2. Initialise $Q^0(s^i, a^i) = 0, \forall s^i \in S$ and $\forall a^i \in A$.
3. Update $Q^{t+1}(s^n, a^n) = (1 - \alpha) \times Q^t(s^n, a^n) + \alpha \times [R(s^n, a^n, s^{n+1}) + \gamma(\max_{a'}(Q(s^{n+1}, a')))]$ where γ is the discount rate, α is the learning rate, and s^{i+1} is the state after action a^i .

One of the applications of using Q-learning is to train a multi-layer perceptron as found in [13]. More recently, a variant of Q-Learning was successfully used in a Convolutional Deep Network when the network was trained to play the *Atari* games using raw pixel images [10].

3 Reinforced Adaptive Denoising Autoencoder (RA-DAE)

3.1 Limitations of MI-DAE

MI-DAE (Algorithm 1) introduces several interesting concepts useful for online learning such as, pooling data and update rules for

ΔMrg and ΔInc . However, the approach has several limitations: (1) The response of the algorithm to changes is slow as it waits for a pool of data (B) to be filled in order to execute an operation; (2) While the algorithm incorporates an intuitive criteria (performance convergence) to modify the network (update rules), the method is based on simple heuristics such as the immediate future reward that does not generally reflect a holistic view of the effect an action has on the network.

3.2 Overview of RA-DAE

Motivated by the drawbacks in MI-DAE, we propose a more robust and principled solution which relies on RL. In essence, our algorithm estimates an utility function $Q(s, a)$ for each state-action pair by sampling from the environment, where actions are modifications in the network structure. Using $Q(s, a)$, the algorithm selects the best action for a given state. The utility function is based on the accuracy measured on an unseen validation batch. Our approach is beneficial as,

- Actions are taken for every batch of data, resulting in fast response to sudden changes in the data distribution;
- The utility function ensures that actions are taken based on the long-term benefit they incur on the accuracy;
- A new pool operation refreshes the network's knowledge by fine-tuning the network using a pool of data containing data points *significantly* different from each other.

Notation: An input data stream is denoted as $\mathcal{D} = \{\{\mathbf{x}_1, \mathbf{y}_1\}, \{\mathbf{x}_2, \mathbf{y}_2\}, \dots\}$, where \mathbf{x}_i is a normalized data point, $\mathbf{x}_i \in [0, 1]^D$, $\mathbf{y}_i \in \{0, 1\}^K$ and y_i^j are the elements of \mathbf{y}_i with $\sum_j y_i^j = 1$. The n^{th} data batch is written as $\mathcal{D}^n = \{\{\mathbf{x}_{(n-1) \times p}, \mathbf{y}_{(n-1) \times p}\}, \dots, \{\mathbf{x}_{n \times p}, \mathbf{y}_{n \times p}\}\}$, where p is the number of examples per batch. Denote the generative error as $L_g^n = \frac{\sum_{\forall \mathbf{x}_i \in \mathcal{D}^n} L_{gen}(\mathbf{x}_i, \hat{\mathbf{x}}_i)}{p}$ and the classification (or discriminative) error as $L_c^n = \frac{\sum_{\forall \mathbf{y}_i \in \mathcal{D}^n} \mathbb{1}_{k_i=k'_i}}{p}$, where $\mathbb{1}$ is the *indicator* function and $k_i = \text{argmax}_{k'}(\{y_i^{k'}\})$, $\forall k' = 1, \dots, K$ of the n^{th} batch. r^n denotes the reward for the n^{th} batch. Finally define two pools

$B_r = \{\mathcal{D}^{n-\tau}, \dots, \mathcal{D}^n\}$ and

$$B_{ft} = \begin{cases} \mathcal{D}^n & \text{if } B_{ft} = \emptyset \\ \mathcal{D}^n \cup B_{ft} & \text{if } d(\mathcal{D}^n, \mathcal{D}^j) > \Lambda \quad \forall \mathcal{D}^j \in B_{ft} \\ B_{ft} - \mathcal{D}^j & \text{if } j = \operatorname{argmin}_{j'} (\forall \mathcal{D}^{j'} \in B_{ft}) \text{ if } |B_{ft}| > \tau \\ B_{ft} & \text{otherwise} \end{cases} \quad (3)$$

for some d distance measure and a similarity threshold $\Lambda \in [0, 1]$. η_1 and η_2 are pre-defined thresholds for starting to collect observed state-action pairs and exploiting Q-values respectively. α is the learning rate for Q-learning. A summary of the notation is in Table 1 for quick reference.

3.3 RL Definitions

To calculate when and which actions to take, we employ a MDP formulation. We define a set of states S , a set of actions A , and a reward function r^n below.

3.3.1 State Space

The state space S is defined as follows. For the n^{th} batch,

$$S = \{\tilde{\mathcal{L}}_g^n(m), \tilde{\mathcal{L}}_c^n(m), \nu_1^n\} \in \mathbb{R}^3 \quad (4)$$

where the moving exponential average ($\tilde{\mathcal{L}}$) is defined as $\tilde{\mathcal{L}}^n(m) = \alpha L^n + (1 - \alpha)\tilde{\mathcal{L}}^{n-1}(m-1)$, $n \geq m$ and m is a pre-defined constant. $\tilde{\mathcal{L}}_g$ and $\tilde{\mathcal{L}}_c$ denote $\tilde{\mathcal{L}}$ w.r.t. L_g and L_c , respectively, and $\nu_l^n = \frac{\text{Node Count}_{\text{current}}}{\text{Node Count}_{\text{initial}}}$ for the l^{th} hidden layer. $\tilde{\mathcal{L}}$ is defined in terms of recursive decay to respond rapidly to immediate changes.

This state space takes into account the following attributes:

- Ability of RA-DAE to classify an unseen batch of data;
- Difference between current data distribution and previously observed distributions;
- Complexity of RA-DAE's current structure.

The justification for the choice of state space is discussed in Section 4.2.1.

3.3.2 Action Space

The actions space is defined as,

$$A = \{Pool, Increment(\Delta\text{Inc}), Merge(\Delta\text{Mrg})\}, \quad (5)$$

where *Increment*(ΔInc) adds ΔInc new nodes and greedily initialise them using pool B_r . The *Merge*(ΔMrg) operation is performed by merging the $2\Delta\text{Mrg}$ nodes. *Merge* operation is executed by selecting the closest pairs (e.g. minimum Cosine distance) of ΔMrg nodes and merging each pair to a single node. The *Pool* operation fine-tunes the network with B_{ft} . Both operations (i.e. *Increment* and *Merge*) are performed in the 1st hidden layer. Equations 6, 7 and 8 outline the calculations for ΔInc and ΔMrg ,

$$\Delta = \lambda \exp\left(\frac{-(\nu - \hat{\mu})}{2\sigma^2}\right) |L_c^n - L_c^{n-1}| \quad (6)$$

$$\Delta\text{Inc} = \begin{cases} \Delta; & \text{if } a = \text{Increment} \\ 0; & \text{Otherwise} \end{cases} \quad (7)$$

$$\Delta\text{Mrg} = \begin{cases} \Delta; & \text{if } a = \text{Merge} \\ 0; & \text{Otherwise} \end{cases} \quad (8)$$

Algorithm 2 RA-DAE algorithm

```

1: procedure RA-DAE
2:   define :  $n$  - Current batch ID
3:   Initialise  $Q(s, a) = 0 \forall s \in S, a \in A$ 
4:    $s, a = null$ 
5:   while  $\mathcal{D}_n \neq NULL$  do
6:      $s', a', Q', \Delta\text{Mrg}, \Delta\text{Inc} = \text{GetCtrlParam}(n, Q, s, a)$ 
7:     if  $a' = \text{Pool}$  then
8:       Fine-tune using  $B_{ft}$ 
9:     else if  $a' = \text{Increment}$  then
10:      Add  $\Delta\text{Inc}$  new nodes to the network
11:      Train the  $\Delta\text{Inc}$  nodes greedily using  $B_r$ 
12:     else if  $a' = \text{Merge}$  then
13:      Merge  $2\Delta\text{Mrg}$  nodes into  $\Delta\text{Mrg}$ 
14:     end if
15:     Train the network with  $\mathcal{D}_n$ 
16:      $s = s', a = a', Q = Q'$ 
17:      $n = n + 1$ 
18:   end while
19: end procedure

```

where λ is a coefficient controlling the amount of change, $\hat{\mu}$ and σ are chosen depending on how large or small the network is allowed to grow, and a is the current action chosen by Algorithm 3. We defined ΔMrg and ΔInc as a function of ν_1^n and L_c^n . The objective of Equation 6 is to minimise the error while preventing the network from growing too large or too small. For example, if the error is high, the algorithm increases Δ to reduce the error. If the error has converged, i.e. has not changed for two consecutive batches, Δ will be small.

The need for two pools, B_r and B_{ft} is justified as follows. The pool operation is designed to revise the existing knowledge. Thus, B_{ft} is composed of a diverse set of data batches that differ in the distribution of the data. The objective of the increment operation is to add the most recent features. B_r is ideal for this purpose as it contains the most recent data.

3.3.3 Reward Function

The reward function r^n is defined as,

$$r^n = \begin{cases} e^n - |\hat{\mu} - \nu_1^n| & \text{if } \nu_1^n < V_1 \\ e^n - |\hat{\mu} - \nu_1^n| & \text{if } \nu_1^n > V_2 \\ e^n; & \text{Otherwise,} \end{cases} \quad (9)$$

$$\text{where } e^n = (1 - (L_c^n - L_c^{n-1})) \times (1 - L_c^n) \quad (10)$$

and V_1 and V_2 are predefined thresholds. e^n is specified so that the reward will be higher for lower errors and higher rates of error change (Equation 10). Equation 9 penalises r^n if the network grows too large or too small.

3.4 RA-DAE Algorithm

With S , A and r^n defined, we present the general approach used to solve the MDP (Algorithm 3). Q-Learning was utilised with the following steps,

For the n^{th} iteration, with data batch \mathcal{D}^n ,

1. Until adequate samples are collected (i.e. $n \leq \eta_1$), train with B_r .

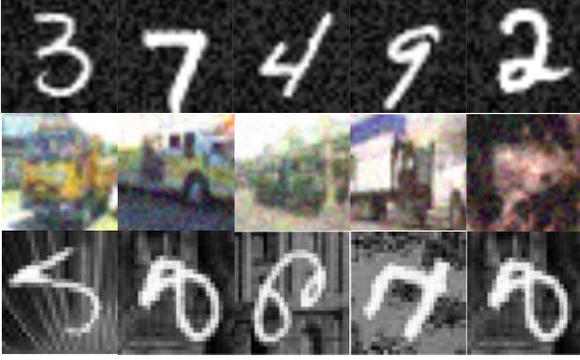


Figure 1. Random examples from the extended MNIST, CIFAR-10 and MNIST-rot-back datasets, respectively

2. With adequate samples collected (i.e. $n > \eta_1$), start calculating Q -values for each state-action pair observed $\{s^n, a^n\}$, where $s^n \in S$, and $a^n \in A$ as defined in Algorithm 3.
3. During $\eta_1 < n \leq \eta_2$, uniformly perform actions from $A = \{\text{Increment}, \text{Merge}, \text{Pool}\}$ to develop a fair utility estimate for all actions in A .
4. With an accurate estimation of Q (i.e. $n > \eta_2$), the best action a' is selected by $a' = \text{argmax}_{a'}(Q(s^n, a'))$ with a controlled amount of exploration (ϵ -greedy).
5. if $a' = \text{Increment}$, calculate ΔInc from Equation 7, add randomly initialised ΔInc nodes and greedily initialise *only* the new nodes with B_r , while keeping the rest constant.
6. if $a' = \text{Merge}$ calculate ΔMrg from Equation 8 and average the closest pairs of ΔMrg nodes to amalgamate $2\Delta\text{Mrg}$ nodes to ΔMrg nodes.
7. if $a' = \text{Pool}$ fine-tune the network with B_{ft} .
8. Train the network with \mathcal{D}^n .
9. Calculate the new state, s^{n+1} (Equation 4) and the reward r^n (Equation 10).
10. Update the Value (Utility) Function $Q(s, a)$ as,

$$Q^{(t+1)}(s^{n-1}, a^{n-1}) = (1-\alpha) \times Q^t(s^{n-1}, a^{n-1}) + \alpha \times q, \quad (11)$$

where $q = r^n + \gamma \times \max_{a'}(Q^t(s^n, a'))$.

3.5 Function Approximation for Continuous Space

For clarity of presentation, Algorithms 2 and 3 assume discrete state space. However, the same algorithms can be extended for continuous state space. The idea is to, for a given action a and an unseen state \tilde{s} , predict the utility value $Q(\tilde{s}, a) = \hat{f}(\tilde{s}, \mathbf{w})$ through function approximation where \hat{f} is the function and \mathbf{w} is the approximated parameter vector [16]. In this paper, Gaussian Process Regression (GPR) [12] with squared exponential kernel, $k_{SE}(x, x') = \sigma^2 \exp(-\frac{(x-x')^2}{2l^2})$ has been used for this regression task. The hyperparameters σ and l are optimised by maximising the marginal likelihood w.r.t. the hyperparameters [12]. Formally, we collect at least $\eta_2 - \eta_1$ observed states and corresponding value pairs $\{s^n, Q(s^n, a^n)\}$. Next, for each $a \in A$, separate curves are fitted with GPR for the $\{s^n, Q(s^n, a^n)\}$ collection of pairs by separating pairs w.r.t. a^n , so that there are $|A|$ curves. Then, for an unseen state \tilde{s} and a given action a' , $Q(\tilde{s}, a')$ is calculated using the curve fitted for action a' . The continuous space is preferred as it provides a detailed representation of the environ-

Algorithm 3 Control Parameter Calculation algorithm

```

1: procedure GETCTRLPARAM( $n, Q, s, a$ )
2:   define :  $n$  - Current batch ID
3:   define :  $Q$  - Utility function
4:   define :  $s, a$  - Previous state, action
5:   define :  $\gamma$  - Discount rate
6:   define :  $\alpha$  - Learning rate
7:   if  $n < \eta_1$  then
8:     return  $null, Pool, Q, 0, 0$ 
9:   end if
10:  Calculate current state,  $s'$  (Equation 4)
11:  if  $s, a \neq null$  then
12:     $q = r^n + \gamma \times \max_{a'}(Q(s', a'))$ 
13:     $Q(s, a) = (1 - \alpha) \times Q(s, a) + \alpha \times q$ 
14:  end if
15:  if  $n < \eta_2$  then
16:    Evenly chose action  $a'$  from  $A$ 
17:  else
18:    Explore with  $\epsilon$ -greedy ( $\epsilon = 0.1$ )
19:    OR
20:     $a' = \text{argmax}_{a'}(Q(s', a'))$ 
21:  end if
22:  Calculate  $\Delta\text{Mrg}$  and  $\Delta\text{Inc}$  (Eq. 7 and 8)
23:  return  $s', a', Q, \Delta\text{Mrg}, \Delta\text{Inc}$ 
24: end procedure

```

ment with fewer variables, as opposed to the discrete space. This is sensible as the information extracted is continuous (e.g. L_g, L_c, ν).

3.6 Summary

Our proposed solution is detailed in Algorithm 2 and can be seen is a repeated application of Algorithm 3. For each batch of data \mathcal{D}^n , the state s^{n+1} and reward r^n is calculated using Equations 4 and 10 respectively. Next, the best action a' for the new state is retrieved by $a' = \text{argmax}_{a'}(Q(s^{n+1}, a'))$. To calculate $Q(s^{n+1}, \hat{a})$ for some action $\hat{a} \in A$, GPR is employed as explained in Section 3.5. Next the action a' is performed. Then the network is fine-tuned using \mathcal{D}^n . This process is repeated until the end of the data stream.

4 Experiments

4.1 Overview and Setup

The experiments are based on extended versions of three datasets (i.e. MNIST², MNIST-rot-back³ and CIFAR-10⁴). Random samples from each dataset are depicted in Figure 1. The extended versions of each dataset consist of 1,000,000 examples. Examples were masked with noise during the generation to make them unique. We generated non-stationary distributions for each dataset using Gaussian processes (GP) [12], simulating the covariate shift effect. Formally, the ratio for each class of labels is generated using $ratio_k(t) = \frac{\exp\{a_k(t)\}}{\sum_{j=1}^K \exp\{a_j(t)\}}$ where $a_k(t)$ is a random curve generated by the GP.

Experiments were conducted with three different types of deep architectures; SDAE (Standard Denoising Autoencoders), MI-DAE (Merge-Incremental Denoising Autoencoders) and RA-DAE (our approach). For MI-DAE, we used a modified "update rule I" introduced

² <http://yann.lecun.com/exdb/mnist/>

³ www.iro.umontreal.ca/lisa/twiki/bin/view.cgi/Public/MnistVariations

⁴ <http://www.cs.toronto.edu/kriz/cifar.html>

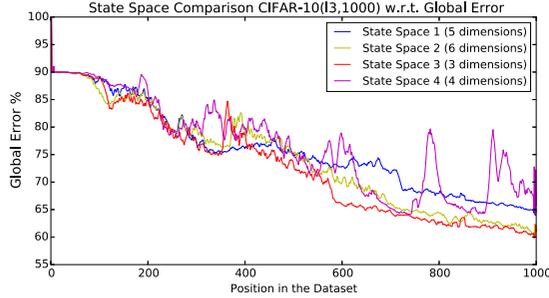


Figure 2. Analysis of Global Error E_{glb} for different state spaces for CIFAR-10 and a network with 3 layers with 1000 neurons in each. The mathematical definitions of state spaces 1,2,3 and 4 can be found in Section 4.2.1. It is clear that State Space 4 shows a steeper reduction of error compared to its counterparts.

in [20] as they claim the performance is fairly robust to different update rules as follows,

$$\Delta N_{t+1} = \begin{cases} \Delta N_t + 30; & , \frac{e_t}{e_{t-1}} < (1 - \epsilon_1) \\ \Delta N_t / 2; & , \frac{e_t}{e_{t-1}} > (1 - \epsilon_2) \\ \Delta N_t, & \text{Otherwise} \end{cases}$$

$\Delta Mrg = \lceil \gamma_{ratio} \Delta Inc \rceil$; for $\gamma_{ratio} = 0.2$, as these modifications produced better performance.

Several initial layer configurations (hidden layer sizes) were used, as outlined in Table 2. To refer to a certain algorithm, we use the following notation. We use the superscript for the number of layers and the subscript to indicate the size of each layer. For example, $SDAE_{1500}^{I3}$ denotes a SDAE with three layers and 1500 nodes in each layer. The configurations in Table 2 maximise the performance of the algorithms tested. The continuous state space (Equation 4) was used for all the experiments. We define two error measures for evaluating performance. A local error $E_{lcl} = L_c^{n+1}$, measured on a validation set, \mathcal{D}^{n+1} (batch succeeding the current batch) and a global error $E_{glb} = \frac{\sum_{i \in \mathcal{D}_{test}} L_c^i}{|\mathcal{D}_{test}|}$ s.t. $\mathcal{D}^i \in \mathcal{D}_{test}$ measured on an unseen independent test set \mathcal{D}_{test} , which contains an approximate uniform distribution of all the classes. These two sets of data enable us to respectively, evaluate how the network preserve immediate past knowledge and the globally accumulated knowledge.

All experiments were carried out using a Nvidia GeForce GTX TITAN GPU and Theano⁵. For all experiments we used 20% corruption level, 0.2 learning rate, batch size of 1000. We empirically chose $\gamma = 0.9$ (Equation 11) m (Equation 4) 30, and η_1 and η_2 (Algorithm 3) to be 30 and 60 respectively. Λ (Equation 3) was selected as 0.7 and 0.995 for non-stationary and stationary experiments respectively. $\tau = 10,000$ (for B_r , B_{ft} and B) was chosen from a set of sizes {1000, 5000, 10000} as 10,000 produced the best results. Results are depicted in Figure 3.

4.2 Results

4.2.1 Evaluation of State Spaces

As mentioned in Section 3.3.1 the state space was chosen while paying close attention to the performance against an unseen data batch, difference between observed data distributions and complexity of the network. We utilised various quantifiable measures. L_g^{n+1}

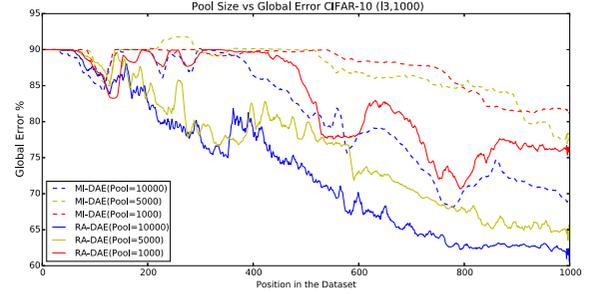


Figure 3. Performance of RA-DAE and MI-DAE for different pool sizes. Pool size of 10000 yielded the best results. It can be seen that RA-DAE with a pool size of 1000 performs similarly to MI-DAE with pool size 10000. This can be attributed to the learned policy and the pooling technique.

Table 2. Initial layer configurations for different datasets. The superscript of the algorithm name specifies the number of layers and the subscript indicates the size of each layer.

MNIST	CIFAR-10	MNIST-rot-back
$SDAE_{500}^{I1}$	$SDAE_{1000}^{I1}$	$SDAE_{1500}^{I1}$
$SDAE_{500}^{I3}$	$SDAE_{1000}^{I3}$	$SDAE_{1500}^{I3}$
$MI-DAE_{500}^{I1}$	$MI-DAE_{1000}^{I1}$	$MI-DAE_{1500}^{I1}$
$MI-DAE_{500}^{I3}$	$MI-DAE_{1000}^{I3}$	$MI-DAE_{1500}^{I3}$
$RA-DAE_{500}^{I1}$	$RA-DAE_{1000}^{I1}$	$RA-DAE_{1500}^{I1}$
$RA-DAE_{500}^{I3}$	$RA-DAE_{1000}^{I3}$	$RA-DAE_{1500}^{I3}$

and L_c^{n+1} were employed to evaluate RA-DAE's ability to classify an unseen batch of data (i.e. \mathcal{D}^{n+1}). Kullback-Leibler divergence ($D_{KL}(P^n || Q^n)$) [8] was used to measure the divergence between the distribution of current data and previously fed data; $D_{KL}(P^n || Q^n) = \sum_i^K P^n(i) \log(\frac{P^n(i)}{Q^n(i)})$, where $P^n(i) = \frac{Count_i^n}{p}$, $Count_i^n$ is the number of data points with class i in \mathcal{D}^n , p is as defined in Table 1 and $Q^n(i) = \frac{\sum_{j=1}^m P^j(i)}{m}$. Finally the complexity of RA-DAE at a given time is captured by ν^n .

With the aforementioned quantities defined, the following state spaces were defined:

- State Space 1 - $\{\tilde{L}_g(m_3), \tilde{L}_c(m_1), \tilde{L}_c(m_2), \tilde{L}_c(m_3), \nu\}$
- State Space 2 - $\{\tilde{L}_g(m_3), \tilde{L}_c(m_1), \tilde{L}_c(m_2), \tilde{L}_c(m_3), \nu, D_{KL}(P^n || Q^n)\}$
- State Space 3 - $\{\tilde{L}_g(m), \tilde{L}_c(m), \nu\}$
- State Space 4 - $\{\tilde{L}_g(m), \tilde{L}_c(m), \nu, D_{KL}(P^n || Q^n)\}$

The constants m_1, m_2, m_3 and m were chosen empirically and set to 5,15,30 and 30 respectively. The reason for calculating \tilde{L} for several m values is to learn whether augmenting the state space of L_g and L_c contribute additional information. However, from the experimental results, it was evident that a simpler state space yields the best results. Furthermore, it was surprising to verify that $D_{KL}(P^n || Q^n)$ had no significant positive impact on the results. The performance of different state spaces is depicted in Figure 2.

4.2.2 Analysis of Structure Adaptation

We studied the adaptation pattern of RA-DAE and MI-DAE in both stationary and non-stationary environments. Figure 5(c) depicts the number of nodes in the first layer for MI-DAE and RA-DAE as they adapt to data distributions changes with the CIFAR-10 dataset. In non-stationary problems, RA-DAE exhibits repeated peaks in the number of nodes. This can be explained by the changes in class distribution in Figure 5(f). Node number changes in Figure 5(c) align with

⁵ <http://deeplearning.net/software/theano/>

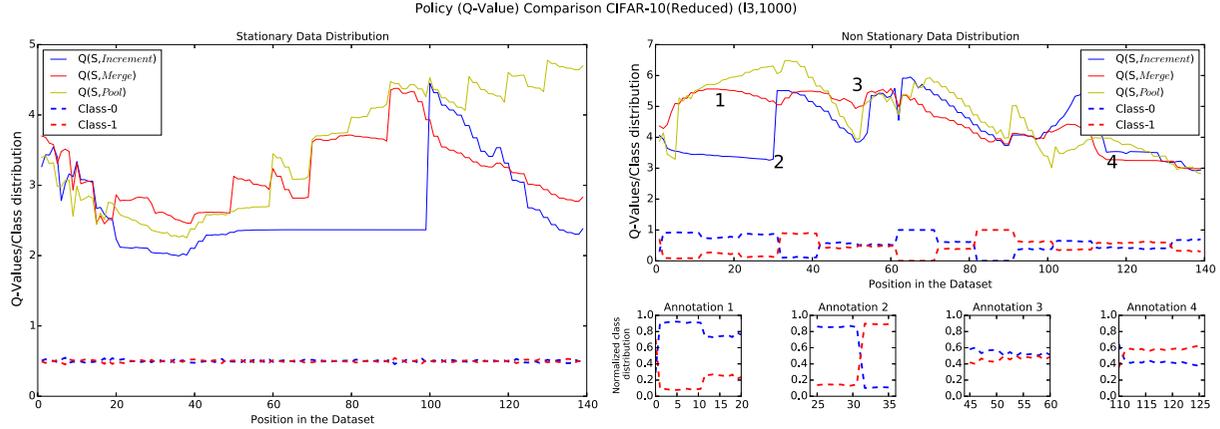


Figure 4. Visualisation of the value function ($Q(s, a)$) evolution for stationary and non-stationary distributions. Annotations on the top-right graph indicate note-worthy behaviours of the value function. The left and top-right graphs depict the complete progression of the data distribution. For clarity, a reduced version of CIFAR-10 with only two classes and 200,000 examples was used. For stationary data distribution, the graph indicates how *Pool* and *Merge* operations dominate the behaviour as there are no significant data distribution changes. In the non-stationary setting, the value function for action *Increment* surges in face of a sharp distribution change (Annotation 2). *Merge* and *Pool* operations take over when data distributions are consistent (Annotation 3 and 4 respectively).

the peaks appearing for various class distributions in Figure 5(f). For sharp distribution changes, RA-DAE quickly increases the number of neurons. However, MI-DAE shows a moderate growth in the number of nodes, despite the rapid changes in the data distribution. This demonstrates that RA-DAE is more responsive than MI-DAE in adapting the architecture in the face of changes. For a stationary data distribution, MI-DAE shows a constant node count after the first few hundred batches, where RA-DAE increases the number of nodes over time. This can be attributed to the fact that reducing the number of neurons tends to increase the error, occasionally making the reduction operation not preferable to RA-DAE. This is acceptable as RA-DAE will not increase nodes unnecessarily as it would lead to poor results due to *overfitting*. An alternative is to perform the pool operation after reduce, which would reduce the error at an increased computational cost.

4.2.3 Analysis of Local and Global Error

Finally, the capability to preserve past knowledge, balancing immediate and global rewards for the algorithms was assessed by using the local error, E_{lcl} , and the global error, E_{glb} . We used the hybrid objective function ($L_{disc} + \lambda L_{gen}$ for $\lambda = 0.2$) [19] to fine-tune the network.

Figure 5 depicts several interesting results. Figure 5(a) illustrates the behaviour of the E_{lcl} . RA-DAE^{l1}₅₀₀ shows a clear improvement w.r.t E_{lcl} over time. Note how in RA-DAE^{l1}₅₀₀ the fluctuations shrink over time. Moreover, Figure 5(d) delineates a significant E_{glb} error margin maintained by RA-DAE^{l1}₅₀₀ compared to SDAE^{l1}₅₀₀ and MI-DAE^{l1}₅₀₀. RA-DAE’s ability to grow the network faster compared to MI-DAE explains this significantly lower error. Figure 5(b) and (e) portray the performance of the algorithm in a stationary environment (CIFAR-10). Though we expected all algorithms to perform comparably well in the stationary environment, RA-DAE^{l3}₁₀₀₀ achieves the lowest E_{lcl} and E_{glb} and the steepest error reduction. Both RA-DAE^{l3}₁₀₀₀ and MI-DAE^{l3}₁₀₀₀ demonstrate better performance than SDAE^{l3}₁₀₀₀. This highlights that structure adaptation strategies enhance the performance of deep networks in both stationary and non-stationary environments.

Table 3 summarises the errors (mean±standard deviation of the last 250 batches) for various datasets. The number 250 was chosen,

as the last 250 batches displayed a consistent performance in most instances. There are several key observations from Table 3. First, RA-DAE^{l3} has outperformed its counterparts in both stationary and non-stationary scenarios, where RA-DAE^{l1} and MI-DAE^{l1} have performed equally well. By observing the performance of RA-DAE^{l1} and RA-DAE^{l3} it is evident that the performance of RA-DAE has improved as the network becomes deeper. MI-DAE has exhibited the same property in most occasions. The rationale being, not only deep networks are more robust to structural modifications in terms of error, but also they are able to learn more descriptive representations as depth increases. However, performance of SDAE^{l3} is worse than SDAE^{l1} in both cases. This observation justifies the need for better techniques to leverage deep architectures in online scenarios.

A surprising observation can be made in {SDAE,MI-DAE,RA-DAE}^{l1} for MNIST-rot-back. Even though we expected RA-DAE to perform the best, SDAE^{l1} shows the best performance with a $52.8 \pm 7.0\%$ and $65.7 \pm 2.7\%$ for E_{lcl} and E_{glb} respectively. Close examination of the behaviours of E_{lcl} and E_{glb} of SDAE, MI-DAE and RA-DAE, shows that MI-DAE and RA-DAE do not perform as well as SDAE. This is due to the fluctuation of E_{lcl} being fast, which causes the algorithm to increase the number of nodes unnecessarily. Consequently, MI-DAE and RA-DAE lead to poor accuracy due to *overfitting*. This issue alleviates as the network becomes deeper.

4.2.4 Analysis of the Policy Learnt

In order to analyse the policy learnt by RA-DAE, it is imperative to take a close look at the value function (i.e. $Q(s, a)$) learnt by RA-DAE. Figure 4 depicts the evolution of the value function over time with note-worthy behaviours annotated. For the purpose of visualisation, a simplified version of CIFAR-10 dataset (CIFAR-10-bin) has been used. CIFAR-10-bin comprises only two classes and has a total of 200,000 data points. Figure 4 depicts the value function for two settings; stationary and non stationary. The annotation graphs at top-right highlight the changes in data distribution at the points of interest in the top graph.

In the stationary setting, it can be seen that *Pool* and *Merge* operations have dominated the policy, Figure 4(right). This is sensible as the data distribution stays constant throughout and a necessity to increase the number of nodes hardly emerges.

Table 3. This table presents the E_{lcl} and E_{glb} obtained for various datasets and depths. Errors are in the format of mean \pm standard deviation for the last 250 batches. The lowest errors are highlighted in bold. RA-DAE has shown the best performance (smallest local and global errors) in most occasions (for both stationary and non-stationary).

	MNIST		CIFAR-10		MNIST-rot-back		CIFAR-10 (Stationary)	
	$E_{lcl}\%$	$E_{glb}\%$	$E_{lcl}\%$	$E_{glb}\%$	$E_{lcl}\%$	$E_{glb}\%$	$E_{lcl}\%$	$E_{glb}\%$
SDAE ^{I1}	10.9 \pm 5.8	27.2 \pm 5.7	65.9 \pm 4.9	82.8 \pm 1.1	52.8 \pm 7.0	65.7 \pm 2.7	67.9 \pm 1.5	70.2 \pm 0.6
MI-DAE ^{I1}	6.4 \pm 3.2	23.9 \pm 4.4	50.4 \pm 4.7	74.9 \pm 3.0	61.8 \pm 9.0	72.0 \pm 2.6	55.5 \pm 1.9	61.4 \pm 0.8
RA-DAE ^{I1}	5.1 \pm 1.4	11.3 \pm 0.7	50.6 \pm 7.2	74.0 \pm 2.4	62.3 \pm 8.8	69.6 \pm 2.8	59.8 \pm 1.9	61.9 \pm 1.1
SDAE ^{I3}	11.2 \pm 5.9	31.6 \pm 5.4	76.3 \pm 6.6	88.4 \pm 1.9	67.6 \pm 8.9	77.1 \pm 3.2	71.8 \pm 1.4	72.7 \pm 0.7
MI-DAE ^{I3}	5.4 \pm 4.4	31.3 \pm 4.0	43.7 \pm 8.5	71.0 \pm 1.6	56.0 \pm 9.2	65.5 \pm 2.1	56.1 \pm 1.8	58.9 \pm 1.1
RA-DAE ^{I3}	4.1 \pm 3.0	13.4 \pm 0.1	32.4 \pm 8.0	62.7 \pm 0.7	48.2 \pm 9.2	60.6 \pm 3.4	50.6 \pm 2.1	53.6 \pm 2.1

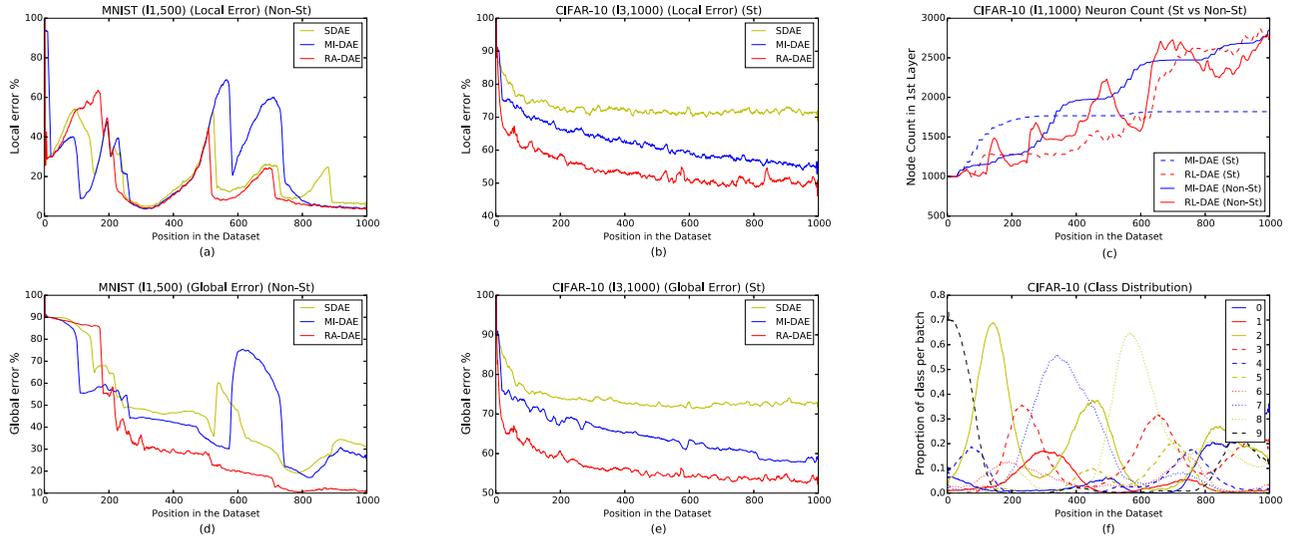


Figure 5. (a) and (d) show the behaviour of E_{lcl} and E_{glb} in a non-stationary (Non-St) situation, where (b) and (e) show the behaviour of E_{lcl} and E_{glb} in a stationary (St) situation. The titles consist of the name of the dataset followed by number of hidden layers and neuron count in each layer, within parenthesis. RA-DAE exhibits the lowest E_{lcl} and E_{glb} at the end, and a more consistent reduction compared to SDAE and MI-DAE. (c) presents node adaptation patterns of MI-DAE and RA-DAE in both stationary and non-stationary situations. (f) shows the class distribution of data over time and each curve denotes a single class. By comparing to (f), (c) clearly indicates that RA-DAE is more sensitive to changes in data distribution than MI-DAE in terms of the neuron adaptation. The horizontal axis represents the number of batches in the training dataset.

For the non-stationary setting, it can be seen how *Pool* and *Merge* operations have a high value as the algorithm has not seen an significant data distributions, thus suppressing *Increment* operation. Next, at annotation 2 it can be seen how the value of *Increment* operation boosts up due to the massive data distribution change. Then, at point 3, *Merge* operation takes over as data distribution is somewhat consistent. And finally, at point 4, *Pool* operation dominates the graph due to the consistency of the distribution of data.

5 Conclusion

Online learning can be widely beneficial for deep architectures as it allows network adaptation for streaming data problems. However, defining the structure of the network, including number of nodes, can be difficult to do in advance. To address this, [19] introduces MI-DAE which can dynamically change the structure of the network but relies on simple heuristics. The novelty of this work is an online learning stacked denoising autoencoder which leverages reinforcement learning to modify the structure of the deep network. In this, we use a model-free reinforcement learning approach and calculate a utility function for actions by sampling from the incoming states.

Compared to the counterpart, our approach is more principled and responsive in adapting to new information. The method leverages RL to make decisions in a dynamic fashion. The control behaviour combined with powerful pooling techniques allows our approach to preserve past-knowledge effectively. Finally, our solution make decisions based on long-term versus immediate reward. Experimental results indicate that our solution often outperforms its counterparts with a lower classification error, and the performance improves as the network becomes deeper. Also, the approach is more sensitive to changes in the data distribution. Future work will address other deep learning architectures such as convolutional neural nets and deep Boltzmann machines.

Acknowledgements

This research was supported by funding from the Faculty of Engineering & Information Technologies, The University of Sydney, under the Faculty Research Cluster Program. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU used for this research.

REFERENCES

- [1] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al., 'Greedy layer-wise training of deep networks', *Advances in neural information processing systems*, **19**, 153, (2007).
- [2] Dan Cireşan, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber, 'Multi-column deep neural network for traffic sign classification', *Neural Networks*, **32**, 333–338, (2012).
- [3] Li Deng and Dong Yu, 'Deep learning: Methods and applications', Technical Report MSR-TR-2014-21, (May 2014).
- [4] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., 'Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups', *Signal Processing Magazine, IEEE*, **29**(6), 82–97, (2012).
- [5] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh, 'A fast learning algorithm for deep belief nets', *Neural computation*, **18**(7), 1527–1554, (2006).
- [6] Geoffrey Hinton and Ruslan Salakhutdinov, 'Reducing the dimensionality of data with neural networks', *Science*, **313**(5786), 504–507, (2006).
- [7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 'Imagenet classification with deep convolutional neural networks', in *Advances in neural information processing systems*, pp. 1097–1105, (2012).
- [8] Solomon Kullback and Richard A Leibler, 'On information and sufficiency', *The annals of mathematical statistics*, **22**(1), 79–86, (1951).
- [9] Xinwang Liu, Guomin Zhang, Yubin Zhan, and En Zhu, 'An incremental feature learning algorithm based on least square support vector machine', in *Frontiers in Algorithmics*, 330–338, Springer, (2008).
- [10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, 'Playing atari with deep reinforcement learning', *arXiv preprint arXiv:1312.5602*, (2013).
- [11] Tomaso Poggio and Gert Cauwenberghs, 'Incremental and decremental support vector machine learning', *Advances in neural information processing systems*, **13**, 409, (2001).
- [12] Carl Edward Rasmussen, 'Gaussian processes for machine learning', (2006).
- [13] Martin Riedmiller, 'Neural fitted q iteration—first experiences with a data efficient neural reinforcement learning method', in *Machine Learning: ECML 2005*, 317–328, Springer, (2005).
- [14] Kenneth O Stanley and Risto Miikkulainen, 'Evolving neural networks through augmenting topologies', *Evolutionary computation*, **10**(2), 99–127, (2002).
- [15] Masashi Sugiyama and Motoaki Kawanabe, *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*, MIT Press, 2012.
- [16] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, volume 1, MIT press Cambridge, 1998.
- [17] Sebastian Thrun and Lorien Pratt, *Learning to learn*, Springer Science & Business Media, 2012.
- [18] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, 'Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion', *The Journal of Machine Learning Research*, **11**, 3371–3408, (2010).
- [19] Guanyu Zhou, Kihyuk Sohn, and Honglak Lee, 'Online incremental feature learning with denoising autoencoders', in *International Conference on Artificial Intelligence and Statistics*, pp. 1453–1461, (2012).
- [20] Guanyu Zhou, Kihyuk Sohn, and Honglak Lee, 'Supplementary material: Online incremental feature learning with denoising autoencoders', in *International Conference on Artificial Intelligence and Statistics*, (2012).

Multi-Class Probabilistic Active Learning

Daniel Kottke¹ and Georg Krempf¹ and
Dominik Lang² and Johannes Teschner² and Myra Spiliopoulou³

Abstract. This work addresses active learning for multi-class classification. Active learning algorithms optimize classifier performance by successively selecting the most beneficial instances from a pool of unlabeled instances to be labeled by an oracle. In this work, we study the influence of the following factors for active learning: (1) an instance’s impact, (2) its posterior, and (3) the reliability of this posterior. To do so, we propose a new decision-theoretic approach, called multi-class probabilistic active learning (McPAL). Building on a probabilistic active learning framework, our approach is non-myopic, fast, and optimizes a performance measure (like accuracy) directly. Considering all influence factors, McPAL determines the expected gain in performance to compare the usefulness of instances. For this purpose, it calculates the density weighted expectation over the true posterior and over all possible labeling combinations in a closed-form solution. Thus, in contrast to other multi-class algorithms, it considers the posterior’s reliability which improved the performance. In our experimental evaluation, we show that the combination of the selected influence factors works best and that McPAL is superior in comparison to various other multi-class active learning algorithms on six datasets.

1 INTRODUCTION

In supervised classification, prediction models are learned from labeled training data. In some applications, unlabeled data is available or easy to collect but the labeling (annotation) of this data is expensive, time-consuming or exhausting. For such applications, active learning methods provide solutions that optimize the labeling process by selecting the most useful unlabeled instances to be passed to an oracle for labeling. Thereby, active learning aims to achieve high performance with as few labeled instances as possible [23].

A particular and little researched challenge [26] in active learning is its generalization to multi-class settings, with multinomial rather than binary labels. The few works that have addressed this task so far mostly use either uncertainty sampling for active learning with support vector machines, thereby concentrating on instances close to the anticipated decision boundary [6, 12, 29], optionally extended by information about density or diversity [4, 14]. Others use expected error reduction by simulating the impact of a label acquisition on the whole dataset to determine the expected performance [13]. Both approaches have known limitations [7, 15]: the former fast, information-theoretic heuristic often fails in exploring the dataspace, the latter decision-theoretic method has high computation time.

We contribute a multi-class active learning approach that combines the advantages of the approaches mentioned above, i.e. optimizing expected performance directly while being nearly as fast as uncertainty sampling. Following the recently proposed probabilistic active learning framework [17], the key idea is to compute the expectation over the true posterior by incorporating the number of labels in a neighborhood of the label candidate as a proxy for the posterior’s reliability. The resulting score is weighted with the density which we use as a proxy for the new label’s impact on the whole dataset. We compare our approach with the most relevant state-of-the-art methods from the literature and present experiments on six datasets.

In addition, we expose the three influence factors that are used in our method: the posterior, the reliability of that posterior, and the impact of a labeling candidate. We explain their role in active learning and evaluate their effect experimentally. To the best of our knowledge, we are the first that use the number of labels inside a candidate’s neighborhood for multi-class active learning, which we show to have a strong impact on the learner’s performance. Furthermore, by adding another decision-theoretic method to propositions in the comparative study of [14], we contribute to the important research question on how to combine the posteriors of many classes into one comparable score.

The next section summarizes the related work by introducing the basic approaches of multi-class active learning. The main section presents our new approach including an analysis of its characteristics, and is followed by our experimental evaluation. The paper is concluded with a summarizing discussion.

2 RELATED WORK

Active learning aims to optimize the annotation of unlabeled instances (candidates), by selecting the ones that improve a given classifier’s performance the most [23]. As active learning in general is far more researched than multi-class active learning, we concentrate on the most relevant work before summarizing multi-class approaches.

Most active learning techniques define a usefulness score for each label candidate. A simple but common information-theoretic heuristic is to use the instances with highest uncertainty [18]. This uncertainty sampling method chooses instances near the classifier’s current decision boundary, i.e. instances with a posterior probability near the decision threshold (for binary cases 0.5). Related approaches like using the posteriors’ entropy have been addressed in [23]. In contrast, the decision-theoretic expected error reduction approach estimates a candidate’s usefulness by simulating its label’s realizations and measuring the resulting model’s performance on a representative set of evaluation instances [21]. This computationally expensive calculation of the expected performance over all possible labels and the instances of the representative set builds the usefulness score [3].

¹ Knowledge Management and Discovery Lab, Otto von Guericke University, Magdeburg, Germany, email: {daniel.kottke, georg.krempf}@ovgu.de

² Faculty of Computer Science, Otto von Guericke University, Magdeburg, Germany, email: {dominik.lang, johannes.teschner}@st.ovgu.de

³ Knowledge Management and Discovery Lab, Otto von Guericke University, Magdeburg, Germany, email: myra@iti.cs.uni-magdeburg.de

Kreml et al. [16] argue that using posterior estimates directly in the expectation step leads to inaccuracies. They observed that these posterior estimates are highly unreliable especially having only few labeled instances. Probabilistic active learning [17] therefore tries to overcome these difficulties by introducing label statistics that include the posterior of the positive class (they only consider binary classification tasks) and the number of nearby labels as a proxy for reliability. The usefulness score is calculated with the expectation over the true posterior as well as over the possibly appearing labels. Other approaches aim to reduce the classification variance by using an ensemble of classifiers and request instances where the ensemble’s disagreement is high [24].

For active learning with multiple classes, the main challenge is the mapping of posterior values into a comparable score to select the most useful labeling candidate. Körner and Wrobel [14] analyzed different heuristics that have been also used by other papers: (1) usual confidence-based uncertainty sampling chooses the instance with the lowest posterior for the best decision, which is comparable to selecting the instances near the decision boundary (see also [5, 11, 28, 29]), (2) entropy-based sampling chooses the instance with highest posterior entropy (see also [30]), (3) Best-vs-Second-Best (BvsSB) sampling (also called margin-based) uses the difference between the posterior of the best and the second best class (see also [5, 12]), and (4) sampling using a specific disagreement that combines margin-based disagreement with the maximal probability⁴.

Expected error reduction-based methods have also been considered for multi-class active learning. Joshi et al. [13] proposed an algorithm called *Value of Information (VoI)* that estimates the expected misclassification costs plus the expected labeling costs. They compare the performance of the current classifier and each hypothetical classifier which are evaluated for each labeling candidate and each class on an evaluation set. As these algorithms take long for execution, the authors propose three approximations for speedup. For music annotation applications, Chen et al. [4] developed a method that finds a set of instances to be labeled based on a volume criterion (similar to SVM volume reduction [25]), a density score that favors dense regions and a diversity score that enforces diversity among instances from the labeling set. More recently, Guo and Wang [6] developed a stepwise method consisting of an initial selection of instances to be labeled (via random, clustering or discrepancy), followed by an active learning step. This is based on the characteristics of One-versus-Rest (OvR) Support Vector Machines (SVMs) where a labeling candidate can belong to one class with support from zero, one or more than one OvR SVMs. To choose the next instance for labeling, they define a rejection score, a compatibility score and an uncertainty score, and propose rules on how these score have to be considered. Wang et al. [26] propose an ambiguity-based multi-class approach that uses possibilistic membership from One-vs-Rest SVMs. These membership values are between 0 and 1 but do not necessarily sum up to one like posteriors. Their ambiguity measure is based on fuzzy logic operations and has a parameter γ which has to be optimized and is not known in advance. A more theoretical work on cost-sensitive multi-class active learning is given by [1]. He analyzed the regret and label complexity for data with labels that are generated with a generalized linear model.

Some approaches consider settings with different costs for misclassifying an instance of a specific class [5, 13]. Additionally, [13] also includes annotation cost, i.e. the cost of labeling one instance.

⁴ Note, that the selection of instance based on confidence and BvsSB would be exactly the same in a two-class problem but is different for multiple classes (see [23]).

The acquisition of instances can be done in a successive manner or in form of instance batches. Most approaches choose to acquire instances one-by-one, except for [4, 30]. Besides SVMs (often used with a probabilistic version), [14] used an ensemble of trees, [11] proposed a probabilistic version of the k-nearest-neighbor (pKNN) classifier, [5] tested their algorithms on a random forest, and [30] used random walks over a markov chain.

3 OUR METHOD

In this section, we propose probabilistic active learning for multiple classes, an extension of the binary version proposed in [16]. In the first subsection, we present the active learning framework and explain our influence factors. Next, we propose our **Multi-class Probabilistic Active Learning (McPAL)** approach, followed by the derivation of a closed-form solution. Finally, we conclude our results and compare its behavior to existing approaches in an analytical way.

3.1 AL framework and influence factors

In an active, multi-class classification tasks with C different classes, each instance has a feature vector \vec{x} and a label $y \in \{1, \dots, C\}$, which is unknown at the beginning. As shown in Fig. 1, the active learner successively selects the most useful instances \vec{x}^* from the candidate pool \mathcal{U} and requests its label y from the oracle. After re-training the classifier with the new labeled set $\mathcal{L} \cup (\vec{x}^*, y)$, this procedure is repeated until the budget b is consumed. In our setting, the active component’s decision is based on outputs (posteriors and distribution of labeled instances) of a generative probabilistic classifier [19], which is updated according to the contents of \mathcal{L} .

```
function al_framework(U) {
  L = {}
  cl = init_classifier()
  for(i=1; i<=b; i++){
    x* = active_learning(U, cl, L)
    y = ask_oracle(x*)
    U = remove(U, {x*})
    L = append(L, {x*, y})
    cl = train_classifier(L)
  }
}
```

Figure 1. Pseudocode of the active learning framework

Throughout our research on active learning, we identified different influence factors that affect active learning positively. The labeling candidate’s *class posterior* $\hat{P}(y | \vec{x})$ is the most commonly used one, as it indicates the probability of an instance \vec{x} to be classified as y . For simplicity, we denote \vec{p} as the vector of estimated posterior probabilities, i.e. $\hat{p}_i = \hat{P}(y = i | \vec{x})$, $1 \leq i \leq C$. If the posteriors for all classes are similar, this indicates a high uncertainty of the classifier at the instance’s location \vec{x} . Here, we have to distinguish between the aleatoric uncertainty that is caused by high Bayesian error, and the epistemic uncertainty, which is caused by a lack of information [22]. We are not able to reduce the aleatoric uncertainty, but we can acquire more labels to reduce the epistemic uncertainty in the currently considered neighborhood.

Measuring the number of nearby labels n as a proxy for the *reliability of the class posterior* enables the separation of the aleatoric and the epistemic uncertainty. The higher this number is, the more

likely it is for the observed posterior \hat{p} to be close to the unknown true posterior.

The third influence factor is the *impact on the whole dataset*. Weighting the usefulness score by the instances' density as a proxy for its impact prefers instances in dense regions over those in sparse ones. We assume that it is more beneficial to focus on regions with high density as more future classification decision benefit from the information increment there.

One of the most important questions in multi-class active learning is how to combine the different posteriors to one comparable score [14]. In binary situations, this function $\hat{p} \mapsto \mathbb{R}$ is only one-dimensional as $\hat{p}_2 = 1 - \hat{p}_1$ and can be easily visualized. Three-class problems typically are visualized with ternary plots (see also [14, 23]). In Fig. 2, we show a ternary heatmap plot where the darker shades indicate higher usefulness. This is a barycentric coordinate system, where each position stands for one specific posterior probability. The figure shows the usefulness values for confidence-based sampling (Conf), and for the Best-vs-Second-Best (BvsSB) approach. The entropy-based score has a more circular shape (not shown here) [23].

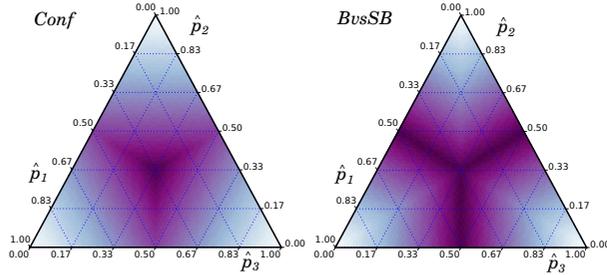


Figure 2. Ternary heatmap plot of the usefulness of confidence-based (Conf) and Best-vs-Second-Best (BvsSB) sampling. Dark color indicates high usefulness of a posterior in that barycentric coordinate system.

In the next section, we propose our method, which combines all three influence factors in a decision-theoretic way. Then, we visualize the behavior of McPAL (without the density weight) also with ternary plots, and evaluate our theory of influence factors experimentally comparing their effects on active learning performance in Sec. 4.2. Our mathematical symbols are summarized in Tab. 1.⁵

C	- Number of classes
$Y = \{1, \dots, C\}$	- Vector of all possible labels
\mathcal{L}	- Set of labeled instances (x, y)
\mathcal{U}	- Set of unlabeled instances (x, \cdot)
$\vec{p} = (p_1, \dots, p_C)$	- Vector of true posteriors
$\vec{k} = (k_1, \dots, k_C)$	- Vector of frequency estimates
$n = \sum k_i$	- Number of observed labels (reliability)
$\hat{\vec{p}} = \vec{k}/n$	- Vector of observed posteriors
$\vec{d} = (d_1, \dots, d_C)$	- Decision vector (see Eq. 8)
$m \in \mathbb{N}$	- Number of hypothetically considered labels
$\vec{l} = (l_1, \dots, l_C) \in \mathbb{N}^C$	- Vector representing the number of hypothetically labels per class ($\sum l_i = m$)

Table 1. Overview of used mathematical symbols.

3.2 Multi-class probabilistic active learning

In probabilistic active learning for two classes, it is assumed that the appearance of a label of class y is a Bernoulli experiment [17]. A label of class i in the neighborhood of an instance \vec{x} appears with

a probability of $P(y = i | \vec{x}) =: p_i$ building the vector of true posteriors \vec{p} . For multiple classes, we naturally generalize the 2-class Binomial distribution to a Multinomial one. The probability of observing a specific labeling situation \vec{k} given the true posterior \vec{p} is then calculated according to Eq. 1. Each entry k_i in the vector \vec{k} represents the number of instances with label i , $1 \leq i \leq C$ in the neighborhood of \vec{x} . This vector also indicates the number of observed labels $n = \sum k_i$, which is used as the reliability proxy ($\vec{k} = n \cdot \vec{p}$). We use the generalized multinomial coefficient for non-integer arguments containing the Γ function by Legendre [20].

$$P(\vec{k} | \vec{p}) = \text{Multinomial}_{\vec{p}}(\vec{k}) = \binom{\sum k_i}{k_1, \dots, k_C} \cdot \prod (p_i^{k_i}) \quad (1)$$

$$= \frac{\Gamma((\sum k_i) + 1)}{\prod (\Gamma(k_i + 1))} \cdot \prod (p_i^{k_i}) \quad (2)$$

In the active learning setting, we do not know the true posteriors \vec{p} , but we are able to estimate the number of observations \vec{k} . To determine a probability distribution for the true posterior, we take the normalized likelihood function [16] as given in Eq. 3-5.

$$L(\vec{p} | \vec{k}) = P(\vec{k} | \vec{p}) \quad (3)$$

$$P(\vec{p} | \vec{k}) = \frac{L(\vec{p} | \vec{k})}{\int_{\vec{p}'} L(\vec{p}' | \vec{k}) d\vec{p}'} = \frac{\Gamma(\sum(k_i + 1))}{\Gamma((\sum k_i) + 1)} \cdot L(\vec{p} | \vec{k}) \quad (4)$$

$$= \frac{\Gamma(\sum(k_i + 1))}{\prod (\Gamma(k_i + 1))} \cdot \prod (p_i^{k_i}) \quad (5)$$

The density function $P(\vec{p} | \vec{k})$ has its maximum for $\vec{p} = \hat{\vec{p}}$ and the variance decreases by increasing $n = \sum k_i$.

Given a performance measure like accuracy, a Bayesian optimal classifier [16] selects the most probable class \hat{y} (based on its observed frequency $k_{\hat{y}}$) according to Eq. 6. The true posterior $p_{\hat{y}}$ of this selected class corresponds to the resulting accuracy, as expressed by the performance function in Eq. 7.

$$\hat{y} = \arg \max_{y \in \{1, \dots, C\}} (k_y) \quad (6)$$

$$\text{perf}(\vec{k} | \vec{p}) = p_{\hat{y}} \quad (7)$$

$$= \prod p_i^{d_i} \quad d_i = \begin{cases} 1 & \text{if } i = \hat{y} \\ 0 & \text{if } i \neq \hat{y} \end{cases} \quad (8)$$

Given such a performance function, we calculate the expected current performance for the neighborhood around \vec{x} with observed frequencies in \vec{k} :

$$\text{expCurPerf}(\vec{k}) = \mathbb{E}_{\vec{p}} [\text{perf}(\vec{k} | \vec{p})] \quad (9)$$

$$= \int_{\vec{p}} P(\vec{p} | \vec{k}) \cdot \text{perf}(\vec{k} | \vec{p}) d\vec{p} \quad (10)$$

The goal of our approach is (1) to estimate the gain of performance resulting from an upcoming label based on the set of unlabeled data \mathcal{U} and of labeled data \mathcal{L} and (2) to choose the candidate with the maximal gain (see Eq. 11). Having chosen a generative, probabilistic classifier cl like the Parzen window classifier [3] or the probabilistic k-nearest-neighbor [11], we are able to count the number of labeled occurrences per class given a kernel function K (see Eq. 12). The kernel function is a similarity score with $K(\vec{x}, \vec{x}) = 1$. Finally, we define our active learning score as the density weighted performance gain given in Eq. 13.

⁵ All unspecified iterators start at $i = 1$ and end at C .

$$\vec{x}^* = \arg \max_{\vec{x} \in \mathcal{U}} (\text{alScore}(\vec{x} \mid \mathcal{L}, \mathcal{U})) \quad (11)$$

$$\vec{k} = \text{cl}(\vec{x} \mid \mathcal{L}); \quad k_i = \sum_{\{(\vec{x}', y') \in \mathcal{L} : y' = i\}} K(\vec{x}, \vec{x}') \quad (12)$$

$$\text{alScore}(\vec{x} \mid \mathcal{L}, \mathcal{U}) = P(\vec{x} \mid \mathcal{L} \cup \mathcal{U}) \cdot \text{perfGain}(\text{cl}(\vec{x} \mid \mathcal{L})) \quad (13)$$

We determine the performance gain in Eq. 14 by the difference between the expected performance considering m new labels and the expected current performance. The latter is simply calculated as in Eq. 9, the more general expected performance (see Eq. 15) considers multiple possibilities of a labeling. Therefore, we additionally calculate the expectation value over these possible labelings $\vec{l} = (l_1, \dots, l_C) \in \mathbb{N}^C$. Given a number of hypothetical labels that are allowed to be acquired $m \in \mathbb{N}$, $\sum l_i = m$ in one step, the labeling vector represents the change of observations that would be added to the \vec{k} vector if this labeling would be obtained. Hence, after receiving a labeling \vec{l} , the classifier output changes to $\vec{k} + \vec{l}$. Note that this calculation is exact for $m = 1$, but only an approximation for $m > 1$, as it is unlikely to have another instance \vec{x}' at exactly the same location as the current label candidate \vec{x} (similarity of \vec{x} and \vec{x}' should be 1 to be exact). However, as we only select one instance for labeling at each step, this effect is negligible. Finally, we divide the gain by m to have the average gain per label acquisition.

$$\text{perfGain}(\vec{k}) = \max_{m \leq M} \left(\frac{1}{m} (\text{expPerf}(\vec{k}, m) - \text{expCurPerf}(\vec{k})) \right) \quad (14)$$

$$\text{expPerf}(\vec{k}, m) = \mathbb{E}_{\vec{p}} \left[\mathbb{E}_{\vec{l}} [\text{perf}(\vec{k} + \vec{l} \mid \vec{p})] \right] \quad (15)$$

The labeling \vec{l} is multinomial distributed given the true posterior:

$$P(\vec{l} \mid \vec{p}) = \text{Multinomial}_{\vec{p}}(\vec{l}) = \frac{\Gamma((\sum l_i) + 1)}{\prod (\Gamma(l_i + 1))} \cdot \prod (p_i^{l_i}) \quad (16)$$

With help of these equations it is possible to determine the next best instance for labeling as given in Eq. 13 numerically. Achieving a good numerical performance would be computationally expensive and highly dependent on the number of classes C as well as the step width for integrating the true posterior \vec{p} .

Hence, we propose a closed-form solution for this approach in the following section that reduces the computational cost seriously.

3.3 Fast closed-form solution

To get rid of numerical integration, it is sufficient to simplify the expected performance, as the expected current performance is a special case of the former (see Eq. 17ff.).

$$\text{expCurPerf}(\vec{k}) = \text{expPerf}(\vec{k}, 0) \quad (17)$$

$$\text{expPerf}(\vec{k}, m) = \mathbb{E}_{\vec{p}} \left[\mathbb{E}_{\vec{l}} [\text{perf}(\vec{k} + \vec{l} \mid \vec{p})] \right] \quad (18)$$

$$= \int_{\vec{p}} P(\vec{p} \mid \vec{k}) \cdot \sum_{\vec{l}} P(\vec{l} \mid \vec{p}) \cdot \text{perf}(\vec{k} + \vec{l} \mid \vec{p}) \, d\vec{p} \quad (19)$$

$$= \sum_{\vec{l}} \int_{\vec{p}} P(\vec{p} \mid \vec{k}) \cdot P(\vec{l} \mid \vec{p}) \cdot \text{perf}(\vec{k} + \vec{l} \mid \vec{p}) \, d\vec{p} \quad (20)$$

$$= \sum_{\vec{l}} \int_{\vec{p}} \frac{\Gamma(\sum(k_i + 1))}{\prod (\Gamma(k_i + 1))} \cdot \prod (p_i^{k_i}) \cdot \frac{\Gamma((\sum l_i) + 1)}{\prod (\Gamma(l_i + 1))} \cdot \prod (p_i^{l_i}) \cdot \text{perf}(\vec{k} + \vec{l} \mid \vec{p}) \, d\vec{p} \quad (21)$$

$$= \sum_{\vec{l}} \frac{\Gamma(\sum(k_i + 1))}{\prod (\Gamma(k_i + 1))} \cdot \frac{\Gamma((\sum l_i) + 1)}{\prod (\Gamma(l_i + 1))} \cdot \int_{\vec{p}} \prod (p_i^{k_i + l_i}) \cdot \text{perf}(\vec{k} + \vec{l} \mid \vec{p}) \, d\vec{p} \quad (22)$$

After separating the normalization factors from the integral, we simplify the integral by inserting the performance from Eq. 8 and by calculating the definite integral as above in Eq. 4.

$$\int_{\vec{p}} \prod (p_i^{k_i + l_i}) \cdot \text{perf}(\vec{k} + \vec{l} \mid \vec{p}) \, d\vec{p} \quad (23)$$

$$= \int_{\vec{p}} \prod (p_i^{k_i + l_i}) \cdot \prod p_i^{d_i} \, d\vec{p} \quad (24)$$

$$= \int_{\vec{p}} \prod (p_i^{k_i + l_i + d_i}) \, d\vec{p} = \frac{\prod \Gamma(k_i + l_i + d_i + 1)}{\Gamma(\sum(k_i + l_i + d_i + 1))} \quad (25)$$

Reinserting the integral into Eq. 22 and sorting the terms yields the following equations.

$$\text{expPerf}(\vec{k}, m) = \sum_{\vec{l}} \frac{\Gamma(\sum(k_i + 1))}{\prod (\Gamma(k_i + 1))} \cdot \frac{\Gamma((\sum l_i) + 1)}{\prod (\Gamma(l_i + 1))} \cdot \frac{\prod \Gamma(k_i + l_i + d_i + 1)}{\Gamma(\sum(k_i + l_i + d_i + 1))} \quad (26)$$

$$= \sum_{\vec{l}} \frac{\Gamma(\sum(k_i + 1))}{\Gamma(\sum(k_i + l_i + d_i + 1))} \cdot \frac{\prod \Gamma(k_i + l_i + d_i + 1)}{\prod (\Gamma(k_i + 1))} \cdot \frac{\Gamma((\sum l_i) + 1)}{\prod (\Gamma(l_i + 1))} \quad (27)$$

The first and second factors are simplified as follows.

$$\frac{\Gamma(\sum(k_i + 1))}{\Gamma(\sum(k_i + l_i + d_i + 1))} \quad (28)$$

$$= \frac{\Gamma(\sum(k_i + 1))}{\Gamma(\sum(k_i + 1) + (\sum l_i) + (\sum d_i))} \quad (29)$$

$$= \left(\prod_{j=\sum(k_i+1)}^{(\sum(k_i+l_i+d_i+1)-1)} \frac{1}{j} \right) \frac{\Gamma(\sum(k_i + 1))}{\Gamma(\sum(k_i + 1))} \quad (30)$$

$$= \prod_{j=\sum(k_i+1)}^{(\sum(k_i+l_i+d_i+1)-1)} \frac{1}{j} \quad (31)$$

$$\frac{\prod \Gamma(k_i + l_i + d_i + 1)}{\prod (\Gamma(k_i + 1))} = \prod \frac{\Gamma(k_i + l_i + d_i + 1)}{\Gamma(k_i + 1)} \quad (32)$$

$$= \prod \frac{\left(\prod_{j=k_i+1}^{k_i+l_i+d_i} j \right) \Gamma(k_i + 1)}{\Gamma(k_i + 1)} = \prod \left(\prod_{j=k_i+1}^{k_i+l_i+d_i} j \right) \quad (33)$$

Using Eq. 27, 31 and 33, we get the fast version of the expected

performance, a value within $[0, 1]$.

$$\begin{aligned} \text{expPerf}(\vec{k}, m) &= \sum_{\vec{l}} \left(\prod_{j=\sum(k_i+1)}^{(\sum(k_i+l_i+d_i+1))-1} \frac{1}{j} \right) \\ &\cdot \prod_{j=k_i+1}^{k_i+l_i+d_i} j \cdot \frac{\Gamma((\sum l_i) + 1)}{\prod(\Gamma(l_i + 1))} \end{aligned} \quad (34)$$

Now, the final McPAL usefulness score from Eq. 13 is calculated using Eq. 14 and Eq. 34.

As an example, we calculate the expected performance for $m = 0$ which is equivalent to the expected current performance. As mentioned before, $\hat{y} = \arg \max_{y \in \{1, \dots, C\}} (k_y)$.

$$\begin{aligned} \text{expPerf}(\vec{k}, 0) &= \sum_{\vec{l}} \left(\prod_{j=\sum(k_i+1)}^{(\sum(k_i+1)+(\sum l_i)+(\sum d_i))-1} \frac{1}{j} \right) \\ &\cdot \prod_{j=k_i+1}^{k_i+l_i+d_i} j \cdot \frac{\Gamma((\sum l_i) + 1)}{\prod(\Gamma(l_i + 1))} \end{aligned} \quad (35)$$

$$= \left(\prod_{j=\sum(k_i+1)}^{\sum(k_i+1)+0+1-1} \frac{1}{j} \right) \cdot (k_{\hat{y}} + 1) \cdot 1 = \frac{k_{\hat{y}} + 1}{\sum(k_i + 1)} \quad (36)$$

3.4 Characteristics of McPAL

As briefly discussed in Sec. 3.1, there are different ways to combine the posterior estimates \vec{p} from the classifier to determine a usefulness score. The examples in Fig. 2 show different shapes that lead to different behavior, which is evaluated in Sec. 4.

Fig. 3 shows the ternary heatmap plots for the performance gain function of the McPAL algorithm, i.e. the active learning score without the density weight. In contrast to all other multi-class active learning approaches, McPAL does not only consider the observed probability \vec{p} but also includes the reliability $n = \sum k_i$, which is summarized in the frequency vector $\vec{k} = n \cdot \vec{p}$. This extends the ternary plot by an additional degree of freedom. Therefore, we provide two exemplary figures, one showing the behavior for $n = 1$, and one for $n = 2$.

The left plot of Fig. 3 shows a similar but not identical shape as the confidence based (Conf in Fig. 2). While contour lines for confidence-based sampling are linear, these of McPAL are slightly concave. The highest gain is in the center, which represents regions of absolute uncertainty as the posteriors are equal. The lowest gains are in the corners of the triangle. An increase of reliability n decreases the gain (see right plot), as the epistemic uncertainty (caused by lack of information) decreases. This means that there are situations where instances with a non-equal posterior vector are preferred over those with equal posteriors if there is more evidence that the equal posteriors are more likely to be correct.

The number of hypothetical label acquisitions M in the neighborhood of a labeling candidate is bounded by the globally available budget. In the beginning, it is sufficient to have $M = 1$, as one instance has the highest average benefit for the classification task. Over time, we need more hypothetical labels to achieve this benefit. In our experiments, it was sufficient to set $M = 2$. Applications with more labels should adjust the M to greater values accordingly.

From a decision-theoretic view, it is more reasonable to prefer confidence based active learning over entropy or best-vs-second-best, but

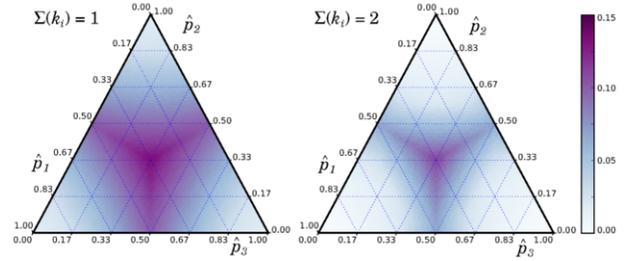


Figure 3. Ternary plot for performance gain for situations with $n = \sum k_i = 1$ (left) and $n = 2$ (right).

the reliability makes a huge difference in the performance as the next section will show.

4 EVALUATION

The goals of our evaluation are twofold: on the one hand, we show the advantage of combining our previously defined impact factors, and on the other hand we compare our multi-class probabilistic active learning approach with state-of-the-art methods. All experiments are conducted based on the setup explained in the following subsection.

4.1 Experimental setup

The proposed method and several other active learning strategies are tested on six datasets, labeling instances successively until the available budget of $b = 60$ label acquisitions has been exhausted. This is done on multiple, seed-based splits of the datasets into independent training and test subsets (training 67%, test 33% of the data) where the number of different training-test-splits for the smaller datasets (ecoli, glass, iris, wine) is 100 and for the large datasets (vehicle, yeast) is set to 50 due to execution time. All experiments are reported by its mean and standard deviation of misclassification cost across all splits. Additionally, we compared each algorithm on all datasets against our method McPAL to determine if our method is significantly better. Therefore, we used a Wilcoxon signed rank test [27] at a p-value of 0.05 and performed the Hommel procedure [10] to prevent the results from errors induced by multiple testing.

The most used visualization of evaluation results are learning curves, which plot the performance in comparison to the number of acquired labels. Our learning curves in Fig. 4 and 5 show the classification error of each active learner on the y-axis, the standard deviation of the error across all splits indicated as an error bar, and the number of instances sampled for the labeled set on the x-axis. In addition to these plots, the results are given in Tab. 4, showing the error and standard deviation of the different active learning methods for all used datasets. The tables show the learner's performance at three different steps, i.e. after 20, 40 and 60 labels have been acquired. Since 60 is the maximum number of sampled instances in the experiments, these steps show the performance in the beginning, intermediate and end phase of the learning process. All results are reported separately for each classifier and dataset. We computed our experiments on a computer cluster running the Neurodebian [8] system.

Besides the proposed method of this paper, six other active learning strategies are used. The McPAL method is executed with $M = 2$, as higher M just increased the execution time but did not change the performance. As a standard baseline, we use a randomly sampling method (Rand). *Confidence-based sampling* (Conf) selects the instance with the lowest maximal posterior ($x^* =$

$\arg \min_{x \in \mathcal{U}} \max_{y \in \mathcal{Y}} \hat{p}_y$) [11]. The next approach uses the shannon entropy to model the uncertainty of an instance (`Entr`) [13]. *Best-vs-Second-Best* (`BvsSB`) samples this instance of the unlabeled set that minimizes the difference of the posterior probabilities of the most probable and the second most probable class [12, 13, 14]. *Maximum-Expected-Cost* (`MaxECost`) determines the value of an instance based on the expected cost associated with the misclassification of that instance. Consequently, the learner samples the instance tied to this score [5]. The last strategy belongs to the expected error reduction based methods. The original *Value of Information* (`VoI`) criterion as suggested by Joshi et al. [13] selects the instance \vec{x} that minimizes a risk measure defined by them. It has to be mentioned that the computational effort of this algorithm forced us to exclude it from the experiments on the vehicle and yeast datasets, since they possess a large number of instances and/or classes, leading to infeasible execution times.

Active learning algorithms require robust classifiers for robust usefulness estimation. Therefore, we choose generative classifiers [19], namely the Parzen window classifier (PWC) [3], and a probabilistic variant of the k-nearest-neighbor classifier (pKNN, with $k = 9$; received good results for our classification tasks (between 3 and 9 classes)) proposed by Jain and Kapoor [11]. These classifiers can be used with any arbitrary similarity function. As the optimization of the overall performance level is not the scope of this paper, we choose to simply standardize each attribute (z-standardization) and use an univariate Gaussian kernel with fixed standard deviation of $\sigma = 0.7$ for all datasets and active learning algorithms. This ensures fair comparability that is independent of a classifier bias.

Table 2. Datasets with the number of instances, the number of attributes and the class frequencies.

Dataset	#Inst.	#Attr.	#Instances per class
Ecoli	336	8	143, 77, 52, 35, 20, 5, 2, 2
Glass	214	10	70, 76, 17, 13, 9, 29
Iris	150	4	50, 50, 50
Vehicle	846	18	212, 217, 218, 199
Wine	178	13	59, 71, 48
Yeast	1484	8	463, 429, 244, 163, 51, 44, 35, 30, 20

We evaluate our algorithm on six multi-class datasets from the UCI repository [2]. The distribution of classes and the number of instances and attributes are summarized in Tab. 2. The *ecoli* dataset was originally used for predicting protein localization sites in eukaryotic cells. The attributes describe properties of proteins. *Glass* was originally generated for classification of types of glass left at a crime scene. The attributes describe chemical ingredients to predict for example whether the glass is from a car window or a window of a building. The *iris* dataset classifies the type of an iris plant, the features describe measures of the plant. *Vehicle* contains features of car models for predicting the manufacturer. The attributes of the *wine* dataset describe the chemical ingredients of a wine instance. The class values are derived from three different cultivars. The *yeast* dataset is also used for predicting the localization site of protein in bacteria. The first column, which held the sequence name, was removed.

The complete results together with an implementation are available at our companion website⁶.

4.2 Impact of influence factors

In Sec. 3.1, we introduced three different influence factors that are considered in McPAL. Fig. 4 shows learning curves on selected datasets and classifiers of McPAL variants with different input parameters using the previously described experimental setup. Thereby, we aim to measure the importance of the different influence factors *posterior*, *reliability*, and *impact*. In addition to the original McPAL algorithm, we show variants that exclude information either (1) about the reliability by normalizing the \vec{k} vector to $\sum \vec{k} = n = 1$ (denoted w/o *reliability*), or (2) about the posterior by replacing the kernel frequency estimate with a uniform one $k_i = n/C, 1 \leq i \leq C$ (denoted w/o *posterior*), or (3) about the density by setting it to a constant (denoted w/o *impact*).

Our selection in Fig. 4 shows that the combination of all influence factors works best. In some cases, the variant without impact is better than the McPAL method. We explain this behavior with the fact that the density, which is used as a proxy for the impact of a label on the complete dataset, gets inaccurate. Especially when there are many labels added to the dataset, this estimate gets worse as the influence also depends on the explicit label situation on the dataset. Nevertheless, the density improved the overall performance although leaving it out is less critical than leaving out one of the other factors.

Especially the results on yeast with the PWC are interesting. Here, leaving out the reliability or the posterior leads to no performance improvement, but unifying these approaches (McPAL) achieves the lowest error.

4.3 Competitiveness of our method

Fig. 5 shows the learning curves of the experiment results with the pKNN classifier, Tab. 4 shows the results using the PWC. As shown in Tab. 4 the McPAL algorithm outperforms its competitors consistently on 4 of the 6 datasets (best performance highlighted in bold text), for the first 20 sampled instances even on 5 out of 6. Using the PWC, our method is only the second best by a close margin after 40 and 60 samples on the vehicle data. After 20 samples random sampling performed best. On the wine dataset, our method scores best at 20 sampled instances but falls behind `Entr` later. As wine data is easy to learn, it is important to mention that the performance almost converged at 30 labels. In general the `BvsSB` and `Entr` algorithms seem to be the most consistent competitors to McPAL in the experiments, the former being the best scoring on the vehicle dataset after 40 samples and the latter outperforming McPAL on the wine dataset after 40 samples.

A good active learning algorithm is characterized by a fast convergence to a good final performance. As can be seen in Fig. 5, our proposed method manages to reduce the classification error quicker than its competitors, in some cases even starting out with a lower error (e.g. *ecoli*, *glass*, *yeast*). Over all datasets, McPAL reduces the error quicker than the other algorithms in the early steps. On top of that, the McPAL algorithm shows a lower standard deviation across all trials compared its competitors (indicated by the error bars in the plots and the brackets in Tab. 4), making it not only the best performing but also the most stable method in the experiments.

For another perspective on the results, the performance of the algorithms in comparison to randomly sampling instances (`Rand`, grey dotted line) should be considered. In case of both the vehicle and yeast dataset McPAL's competitors surpass random instance sampling only late in the learning process in terms of classification error. Even on the iris dataset `Conf`, `BvsSB` and `VoI` struggle to perform better than random selection.

⁶ <http://kmd.cs.ovgu.de/res/mcpal/>

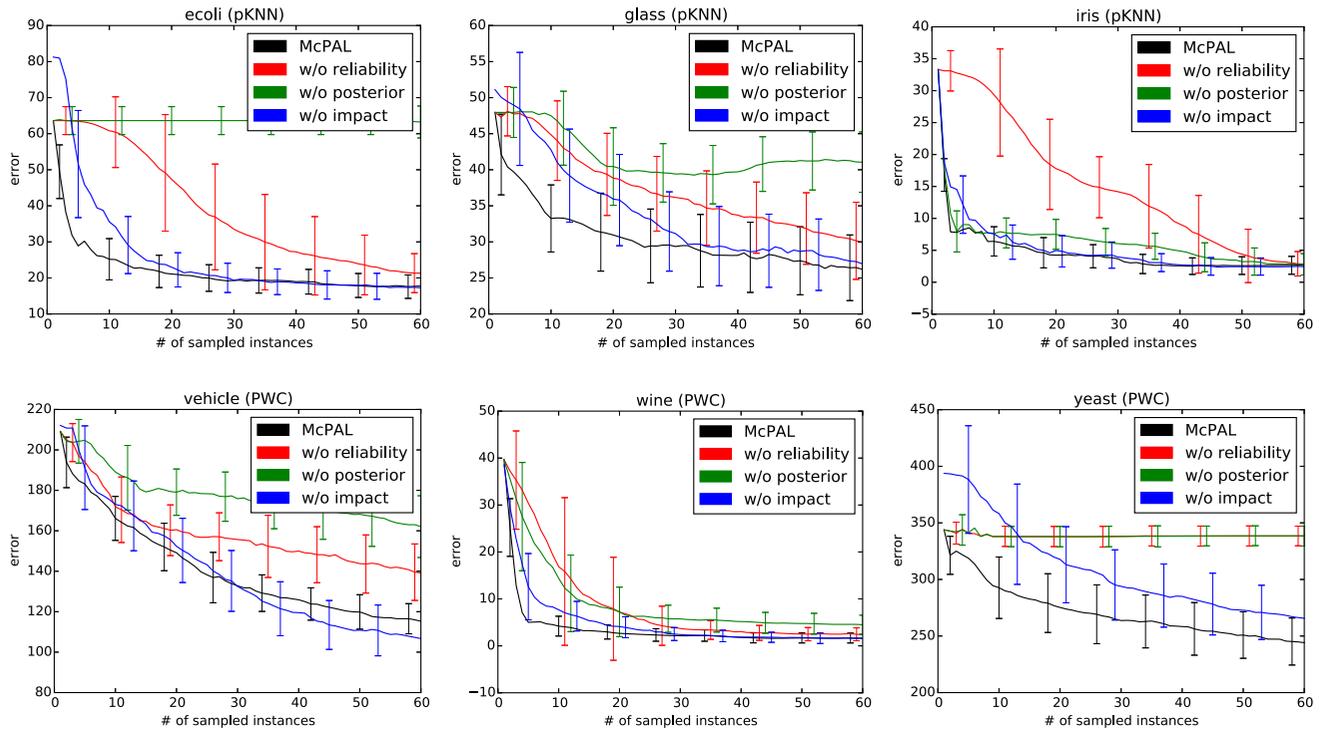


Figure 4. Learning curves of mean misclassification cost (including standard deviation as error bars) different variants of the McPAL algorithm on all six datasets. The upper plots show results from the pKNN classifier, the lower ones with the PWC.

Table 3. Mean execution time for each algorithm for choosing one instance for labeling on the specified dataset in s (sorted by dataset size)

Dataset	McPAL	BvsSB	MaxEC.	Conf	Entr	VoI	Rand
Iris	0.363	0.085	0.083	0.097	0.092	15.94	0.001
Wine	0.584	0.145	0.148	0.153	0.147	36.22	0.001
Glass	1.794	0.200	0.205	0.204	0.204	136.1	0.001
Ecoli	4.590	0.306	0.317	0.313	0.308	518.5	0.001
Vehicle	2.128	0.389	0.394	0.385	0.386	NA	0.001
Yeast	28.06	1.175	1.207	1.171	1.186	NA	0.001

In Tab. 3, we summarized the mean execution time of all algorithms on every dataset. Our proposed method does require more time to sample an instance than its competitors with exception of the VoI algorithm, which takes much longer than any other algorithm used in the experiments. Due to the higher complexity of the McPAL method in comparison to more simple methods like uncertainty-based ones, a longer execution time is to be expected. Considering the performance and stability of McPAL mentioned before, the increased time requirement is still a good trade off. In contrast to the fast methods, McPAL has an additional factor which is the sum over each labeling that is dependent on the m value.

5 CONCLUSION

This paper addresses active learning for multiple classes. This challenging topic opens up different aspects like the combination of

the posterior vector into one comparable score. In this paper, we proposed a new multi-class probabilistic active learning method (McPAL) that addresses this problem in a decision-theoretic way. To this end, we developed a generalized probabilistic model that combines all of our mentioned influence factors impact, posterior, and the reliability of the posterior. Our approach directly optimizes a performance measure like accuracy, is non-myopic and fast. We showed how the influence factors depend on each other in our probabilistic framework and evaluated their behavior in multiple experiments. Especially the combination of the posterior and its reliability makes a huge difference. Our experimental comparison with the most relevant multi-class active learning approaches shows that McPAL is superior in most cases or at least comparable. We suggest that our approach can still be optimized by replacing the proxies of our influence factors by even more appropriate ones, which will be part of our future research. The complete results together with an implementation are available at our companion website⁷.

ACKNOWLEDGEMENTS

We would like to thank the reviewers, Michael Hanke, Alex Waite and our colleague Pawel Matuszyk for all discussions. Ternary plots are generated with python-ternary [9].

⁷ <http://kmd.cs.ovgu.de/res/mcpal/>

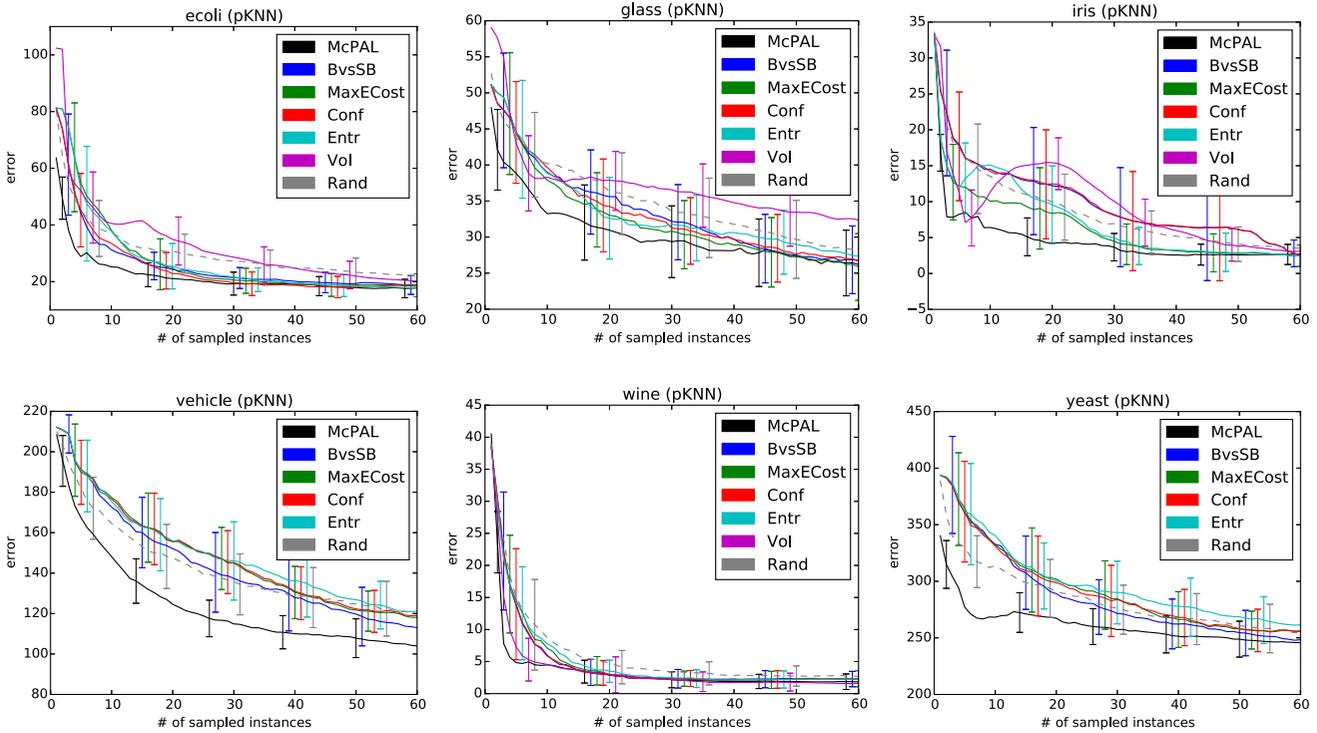


Figure 5. Learning curves of mean misclassification cost (including standard deviation as error bars) of McPAL and its competitors on all six datasets using the pKNN classifier.

Table 4. Mean misclassification cost and its standard deviation of the all algorithms on all six datasets using the Parzen window classifier. We report the results after 20, 40, and 60 acquired labels. The best method is printed in bold numbers. Results showing significant superiority of McPAL against other algorithms are indicated with *.

	ecoli	glass	iris	vehicle	wine	yeast
20 samples						
McPAL	22.70 (± 4.45)	30.17 (± 4.22)	3.94 (± 1.97)	149.14 (± 11.94)	2.66 (± 1.43)	275.24 (± 26.35)
BvsSB	24.75 (± 4.84) *	35.95 (± 5.57) *	12.63 (± 7.06) *	148.68 (± 18.25)	2.80 (± 1.67)	289.90 (± 23.13) *
MaxECost	25.42 (± 6.63) *	33.33 (± 5.09) *	8.23 (± 6.24) *	155.98 (± 17.71)	2.95 (± 1.88)	294.20 (± 32.95) *
Conf	24.64 (± 7.07) *	33.93 (± 5.02) *	12.48 (± 7.32) *	156.52 (± 17.19)	2.90 (± 1.79)	292.92 (± 34.42) *
Entr	26.94 (± 8.01) *	33.04 (± 5.50) *	14.61 (± 3.17) *	153.44 (± 18.82)	3.41 (± 1.76) *	298.60 (± 32.63) *
VoI	40.14 (± 9.59) *	38.20 (± 3.98) *	16.55 (± 2.67) *	NA	2.89 (± 2.68)	NA
Rand	32.52 (± 7.89) *	36.69 (± 5.00) *	9.91 (± 4.47) *	145.38 (± 13.27)	4.35 (± 3.04) *	300.12 (± 23.56) *
40 samples						
McPAL	19.15 (± 4.06)	29.14 (± 4.22)	2.85 (± 1.58)	125.88 (± 8.99)	1.78 (± 1.06)	258.36 (± 24.40)
BvsSB	21.02 (± 4.42) *	32.28 (± 4.36) *	11.78 (± 7.78) *	122.90 (± 14.43)	1.92 (± 1.26)	273.52 (± 22.95) *
MaxECost	20.80 (± 4.10) *	29.70 (± 4.46)	7.70 (± 6.44) *	131.82 (± 14.44)	1.90 (± 1.16)	274.54 (± 30.65) *
Conf	19.60 (± 4.30) *	29.79 (± 4.87)	11.69 (± 7.79) *	133.56 (± 14.90) *	1.94 (± 1.19)	276.36 (± 32.40) *
Entr	23.55 (± 4.80) *	30.64 (± 4.61) *	13.88 (± 3.49) *	139.02 (± 18.57) *	1.77 (± 1.14)	284.38 (± 28.05) *
VoI	41.46 (± 7.22) *	38.06 (± 3.78) *	16.74 (± 2.58) *	NA	1.92 (± 1.89)	NA
Rand	29.80 (± 6.57) *	34.57 (± 5.18) *	8.28 (± 4.03) *	129.88 (± 13.31)	2.65 (± 1.61) *	281.84 (± 25.48) *
60 samples						
McPAL	18.41 (± 3.69)	27.08 (± 3.95)	5.81 (± 2.54)	115.26 (± 7.60)	1.63 (± 1.06)	244.12 (± 20.71)
BvsSB	19.69 (± 4.44) *	29.71 (± 4.22) *	12.71 (± 7.64) *	113.42 (± 9.95)	1.76 (± 1.13)	259.68 (± 22.66) *
MaxECost	20.29 (± 4.55) *	27.99 (± 4.25)	8.12 (± 5.62) *	120.06 (± 12.42) *	1.66 (± 1.03)	257.60 (± 26.75) *
Conf	19.91 (± 4.29) *	28.46 (± 4.59) *	12.40 (± 7.59) *	122.34 (± 13.39) *	1.62 (± 1.12)	259.98 (± 25.76) *
Entr	22.54 (± 4.55) *	31.65 (± 4.91) *	11.94 (± 4.07) *	126.06 (± 14.60) *	1.53 (± 1.00)	272.44 (± 24.93) *
VoI	34.20 (± 5.78) *	37.22 (± 4.72) *	15.06 (± 3.49) *	NA	1.54 (± 1.22)	NA
Rand	28.32 (± 5.65) *	33.55 (± 5.17) *	6.92 (± 2.76) *	123.28 (± 13.26) *	2.30 (± 1.43) *	276.42 (± 26.98) *

REFERENCES

- [1] Alekh Agarwal, ‘Selective sampling algorithms for cost-sensitive multiclass prediction’, in *Proceedings of the 30th International Conference on Machine Learning*, pp. 1220–1228, (2013).
- [2] Arthur Asuncion and David J. Newman. UCI machine learning repository, 2015.
- [3] Olivier Chapelle, ‘Active learning for parzen window classifier’, in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pp. 49–56, (2005).
- [4] Gang Chen, Tian-jiang Wang, Li-yu Gong, and Perfecto Herrera, ‘Multi-class support vector machine active learning for music annotation’, *International Journal of Innovative Computing, Information and Control*, **6**(3), 921–930, (2010).
- [5] Po-Lung Chen and Hsuan-Tien Lin, ‘Active learning for multiclass cost-sensitive classification using probabilistic models’, in *Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 13–18, (2013).
- [6] Husheng Guo and Wenjian Wang, ‘An active learning-based svm multi-class classification model’, *Pattern Recognition*, **48**(5), 1577–1597, (2015).
- [7] Isabelle Guyon, Gavin Cawley, Gideon Dror, Vincent Lemaire, and Alexander Statnikov, ‘Active learning challenge’, *Challenges in machine learning*, **6**, (2012).
- [8] Yaroslav O Halchenko and Michael Hanke, ‘Open is not enough. let’s take the next step: an integrated, community-driven computing platform for neuroscience’, *Frontiers in neuroinformatics*, **6**, (2012).
- [9] Marc Harper, Bryan Weinstein, Cory Simon, chebee7i, Nick Swanson-Hysell, The Gitter Badger, Maximiliano Greco, and Guido Zuidhof. python-ternary: Ternary plots in python, December 2015.
- [10] Gerhard Hommel, ‘A stage-wise rejective multiple test procedure based on a modified bonferroni test’, *Biometrika*, **75**, 383–386, (2010).
- [11] Paril Jain and Ajay Kapoor, ‘Active learning for large multi-class problems’, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 762–769. IEEE, (2009).
- [12] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos, ‘Multi-class active learning for image classification’, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2372–2379, (June 2009).
- [13] Ajay J Joshi, Fatih Porikli, and Nikolaos P Papanikolopoulos, ‘Scalable active learning for multiclass image classification’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(11), 2259–2273, (2012).
- [14] Christine Körner and Stefan Wrobel, ‘Multi-class ensemble-based active learning’, in *European Conference on Machine Learning (ECML)*, 687–694, Springer, (2006).
- [15] Daniel Kottke, Georg Krempel, and Myra Spiliopoulou, ‘Probabilistic active learning in data streams’, in *Advances in Intelligent Data Analysis XIV - 14th Int. Symposium, IDA 2015, St. Etienne, France*, eds., Tijl De Bie and Elisa Fromont, volume 9385 of *Lecture Notes in Computer Science*, pp. 145–157. Springer, (2015).
- [16] Georg Krempel, Daniel Kottke, and Vincent Lemaire, ‘Optimised probabilistic active learning (OPAL) for fast, non-myopic, cost-sensitive active classification’, *Machine Learning (Special Issue of ECML PKDD 2015)*, **100**(2), 449–476, (2015).
- [17] Georg Krempel, Daniel Kottke, and Myra Spiliopoulou, ‘Probabilistic active learning: A short proposition’, in *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI2014), August 18 – 22, 2014, Prague, Czech Republic*, eds., Torsten Schaub, Gerhard Friedrich, and Barry O’Sullivan, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 1049–1050. IOS Press, (2014).
- [18] David D. Lewis and William A. Gale, ‘A sequential algorithm for training text classifiers’, in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’94*, pp. 3–12, New York, NY, USA, (1994). Springer-Verlag New York, Inc.
- [19] Andrew Y. Ng and Michael I. Jordan, ‘On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes’, in *Advances in Neural Information Processing Systems 14 (NIPS)*, pp. 841–848, (2001).
- [20] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, Cambridge University Press, 2 edn., 1992.
- [21] Nicholas Roy and Andrew McCallum, ‘Toward optimal active learning through sampling estimation of error reduction’, in *Proc. of the 18th Int. Conf. on Machine Learning, ICML 2001, Williamstown, MA, USA*, pp. 441–448, San Francisco, CA, USA, (2001). Morgan Kaufmann.
- [22] Robin Senge, Stefan Bösner, Krzysztof Dembczyński, Jörg Haasenritter, Oliver Hirsch, Norbert Donner-Banzhoff, and Eyke Hüllermeier, ‘Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty’, *Information Sciences*, **255**, 16–29, (January 2014).
- [23] Burr Settles, *Active Learning*, number 18 in Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool Publishers, 2012.
- [24] Katrin Tomanek and Udo Hahn, ‘Reducing class imbalance during active learning for named entity annotation’, in *Proceedings of the 5th International Conference on Knowledge Capture (K-CAP), September 1–4, 2009, Redondo Beach, California, USA*, eds., Yolanda Gil and Natasha Fridman Noy, pp. 105–112. ACM, (2009).
- [25] Simon Tong and Edward Chang, ‘Support vector machine active learning for image retrieval’, in *Proceedings of the Ninth ACM International Conference on Multimedia*, pp. 107–118. ACM, (2001).
- [26] R. Wang, C. Y. Chow, and S. Kwong, ‘Ambiguity-based multiclass active learning’, *IEEE Transactions on Fuzzy Systems*, **24**(1), 242–248, (Feb 2016).
- [27] Frank Wilcoxon, ‘Individual comparisons by ranking methods’, *Biometrics bulletin*, **1**(6), 80–83, (1945).
- [28] Rong Yan and Alexander Hauptmann, ‘Multi-class active learning for video semantic feature extraction’, in *IEEE International Conference on Multimedia and Expo (ICME)*, volume 1, pp. 69–72. IEEE, (2004).
- [29] Rong Yan, Jie Yang, and Alexander Hauptmann, ‘Automatically labeling video data using multi-class active learning’, in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pp. 516–523. IEEE, (2003).
- [30] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann, ‘Multi-class active learning by uncertainty sampling with diversity maximization’, *International Journal of Computer Vision*, **113**(2), 113–127, (2014).

Online Prediction of Exponential Decay Time Series with Human-Agent Application

Ariel Rosenfeld,¹ Joseph Keshet,² Claudia V. Goldman,³ Sarit Kraus⁴

Abstract. Exponential decay time series are prominent in many fields. In some applications, the time series behavior can change over time due to a change in the user's preferences or a change of environment. In this paper we present an innovative online learning algorithm, which we name *Exponentron*, for the prediction of exponential decay time series. We state a regret bound for our setting, which theoretically compares the performance of our online algorithm relative to the performance of the best batch prediction mechanism, which can be chosen in hindsight from a class of hypotheses after observing the entire time series. In experiments with synthetic and real-world data sets, we found that the proposed algorithm compares favorably with the classic time series prediction methods by providing up to 41% improvement in prediction accuracy. Furthermore, we used the proposed algorithm for the design of a novel automated agent for the improvement of the communication process between a driver and its automotive climate control system. Throughout extensive human study with 24 drivers we show that our agent improves the communication process and increases drivers' satisfaction, exemplifying the *Exponentron*'s applicative benefit.

1 Introduction

Exponential decay functions are popular in modeling real-world phenomena. For example, the decrease in radioactivity levels of radioactive substances, the cooling of an object in a cold environment [24] and human decline of memory retention over time [13] are all assumed to decay exponentially over time, and are usually modeled using exponential decay functions.

There are several learning algorithms for the prediction of exponential decay time series such as the Bayesian [6], regressive [12] or autoregressive [9] methods based on labeled training data (i.e. set of observed time series). However, most of these algorithms introduce two limitations: First, the algorithms are batch in nature and need to get the whole training set in advance. As such, *batch* algorithms cannot adapt to changes over time in the modeled phenomenon which may occur after the initial training phase, for example when modeling human preferences over time. Second, the algorithms are *general purpose*, and do not exploit the exponential decay nature of the time series. While a few online learning versions of the above models have been investigated recently [20], to the best of our knowledge none have specifically addressed exponential decay time series.

In this paper we present an online learning algorithm, which we call *Exponentron*, for the prediction of exponential decay time series.

The online learning algorithm takes place in a sequence of consecutive rounds. In each round, the learner first receives a time series instance. Then, the learner is required to predict its parameters. At the end of the round, the learner obtains the correct parameters, and uses this information to improve its future predictions.

The *Exponentron* algorithm focuses on a special hypothesis family capturing the assumed exponential decay behavior of time series. It is aimed at optimizing the square loss function, and like other online learning algorithms it does not require any training data before deployment.

We state a regret bound for the *Exponentron* algorithm. Regret bounds are common in the analysis of online learning algorithms. A regret bound measures the performance of an online algorithm relative to the performance of the best competing hypothesis, which can be chosen in hindsight from a class of hypotheses, after observing the entire time series.

Empirically, we show that the algorithm significantly outperforms classic time series prediction methods' accuracy in predicting assumed exponential decay time series in two repeated real-world settings by up to 41%. *Exponentron* is then used for the enhancement of driver-automotive climate control system (CCS) interaction. A novel intelligent agent for Natural Interaction with humans in CCS using the *Exponentron* algorithm, which we named the NICE agent, is presented. The NICE agent was extensively evaluated with 24 human drivers in hot summer conditions. The agent successfully reduced the number of interactions needed by the driver to achieve her desired comfort state by 19% compared to state-of-the-art CCSs. The agent is also shown to achieve high driver satisfaction.

This paper makes the following contributions; (1) We propose a novel online learning algorithm, named *Exponentron*, for the prediction of exponential decay time series. The algorithm is the first of its kind as it is tailored to a unique class of time series. We provide both theoretical analysis and empirical evaluation of the algorithm. (2) We present a novel intelligent agent, named the *NICE* agent, that provides the driver with intelligent, natural interface with which she can change the parameters of her automotive climate control system (CCS). The agent reduces the driver's need for interaction with the CCS, and increases the driver's subjective satisfaction.

2 Time Series Preliminaries

A time series $s = (s_0, s_1, \dots, s_T)$ is an ordered sequence of values measured in equally spaced time intervals, where $s_t \in \mathbb{R}$ denotes an element at time t , $0 \leq t \leq T$. In this work, we assume that the time series was created by an exponential decay process. Assuming $(s_0, s_1, \dots, s_{t-1})$ is the beginning of a series, the prediction of the next element is denoted \hat{s}_t .

¹ Bar Ilan University, Israel, email: arielros1@gmail.com

² Bar Ilan University, Israel.

³ General Motors Advanced Technical Center, Herzliya, Israel.

⁴ Bar Ilan University, Israel.

Intelligent agents often use time series prediction to improve their decision-making. A few recent examples include the repositioning of bikes in bike-sharing systems based on the prediction of bike usage [25], maintenance scheduling based on the prediction of ongoing game scores [31] and the prediction of passenger demand for better taxi routing [22]. A common theme among these systems is the use of classical, *general purpose* time series prediction methods as the basis for their *domain-specific* proposed approach and design.

Among the most commonly applied techniques for forecasting the continuation of a time series given its beginning are the autoregressive (AR) model, the autoregressive-moving average (ARMA) model (see [9] for a review) and the exponential smoothing (ES) forecasting method (see [15] for a review).

The autoregressive (AR) model generates its prediction using the following equation:

$$\hat{s}_t = c + \sum_{i=1}^p \varphi_i s_{t-i} \quad (1)$$

where c , p and φ_i are parameters of the model.

An AR model is in fact a linear regression of the current value of the time series against one or more prior values of the time series. The value of p is called the order of the AR model. The most commonly applied AR method, which is also used in this study, uses $p = 1$. This model is sometimes denoted $AR(1)$.

A popular extension of the AR model uses a moving average (MA), resulting in the autoregressive-moving average (ARMA) model [9]. ARMA is also known as the Box-Jenkins Approach. The ARMA model generates its prediction using the following equation:

$$\hat{s}_t = c + \sum_{i=1}^p \varphi_i s_{t-i} + \sum_{i=1}^q \theta_i e_{t-i}. \quad (2)$$

where $e_j = s_j - \hat{s}_j$ and $c, p, q, \varphi_i, \theta_i$ are parameters of the model.

An ARMA model is in fact a linear regression of the current value of the time series against one or more prior values of the time series and one or more prior noise terms. The value of p is called the order of the AR part of the model and q is the order of the MA part of the model. The most commonly applied ARMA method, which is also used in this study, uses $p = q = 1$. This model is sometimes denoted $ARMA(1, 1)$.

Another prediction method is the exponential smoothing (ES) scheme (also known as the exponentially weighted moving average), which weighs past elements of the time series using exponentially decreasing weights. The most suitable exponential smoothing method, which is used in this study, is the *double* exponential smoothing method, denoted ES . ES forecasts the continuation of a time series using the following equations:

$$\begin{aligned} \hat{s}_t &= \alpha s_{t-1} + (1 - \alpha)(\hat{s}_{t-1} + b_t) \\ b_t &= \gamma(\hat{s}_t - \hat{s}_{t-1}) + (1 - \gamma)b_{t-1} \end{aligned} \quad (3)$$

where $0 < \alpha \leq 1$ and $0 < \gamma \leq 1$.

There are several methods for choosing \hat{s}_0 and b_0 . The most common one, which is also used in this study, is $\hat{s}_0 = s_0$ and $b_0 = 0$. Note that *single* exponential smoothing does not fit time series which present trends, and therefore is unsuitable for this study. *Triple* exponential smoothing, also known as the Holt-Winters exponential smoothing technique [16], is popular in forecasting *seasonal* time series. In our settings we assume no seasonality. The smoothing parameters, α and γ , used by the ES method are usually found using grid search.

Note that the above three methods are both *general purpose* (that is, they can fit a large variety of time series) and work *batch* (the models' parameters do not change during the prediction).

An online version of the ARMA model, denoted O -ARMA was recently analyzed in [1]. The proposed version uses the ARMA model specified in Equation 2, yet the model's parameters may change over time. Nevertheless, note that this method is still general purpose, as is the basic ARMA model.

In this work we provide a solution to the task of *online* predicting the continuation of an assumed *exponential decay* time series. To the best of our knowledge, no intelligent system or machine learning algorithm has addressed this challenge to date.

The above four models (AR , $ARMA$, ES and O -ARMA) are evaluated as baseline models in Section 4 of this study, showing the Exponentron algorithm's superiority.

3 The Exponentron Algorithm

We assume the following set of hypotheses:

$$\hat{s}_t(\boldsymbol{\theta}) = a + b e^{-c(t-t_0)}, \quad (4)$$

where $\boldsymbol{\theta} = (a, b, c)$ is the set of 3 parameters that should be estimated, $\boldsymbol{\theta} \in \mathbb{R}_+^3$, and $t_0 \in \mathbb{R}_+$ is a time offset parameter.

In this work we focus on the online settings, where learning takes place in rounds. In round t , the algorithm observes the series $(s_0, s_1, \dots, s_{t-1})$ and is required to make a prediction for the next element in the series, \hat{s}_t . The algorithm maintains a set of parameters that are updated every round. After making its prediction, \hat{s}_t , the correct value, s_t , is revealed and an instantaneous loss $\ell(\hat{s}_t, s_t)$ is encountered. The round ends with an update of the parameters $\boldsymbol{\theta}$ according to the encountered loss. In this work we use the squared loss function, namely

$$\ell(\hat{s}_t(\boldsymbol{\theta}), s_t) = (\hat{s}_t(\boldsymbol{\theta}) - s_t)^2 = (a + b e^{-c(t-t_0)} - s_t)^2. \quad (5)$$

Our algorithm, which is called Exponentron, is given in Algorithm 1. The algorithm is aimed at minimizing the cumulative loss.

The algorithm starts with a set of feasible parameters $\boldsymbol{\theta}_0 \in \mathbb{R}_+^3$, that is, $\boldsymbol{\theta}_0 = (a_0, b_0, c_0)$ satisfies the constraints on the parameters. We initialize t_0 by setting the first prediction to be correct, namely, $\hat{s}_t = s_t$ for $t = 0$, and get

$$t_0 = \log((s_0 - a)/b)/c. \quad (6)$$

Now the set of parameters needs to satisfy the constraints $a \leq s_0$, $b \geq 0$ and $c \geq 0$.

The following proposition states that the hypothesis function is a convex function.

Proposition 1. The hypothesis function

$$\hat{s}_t(\boldsymbol{\theta}) = a + b e^{-c(t-t_0)} \quad (7)$$

is a convex function in the set of parameters $\boldsymbol{\theta} = (a, b, c)$.

Proof. Since $b > 0$ we can rewrite it as $b = e^{\tilde{b}}$, and the hypothesis becomes $\hat{s}_t(\boldsymbol{\theta}) = a + e^{\tilde{b}-c(t-t_0)}$. Now it is easy to verify that

$$\hat{s}_t(\alpha \boldsymbol{\theta}_1 + (1 - \alpha) \boldsymbol{\theta}_2) \leq \alpha \hat{s}_t(\boldsymbol{\theta}_1) + (1 - \alpha) \hat{s}_t(\boldsymbol{\theta}_2),$$

for the sets $\boldsymbol{\theta}_1 = (a_1, \tilde{b}_1, c_1)$, and $\boldsymbol{\theta}_2 = (a_2, \tilde{b}_2, c_2)$ which satisfy the constraints, and $\alpha \in (0, 1)$. \square

Since our loss function in Eq. (5) is also a convex function, it turns out that the loss is convex.

Our algorithm is based on *gradient projected methods* [7, pp. 228]. The algorithm starts with a set of feasible parameters $\theta_0 \in \mathbb{R}_+^3$, that is, θ_0 satisfies the constraints. At the t -th round the algorithm predicts the next element in the time series based on the parameters θ_{t-1} . Then, if the encountered loss, $\ell(\hat{s}_t(\theta_{t-1}), s_t)$, is greater than zero, the parameters are updated by a gradient step: $\theta' = \theta_{t-1} + \eta_t \nabla_{\theta} \ell$, where the gradient of the loss is the following vector:

$$\nabla_t = 2(\hat{s}_t(\theta) - s_t)[1, e^{-c(t-t_0)}, -b(t-t_0)e^{-c(t-t_0)}]. \quad (8)$$

The parameter η_t is the learning rate. Specifically in our case we set $\eta_t = \eta_0/\sqrt{t}$, where η_0 is chosen when the algorithm starts (as in [33, 8]).

At the end of each round the algorithm projects θ' on a set of constraints in order to get θ_t , a feasible vector.

Algorithm 1 The Exponentron Algorithm

Require: initialize θ_0 , learning parameter η .

- 1: observe s_0 and set t_0 according to Eq. (6)
 - 2: **for** $t = 1, 2, \dots, T$ **do**
 - 3: predict $\hat{s}_t = a_{t-1} + b_{t-1} e^{-c_{t-1}(t-t_0)}$
 - 4: observe true value s_t
 - 5: encounter loss $\ell(\hat{s}_t, s_t)$
 - 6: update parameters and project
 - 7: $a_t = \min\{s_0, a_{t-1} - 2\eta_t(\hat{s}_t - s_t)\}$
 - 8: $b_t = \max\{0, b_{t-1} - 2\eta_t(\hat{s}_t - s_t)e^{-c_{t-1}(t-t_0)}\}$
 - 9: $c_t = \max\{0, c_{t-1} + 2\eta_t(\hat{s}_t - s_t)b_{t-1} / (t-t_0)e^{-c_{t-1}(t-t_0)}\}$
-

Note that the Exponentron algorithm does not require any training before deployment.

Often the performance of an online algorithm is measured by how *competitive* it is with the hypothesis of the best *fixed* parameters θ^* . This is captured by the notion of the algorithm's *regret*, which is defined as the excess loss for not consistently predicting with the parameters θ^* ,

$$\text{regret}(\theta^*, T) \triangleq \sum_{t=1}^T \ell(\hat{s}_{t-1}(\theta), s_t) - \sum_{t=1}^T \ell(\hat{s}_t(\theta^*), s_t). \quad (9)$$

The following theorem states that the regret of the Exponentron algorithm is bounded.

Theorem 2. *The Exponentron algorithm has the following regret bound for every θ^* in \mathbb{R}_+^3 ,*

$$\text{regret}(\theta^*, T) \leq \frac{\sqrt{T}}{2} \|\theta\|^2 + \frac{1}{2\sqrt{T}} \sum_{t=1}^T \|\nabla_t\|^2. \quad (10)$$

Proof. The analysis is based on the stochastic gradient descent with projection analysis [7]. Denote by θ_{t-1} the set of parameters before the update, by $\theta_{t-1/2}$ the set of parameters after the gradient step,

and by θ_t the set of parameters after the projection step. We have

$$\begin{aligned} & \|\theta_t - \theta^*\|^2 - \|\theta_{t-1} - \theta^*\|^2 \\ &= \|\theta_t - \theta^*\|^2 - \|\theta_{t-1/2} - \theta^*\|^2 \\ & \quad + \|\theta_{t-1/2} - \theta^*\|^2 - \|\theta_{t-1} - \theta^*\|^2 \\ & \leq \|\theta_{t-1/2} - \theta^*\|^2 - \|\theta_{t-1} - \theta^*\|^2 \\ &= \|\theta_{t-1} - \eta \nabla_t - \theta^*\|^2 - \|\theta_{t-1} - \theta^*\|^2 \\ &= -2\eta(\theta_{t-1} - \theta^*) \cdot \nabla_t + \eta^2 \|\nabla_t\|^2 \\ & \leq -2\eta \left(\ell(\hat{s}_t(\theta_{t-1}), s_t) - \ell(\hat{s}_t(\theta^*), s_t) \right) + \eta^2 \|\nabla_t\|^2 \end{aligned}$$

From the second line to the third line we used the property of projections, $\|\theta_t - \theta^*\|^2 \leq \|\theta_{t-1/2} - \theta^*\|^2$. We now sum over all t , and get

$$\begin{aligned} & \|\theta_T - \theta^*\|^2 - \|\theta_0 - \theta^*\|^2 \\ & \leq -2\eta \sum_{t=1}^T \left(\ell(\hat{s}_t(\theta_{t-1}), s_t) - \ell(\hat{s}_t(\theta^*), s_t) \right) + \frac{1}{2}\eta \sum_{t=1}^T \|\nabla_t\|^2 \end{aligned}$$

By rearranging terms we get the desired result. \square

4 Exponentron Evaluation

We first evaluate the Exponentron algorithm using synthetic exponential decay time series. Then, we evaluate the Exponentron algorithm in two real-world prediction tasks of significant importance: First, we evaluate the Exponentron in predicting drivers' desired interior cabin temperature during a drive. Specifically, we focus on the cooling condition, where a driver wishes to cool the interior cabin temperature of a car in order to achieve a comfortable state. Second, we wish to predict the number of arriving calls at a call center. Specifically, we consider a call center in which human service agents can handle both inbound calls and other back-office tasks, making the prediction of arriving calls an important factor in real-time work schedule adjustment and managerial decision-making [14].

We compare Exponentron against 4 time series prediction methods, *AR*, *ARMA*, *ES* and *O-ARMA* which are described in Section 2.

4.1 Synthetic Data Prediction

To gain intuition about the relative strengths of the Exponentron algorithm, we evaluate the Exponentron in a synthetic exponential decay time series prediction task.

To this end, we synthetically generated 1,000 exponential decay time series, which all start at the value 100 at $t = 0$, denoted $s_0 = 100$. Each time series is represented as a tuple $\theta = (a, b, c)$ and is generated according to Equations 4 and 6; $s_t(\theta) = a + b e^{-c(t-t_0)}$ where $t_0 = \log((s_0 - a)/b)/c$. We synthetically generated the time series by sampling $a \in U[10, 80]$, $b \in U[20, 90]$ and $c \in U[0.1, 0.5]$.⁵ We then randomly assigned 900 time series to a training set and the remaining 100 time series were assigned to the test set. The training set time series are used to train the Exponentron algorithm, alongside the four baseline models described in Section 2. The coefficients used by the *AR* and *ARMA* methods were learned using a simple linear regression over the training set time series.

⁵ a , b and c were chosen as such to allow a significant range of possible hypotheses and yet restrict the range to allow a reasonable first estimation of a , b and c by each of the tested algorithms.

Similarly, the smoothing parameters used by the *ES* method were found using a grid search. *O-ARMA* was implemented according to [1]. The Exponentron’s initial parameters, θ_0 , were set to the least squares regression parameters calculated based on the training data as described in [32].

The prediction models were tested over the test set time series. Overall, Exponentron significantly outperforms each of the tested baseline models using univariate ANOVA with the prediction method as the independent factor and the prediction mean absolute error as the dependent variable, $p < 0.05$. An illustration of the predicted values made by the Exponentron and the *ARMA* model is presented in Figure 1.

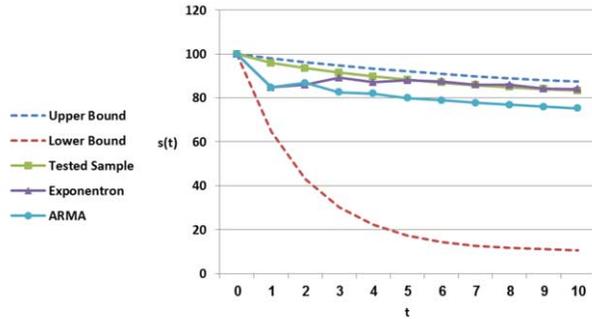


Figure 1: All time series were sampled from the confined space between the upper and lower bounds. The green line is one of the tested time series (represented by $a = 79, b = 20, c = 0.15$) and the purple and light blue lines are the predictions made by the Exponentron and ARMA, respectively.

This synthetic experiment demonstrates the advantage of the Exponentron in exponential decay time series prediction. We now turn to investigate the Exponentron’s advantages using two real-world data sets.

4.2 Predicting Desired Climate Changes in a Car’s Interior

4.2.1 Data Collection

The cabin temperature time series is (s_0, s_1, \dots, s_T) where s_0 is the initial cabin temperature when a driver turns the CCS on, and s_j is the cabin temperature k seconds after s_{j-1} . In our setting, k was set to 15 seconds.⁶ A driver’s preferred cabin temperature time series is (s_0^*, s_1^*, \dots) where $s_0^* = s_0$ and s_i^* is the desired cabin temperature at time frame i . A cabin temperature is said to be **steady** if in a period of 1 minute the driver does not change the CCS features and the cabin temperature does not change more than ϵ . In our setting, ϵ was set to 0.1°C .⁷ We focus on the task of predicting s_i^* given $(s_0^*, \dots, s_{i-1}^*)$ until a steady cabin temperature has been reached.

We recruited 28 drivers, ranging in age from 25 to 57 (average of 35), 22 males and 6 females. Each subject was asked to enter a car, which was parked in a garage, in order to experience the environmental conditions – temperatures ranging from 21°C to 31°C , averaging 27°C . Each subject was presented with a newly designed graphical

interface presented on a tablet which we call the *natural interface*. The natural interface presents natural terms such as “Too cold”, “Too hot” and “Noisy”, which are unavailable in most CCSs. Each subject was instructed to interact with the system, such that after any button was pressed the subject had to *manually* change the features of the CCS using the car’s standard interface with the help of our research assistant. Namely, the natural interface did not have any functionality behind it at this point. The session stopped once the cabin temperature of the car was *steady*. While in the car the subject was given a cell phone with a driving simulator “Bus Simulator 3D”⁸ to be played while the experiment was conducted. The motivation was to set the conditions similar to regular driving conditions and give the subjects something to do. Unfortunately, due to insurance reasons, we could not conduct the study while subjects were actually driving.⁹ The subject was then asked to exit the car for a period of 10 minutes while the car’s doors were left open in order to simulate the initial conditions. Note that the subject could choose to click on any button at any given moment, thereby changing the CCS (manually).

During the session, the internal cabin temperature of the car was recorded using a state-of-the-art thermometer that we placed between the driver’s and the front passenger’s seats. The temperature was measured once every 15 seconds (again, to allow sufficient time for our thermometer to adapt).

Overall, 56 time series were collected. The shortest time series consisted of 6 data points whereas the longest consisted of 26 data points (mean of 13).

4.2.2 Analysis

The Exponentron algorithm, alongside the four baseline models described in Section 2, was evaluated based on the collected data.

The baseline models were trained and evaluated using a one-left-out methodology. Namely, we took out one series at a time from the data set and used the remaining series as training data. All four models were trained as described in Section 4.1.

Note that the Exponentron algorithm and *O-ARMA* do not necessitate any training prior to deployment, but instead require an initialization of the parameters (θ_0 in Algorithm 1). Nevertheless, we chose to set the initial parameters to the least squares regression parameters calculated based on the training data using a one-left-out methodology as described in [32]. We also examined the initialization of the Exponentron’s parameters using only a subset of the training data. Surprisingly, using *any* single time series from the training data to determine the Exponentron’s initial parameters resulted in a less than 10% decrease in the Exponentron’s accuracy compared to using all of the training data.

The Exponentron’s mean absolute error was found to be 0.13°C per value in the time series. The Exponentron’s predictions are 28% more accurate as compared to the best tested baseline model, *O-ARMA*, which yields a 0.18°C mean absolute error. Table 1 provides a summary of the tested models’ prediction errors.

Overall, Exponentron significantly outperforms each of the tested baseline models using univariate ANOVA with the prediction method as the independent factor and the prediction mean absolute error as the dependent variable, $p < 0.05$.

⁶ A time interval of 15 seconds was chosen to allow sufficient time for our thermometer to adapt to the changing temperature inside the car. The thermometer specification states that it takes up to 15 seconds for the thermometer to adapt to its environment.

⁷ The inaccuracy interval of our thermometer.

⁸ Available free at Google Play store.

⁹ “Bus Simulator 3D” was also used in previous human-CCS interaction studies in order to simulate driving conditions [5].

Method	Mean Absolute Error (per 15 seconds)
$AR(1)$	0.21
$ARMA(1, 1)$	0.2
$ES(1, 0)$	0.24
$O-ARMA(1, 1)$	0.18
Exponentron	0.13

Table 1: Prediction of the desired interior temperature of the car. Numbers indicate the mean absolute error made by each prediction method per 15 second frame.

4.3 Inbound Calls in a Real-World Call Center

4.3.1 Real-World Call Center – Secondary Data

We use data that was collected and analyzed in [21]. The data accounts for all inbound calls arriving at the small call center of a bank in 1999 [17]. On weekdays the center is open from 7am to midnight (17 hours) and provides service to over 2,000 callers (on average). We focus on the 16:00-24:00 (8 hours) time frame, in which an average of approximately 750 calls arrive at the call center in an assumed exponential decay manner.

Following the original analysis procedure, we processed the data such that all national holidays were removed and each of the daily recordings was translated into a time series. For this evaluation we used a time series of the form $(c_{17}, c_{18}, \dots, c_{24})$ where c_i is the number of arriving calls during the $(i - 1)^{\text{th}}$ hour of the day. For example, all calls arriving between 17:00 and 18:00 will count as c_{18} .

Overall, 222 time series were constructed. Each time series consists of 8 data points.

4.3.2 Analysis

The Exponentron algorithm, alongside the baseline models described in Section 2, was evaluated using the same procedure as described in Section 4.2.2. Again, we noticed that using *any* single time series from the training data to determine the Exponentron’s initial parameters resulted in a less than 15% decrease in the Exponentron’s accuracy compared to using the entire training data.

At each time frame of one hour the Exponentron’s mean absolute error was found to be 5.8 calls. The Exponentron’s predictions are 41% more accurate than the best tested baseline model, $ARMA$, which yields a mean absolute error of 9.8 calls. Table 2 provides a summary of the tested models’ prediction errors.

Method	Mean Absolute Error (hourly)
$AR(1)$	10.2
$ARMA(1, 1)$	9.8
$ES(0.9, 1)$	13.7
$O-ARMA(1, 1)$	9.9
Exponentron	5.8

Table 2: Prediction of call arrivals in a real-world call center. Numbers indicate the mean absolute error made by each prediction method.

Figure 2 demonstrates the predictions provided by the Exponentron and $ARMA(1, 1)$ for the number of arriving calls per hour during the evening of February 8th, 1999.

Overall, Exponentron significantly outperforms each of the tested baseline models using pairwise t-tests ($p < 0.05$).

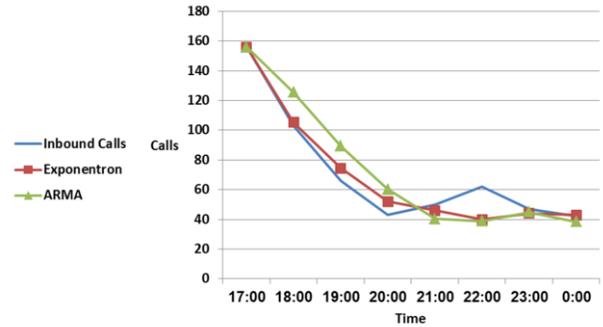


Figure 2: The prediction of inbound calls made by the $ARMA$ and Exponentron models for the evening of 8/2/1999.

5 The Exponentron-Based NICE Agent

5.1 The Agent-Human Interaction Challenge

In this section we focus on an agent-human interaction challenge in Climate Control Systems (CCSs). We aim to automatically adjust the CCS features (e.g. fan speed) in order to provide comfortable settings for the user. Specifically, we address a situation in which a driver enters a hot vehicle and the agent’s goal is to automatically set and adjust the car’s CCS features throughout the ride to the driver’s satisfaction.

The agent’s main goal is to bring about the driver’s desired cabin setting in the car, namely, the appropriate interior cabin temperature and other CCS features. Reaching the target cabin settings is not instantaneous, and may take time and adjustment of the CCS features.

In an interview-based preliminary experiment we conducted with 18 drivers, we noticed that the drivers’ satisfaction from their CCS is affected not only by the target cabin setting of the car but, and even more importantly, by the cabin setting endured during the adjustment process. Furthermore, we observed that people have different preferences for both of the above. However, their preferences during the adjustment process exhibit similar *exponential decay tendencies*. For example, Alice’s target interior cabin temperature is 21°C and she wishes to reach it as fast as possible (she does not mind enduring extreme CCS settings in the process). Bob, on the other hand, wants to reach 19°C but refrains from settings in which the fan speed is higher than 3 and thus prefers milder adjustments. However, both prefer that the interior temperature decrease exponentially.

5.2 Background

Recent evidence suggests that drivers’ current user experience often does not meet drivers’ wishes, making many drivers desire more natural car interfaces [19, 27]. For that purpose, some intelligent systems use drivers’ observed behavior to automatically elicit the drivers’ state or goals [11]. For example, in [29, 30], the authors have shown that learning drivers’ behavior can improve the performance of the adaptive cruise control system to drivers’ satisfaction. Others offer more expressive interfaces that are more natural for the driver to use and understand [18, 26]. The most relevant works within the context of CCS are [5, 28, 4], in which the authors try to elicit drivers’ climate control preferences in order to provide advice to the driver that will help him reduce the climate system’s energy consumption. The authors did not account for the possibility of the agent automatically changing the CCS settings nor did they allow for natural input from the driver.

The thermal comfort of human subjects has been exhaustively investigated over the last four decades, resulting in the ISO 7730 standard¹⁰ [2]. The standard, which was also found to be applicable in car cabins, is aimed at predicting the degree of thermal comfort of an *average person* exposed to a certain *steady* environment (see [10] for a recent survey). Unfortunately, the standard does not provide answers on how a system should bring about a comfortable state.

Furthermore, the standard relies on the assumption that user-specific parameters are available such as thermal sensitivity, clothing and activity level. Despite recent attempts to personalize thermal comfort models [3], state-of-the-art thermal comfort models do not provide personalized or adaptive thermal comfort predictions.

Using the Exponentron algorithm, we will next describe a competing approach which does not necessitate the identification of user-specific characteristics prior to its deployment.

5.3 The NICE Agent Design

The NICE agent's goal is to minimize the number of interactions needed by a driver to reach her desired comfort state and maximize the driver's satisfaction from the interaction process.

The agent implements the Exponentron algorithm (Algorithm 1) in order to predict the driver's desired climate changes during the ride and thereby change the CCS setting.

During the process, the driver may provide feedback to the agent using natural comments, such as "Too cold" or "Too hot", using the natural interface described in Section 4.2. These comments, in turn, are used to adapt the Exponentron's predictions, as we will soon describe.

A CCS setting is a tuple $\omega = \langle temp, f, d \rangle$, where *temp* is the set temperature (an integer between 16 and 35 degrees C), *f* is the fan strength (an integer between 1 and 8) and *d* is the air delivery (1=face only, 2=face and feet, 3=feet). Two additional parameters are *e*, which is the external temperature (the temperature outside the car), and *i*, which is the internal cabin temperature. At time *t*, we denoted the CCS setting as ω_t , the external temperature as e_t and the internal cabin temperature as i_t .

The NICE agent uses 3 models; a *CCS model*, a *human driver model* and an *Exponentron prediction model*. The construction of these models is described later in this section. The NICE agent uses the three models in the following manner: At time *t*, the NICE agent predicts the driver's desired cabin temperature for the next time frame, \hat{i}_{t+1} , using the Exponentron prediction model. Given \hat{i}_{t+1} , the agent calls the CCS model and receives and implements a CCS setting ω_t which is predicted to bring about \hat{i}_{t+1} . Given that no comment is presented by the driver during the next 15 seconds, the agent assumes that the Exponentron's prediction, \hat{i}_{t+1} , and the CCS setting ω_t suit the driver's preferences and the process is repeated. The *com* signal takes the value of 0 since no comment was given by the driver. The driver can interrupt the above process (which otherwise will continue throughout the entire ride) by providing feedback. If a comment (*c*) is given within 15 seconds of implementing ω_t , then the agent uses the human driver model to predict the driver's desired CCS setting, $\hat{\omega}_t$, and implements it instantaneously in the CCS. Then, the system maintains the new CCS setting until 15 seconds pass in which no further feedback is provided by the driver. Namely, if the driver provides another comment within 15 seconds of his last comment, the human driver's model is called on once again and the 15-second timer is re-set. Once 15 seconds pass without further com-

ments, the resulting cabin temperature is used to update the Exponentron's parameters. To that end, the *com* signal is set to 1. That is, the Exponentron's parameters can only be adjusted when the driver interacts with the agent. Figure 3 illustrates the agent's algorithmic scheme.

5.3.1 The CCS Model

Recall that the CCS model is used to determine which CCS setting ω_t will bring about the desired change in the internal cabin temperature over a course of 15 seconds. For that purpose, the model receives i_t , ω_{t-1} and \hat{i}_{t+1} .

In order to train the CCS model, thirty distinct CCS settings were selected such that their set temperature, *temp*, was lower than the initial cabin temperature, i_0 , at the time of the experiment. This property is required to enforce a cooling condition, which we examine in this study. We counter-balanced the selected CCS settings to account for the different possible ω s; namely, different *temp*, *f* and *d* values. Each CCS setting was manually configured to the CCS at the beginning of the trial. The cabin temperature, *i*, and the external temperature, *e*, were recorded every 15 seconds until the car's cabin temperature reached a steady state. Between every 2 consecutive experiments the car was turned off and the car's doors were left open for 10 minutes so as to simulate the initial conditions.

From the 30 trials we conducted over the course of 3 days, we recorded 657 measurements. Each of the measurements corresponds to a change in the car's cabin temperature, $i_{j+1} - i_j$, given e_j , and the CCS setting ω used in the trial. We fit the data using a simple linear regression model which yields the best fit out of the tested models¹¹. Namely, we constructed a model which, given i_t and ω , predicts i_{t+1} . To find a ω which is most likely to bring about the desired change, we iterate through all possible ω s. In the case of a tie, where more than a single CCS setting is expected to change the cabin temperature in the desired manner, the model outputs one of the CCS settings which is most similar to the previous CCS setting, ω_{t-1} . Recall that a CCS setting is a vector $\langle temp, f, d \rangle$, therefore similarity is easily defined. In this work we used the cosine similarity.

Using cross-validation, the learned model yields a mean absolute error of 0.15°C and a strong correlation coefficient of 0.9. In comparison, using the last cabin temperature change, $\Delta_j = i_j - i_{j-1}$, as an estimation for the next cabin temperature, $\hat{i}_{j+1} = i_j + \Delta_j$, yields a mean absolute error of 0.51 and a correlation coefficient of 0.3.

5.3.2 The Human Driver Model

Given a driver's comment, denoted c_0 , the human driver model is used to predict the driver's desired CCS settings. The model is based on multi-dimensional regression: At time *t* when a comment (denoted as *c*) is given (i.e. a button is pressed), the model predicts the desired CCS setting $\hat{\omega}_t$, given ω_t , e_t , i_t , c_0 and the last 2 previously provided comments, denoted c_1, c_2 .

In order to train the human driver model, we used the data collected in Section 4.2. Recall that in our experiment drivers were asked to interact with the newly designed natural interface while changing the CCS settings *manually*. The experiment recordings were translated into more than 100 vectors of the form \langle

¹⁰ Also known as Fanger's Predicted Mean Vote (PMV) criteria.

¹¹ We also examined other, more sophisticated modeling, for example using SVM with kernels. These models did not provide a significant improvement in prediction accuracy.

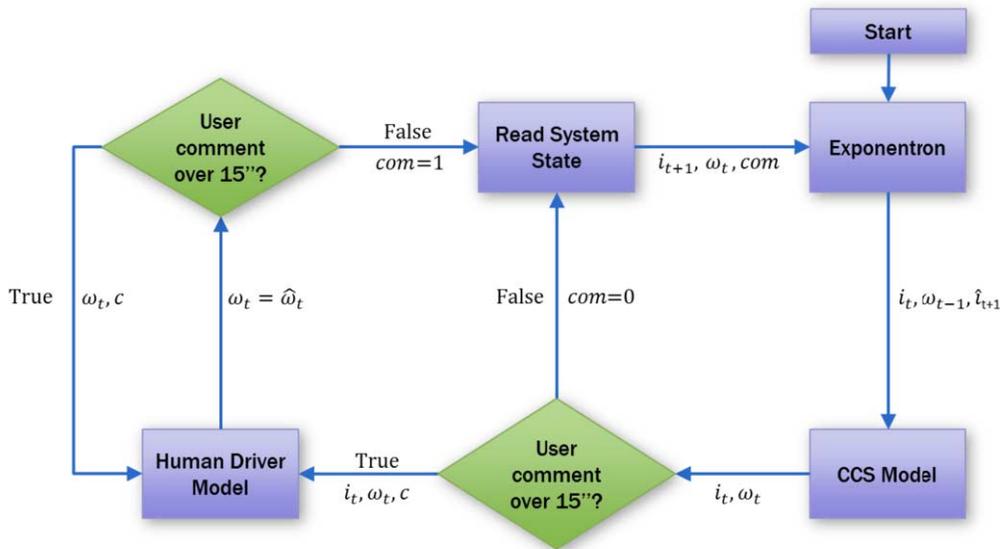


Figure 3: The NICE agent's algorithmic scheme.

$\omega_t, e_t, i_t, c_0, c_1, c_2$ > as described above, with the drivers' manually set CCS setting, ω' , as their label. Each session resulted in a different number of vectors, depending on the session's length.

The multi-dimensional regression consists of 3 linear regression models, each predicting a different component of the desired CCS setting $\omega' = \langle temp, f, d \rangle$. Using cross-validation, the prediction model yields a mean absolute error of 0.9 in predicting the next fan speed, f , and a mean absolute error of 1.02°C for the next set temperature, t . A high 97% accuracy in predicting the desired air delivery, d , was also recorded.

5.3.3 The Exponentron Prediction Model

The Exponentron prediction model implements the Exponentron algorithm (Algorithm 1). The Exponentron is trained with the same procedure used in Section 4.2. The model receives an additional input bit com , signaling whether the driver provided a comment in the last time frame. If and only if the com bit is 1, then an adaptation of the model parameters is executed using the current cabin temperature i_t .

The θ learned parameters represent the driver's preferences. Specifically, the parameter a represents the driver's intended steady state cabin temperature and the parameters b and c represent the way in which the driver wishes to bring about the desired cabin temperature.

5.4 Evaluation

5.4.1 Experimental Methodology

We recruited 24 drivers who did not participate in the data collection phase described in Section 4.2, with an equal number of males and females, ranging in age from 25 to 60 (average of 34). In a similar protocol to that described in Section 4.2, each subject was asked to enter a car that was parked in a garage, recreating the environmental conditions with temperatures ranging from 32°C to 37°C , averaging 35°C . Each subject participated in two consecutive trials. In each trial the subject was equipped with either the *Technical CCS* or the *NICE agent*. The technical CCS presented buttons similar to those

available in the common CCS, with which the driver can explicitly select her desired CCS setting. Namely, it presented two scales: one for the fan speed and the other for the temperature. The driver could change the setting by selecting her preferred fan speed and temperature on the designated scales. Namely, in a single interaction, the driver could change the CCS setting completely. Note that no intelligent agent was implemented to support it. The NICE agent uses the natural interface which is the same interface as described in Section 4.2. In both conditions, the GUI was presented on a tablet covering the car's central stack in order to avoid biasing the results. Each subject was instructed to interact with the system as she saw fit by using the buttons available in the presented interface. While in the car the subject was given a cell phone with a driving simulator "Bus Simulator 3D" to be played while the experiment was conducted.

Once the cabin temperature, i , reached a steady state, the session came to an end. Each session lasted 2-6 minutes (mean of 4 minutes). After the session ended, the driver was asked to exit the car for a period of 10 minutes while the car's doors were left open in order to simulate initial conditions. The process was repeated once more under the condition that was not examined in the first session. Subjects were counter-balanced as to which condition they experienced first in order to maintain the scientific integrity of the results.

During each session we recorded the number of interactions needed by the driver in order to reach her desired steady state. At the end of the experiment, drivers were asked to fill out a post-experiment questionnaire aimed at evaluating their satisfaction from the examined interfaces.

5.4.2 Results and Analysis

We first analyze the number of interactions needed by drivers to reach their desired steady states under the examined conditions. Then we summarize the subjects' answers in the post-experiment questionnaire. Note that the technical CCS is the current state-of-the-art CCS and acts as the benchmark in the following analysis.

The NICE agent required a significantly lower number of interactions from the driver compared to the technical CCS using t-test ($p < 0.05$). The NICE agent averaged 5.35 interactions carried out

by the driver until a steady state was reached while the technical CCS averaged 6.54 interactions. Out of the 24 subjects, only 8 subjects required more interactions while equipped with the NICE agent compared to their benchmark score. See Figure 4 for a summary.

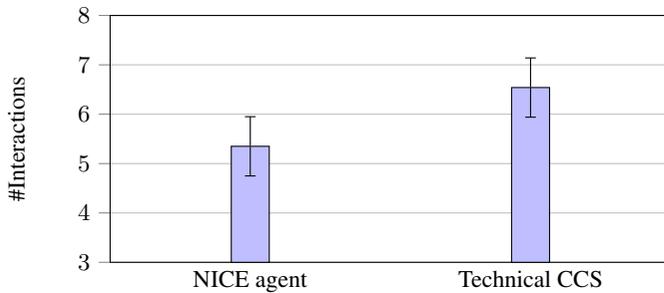


Figure 4: Average number of interactions per interface; (the lower the better). Error bars indicate standard errors.

Recall that the NICE agent’s goal is to automatically set and adjust the car’s CCS setting throughout the ride to the driver’s satisfaction. In order to assess the driver’s satisfaction from the interaction, at the end of the experiment we asked each driver which, if any, of the tested conditions she would want to see available in her car. Out of the 24 subjects, 13 subjects stated that they want to use the NICE agent, while 10 subjects stated their preference for the technical CCS. We also asked subjects to state their satisfaction level from the tested conditions. Subjects reported an average score of 6.2 out of 10 when asked for their satisfaction from the technical CCS. This result is significantly lower, using t-test ($p < 0.05$), compared to the NICE agent which recorded an average score of 7.2 out of 10.

To summarize, the results indicate that the NICE agent is able to reduce the number of interactions needed by drivers in order to achieve their desired comfort states (6.54 vs. 5.35) to the drivers’ satisfaction, as portrayed in the increase of the subjects’ subjective satisfaction (7.2 vs. 6.2) and the subjects’ preferred interaction mode (13 vs. 10).

6 Conclusions

In this paper we presented the Exponentron algorithm for the online prediction of assumed exponential decay time series. The Exponentron algorithm was evaluated both theoretically and empirically; theoretically we show a regret bound that compares our algorithm to the best batch algorithm, which is given the entire time series in advance. Empirically, the Exponentron was evaluated in synthetic and real-world prediction tasks in which it significantly outperformed classic time series prediction methods. Furthermore, we demonstrated the Exponentron algorithm’s benefit using the novel NICE agent, which significantly enhances the driver-automotive CCS interaction process compared to standard CCS control.

From an applicative perspective, our proposed methodology is not restricted to CCS-based agents. For example, in the development of personal assistance agents, the prediction of human forgetfulness may be beneficial. Specifically, an agent may need to predict the time it would take for its user to forget an important piece of information and provide a reminder for it. The forgetting curve [13] predicts the decline of memory retention in time and is assumed to decay exponentially for all people, though significant differences between individuals may be observed. Significant differences over time may also be presented for any specific person, which would necessitate online

adaptation. In a similar fashion to the natural interface, which was presented in Section 4.2, a user can express her feedback in a natural manner, e.g., “Remind me later”.

Future work will include the investigation of other time-dependent phenomena that are likely to adhere to an a priori assumed functional behavior. For example, we will tackle the challenge of automatically adjusting exercise levels in online tutoring systems. The progress in which new skills are learned is commonly assumed to follow a sigmoid curve, with some measure of skill on the Y axis and the number of trials on the X-axis [23]. In the spirit of the presented work and the NICE agent’s design, the agent will be able to adjust, online, the exercise level according to its estimation of the student’s learning curve. The student will be able to express her feedback in a natural manner, for example “The exercises are too difficult”, and thus make the agent adapt its behavior.

ACKNOWLEDGEMENTS

We would like to thank the ERC (grant #267523) for their support in this research.

REFERENCES

- [1] Oren Anava, Elad Hazan, Shie Mannor, and Ohad Shamir, ‘Online learning for time series prediction’, in *Proc. of the Conference on Learning Theory*, pp. 172–184, (2013).
- [2] ASHRAE, ‘Standard 55-2013. thermal environmental conditions for human occupancy. ashrae’, *American Society of Heating, Refrigerating and Air-Conditioning Engineering*, (2013).
- [3] Frederik Auffenberg, Sebastian Stein, and Alex Rogers, ‘A personalised thermal comfort model using a bayesian network’, in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, (2015).
- [4] Amos Azaria, Yaakov Gal, Sarit Kraus, and Claudia V Goldman, ‘Strategic advice provision in repeated human-agent interactions’, *Autonomous Agents and Multi-Agent Systems*, **30**(1), 4–29, (2016).
- [5] Amos Azaria, Ariel Rosenfeld, Sarit Kraus, Claudia V Goldman, and Omer Tsimhoni, ‘Advice provision for energy saving in an automobile climate-control system’, *AI Magazine*, **36**(3), 61–73, (2015).
- [6] José M Bernardo and Adrian FM Smith. Bayesian theory, 2001.
- [7] Dimitri P Bertsekas, ‘Nonlinear programming’, (1999).
- [8] Léon Bottou, ‘Stochastic gradient tricks’, *Neural Networks, Tricks of the Trade, Reloaded*, 430–445, (2012).
- [9] George Box, Gwilym M. Jenkins, and Gregory C. Reinsel, *Time Series Analysis: Forecasting and Control*, Prentice-Hall, 1994.
- [10] Cristiana Croitoru, Ilinca Nastase, Florin Bode, Amina Meslem, and Angel Dogeanu, ‘Thermal comfort models for indoor spaces and vehicles: current capabilities and future perspectives’, *Renewable and Sustainable Energy Reviews*, **44**, 304–318, (2015).
- [11] Sergio Damiani, Enrica Deregibus, and Luisa Andreone, ‘Driver-vehicle interfaces and interaction: where are they going?’, *European transport research review*, **1**(2), 87–96, (2009).
- [12] Norman Richard Draper, Harry Smith, and Elizabeth Pownell, *Applied regression analysis*, volume 3, Wiley New York, 1966.
- [13] Hermann Ebbinghaus, *Memory: A contribution to experimental psychology*, number 3, University Microfilms, 1913.
- [14] Noah Gans, Ger Koole, and Avishai Mandelbaum, ‘Telephone call centers: Tutorial, review, and research prospects’, *Manufacturing & Service Operations Management*, **5**(2), 79–141, (2003).
- [15] Everette S Gardner, ‘Exponential smoothing: The state of the art’, *Journal of forecasting*, **4**(1), 1–28, (1985).
- [16] Paul Goodwin, ‘The holt-winters approach to exponential smoothing: 50 years old and going strong’, *Foresight: The International Journal of Applied Forecasting*, (19), 30–33, (2010).
- [17] I. Guedj and A. Mandelbaum, ‘Call center data’, Technical report, Technion, Israel Institute of Technology, (2000).
- [18] Jee Yeon Hwang, Kent Larson, Ryan Chin, and Henry Holtzman, ‘Expressive driver-vehicle interface design’, in *Proceedings of the 2011 Conference on Designing Pleasurable Products and Interfaces*, p. 19. ACM, (2011).

- [19] Li Li, Ding Wen, Nan-Ning Zheng, and Lin-Cheng Shen, 'Cognitive cars: A new frontier for adas research', *Intelligent Transportation Systems, IEEE Transactions on*, **13**(1), 395–407, (2012).
- [20] Chenghao Liu, Steven CH Hoi, Peilin Zhao, and Jianling Sun, 'Online arima algorithms for time series prediction', in *Thirtieth AAAI Conference on Artificial Intelligence*, (2016).
- [21] Avishay Mandelbaum, Anat Sakov, and Sergey Zeltyn, 'Empirical analysis of a telephone call center', Technical report, Technion, Israel Institute of Technology, (2001).
- [22] Luis Moreira-Matias, Joao Gama, Michel Ferreira, João Mendes-Moreira, and Luis Damas, 'Predicting taxi-passenger demand using streaming data', *Intelligent Transportation Systems, IEEE Transactions on*, **14**(3), 1393–1402, (2013).
- [23] Jaap MJ Murre, 'S-shaped learning curves', *Psychonomic bulletin & review*, **21**(2), 344–356, (2014).
- [24] Isaac Newton, *Scala graduum caloris: calorum descriptiones & signa*, Royal Society of London, 1701.
- [25] Eoin O'Mahony and David B Shmoys, 'Data analysis and optimization for (citi) bike sharing', in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, (2015).
- [26] Ioannis Politis, Stephen Brewster, and Frank Pollick, 'To beep or not to beep?: Comparing abstract versus language-based multimodal driver displays', in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 3971–3980. ACM, (2015).
- [27] Simon Ramm, Joseph Giacomin, Duncan Robertson, and Alessio Malizia, 'A first approach to understanding and measuring naturalness in driver-car interaction', in *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pp. 1–10. ACM, (2014).
- [28] Ariel Rosenfeld, Amos Azaria, Sarit Kraus, Claudia V Goldman, and Omer Tsimhoni, 'Adaptive advice in automobile climate control systems', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 543–551. International Foundation for Autonomous Agents and Multiagent Systems, (2015).
- [29] Avi Rosenfeld, Zevi Bareket, Claudia V Goldman, Sarit Kraus, David J LeBlanc, and Omer Tsimhoni, 'Towards adapting cars to their drivers', *AI Magazine*, **33**(4), 46, (2012).
- [30] Avi Rosenfeld, Zevi Bareket, Claudia V Goldman, David J LeBlanc, and Omer Tsimhoni, 'Learning drivers behavior to improve adaptive cruise control', *Journal of Intelligent Transportation Systems*, **19**(1), 18–31, (2015).
- [31] Avraham Shvartzon, Amos Azaria, Sarit Kraus, Claudia V Goldman, Joachim Meyer, and Omer Tsimhoni, 'Personalized alert agent for optimal user performance', in *Proceedings of the 30th International Conference on Artificial Intelligence (AAAI)*. AAAI, (2016).
- [32] Gordon K Smyth, 'Nonlinear regression', *Encyclopedia of environmental metrics*, (2002).
- [33] Martin Zinkevich, 'Online convex programming and generalized infinitesimal gradient ascent', (2003).

Set-Valued Conditioning in a Possibility Theory Setting

Salem Benferhat and Amélie Levray and Karim Tabia¹ and Vladik Kreinovich²

Abstract. Possibilistic logic is a well-known framework for dealing with uncertainty and reasoning under inconsistent or prioritized knowledge bases. This paper deals with conditioning uncertain information where the weights associated with formulas are in the form of sets of uncertainty degrees. The first part of the paper studies set-valued possibility theory where we provide a characterization of set-valued possibilistic logic bases and set-valued possibility distributions by means of the concepts of compatible possibilistic logic bases and compatible possibility distributions respectively. The second part of the paper addresses conditioning set-valued possibility distributions. We first propose a set of three natural postulates for conditioning set-valued possibility distributions. We then show that any set-valued conditioning satisfying these three postulates is necessarily based on conditioning the set of compatible standard possibility distributions. The last part of the paper shows how one can efficiently compute set-valued conditioning over possibilistic knowledge bases.

1 INTRODUCTION

Possibilistic logic is a well-known framework for dealing with uncertainty, reasoning under inconsistent and prioritized knowledge bases and partial knowledge [25]. Many extensions have been proposed for possibilistic logic to deal for instance with imprecise certainty degrees [4, 5], symbolic certainty weights [6, 7], multi-agent beliefs [2], temporal and uncertain information [15], uncertain conditional events [10, 9, 11], generalized possibilistic logic [8, 19, 21], reasoning with justified beliefs [22], etc.

This paper proposes a new extension of possibilistic logic where the weights associated with formulas are in the form of sets of uncertainty degrees. Standard possibilistic logic expressions are propositional logic formulas associated with positive real degrees belonging to the unit interval $[0, 1]$. However, in practice it may be difficult for an agent to provide exact degrees associated with formulas of a knowledge base. This paper proposes an extension of standard possibility distributions and standard possibilistic bases where a set of possibility/certainty degrees may be associated with interpretations or formulas. A set of certainty degrees associated with a formula may represent the reliability levels of different sources that support the formula (see Example 1). Another important issue dealt with in this paper is the one of updating or conditioning a set-based knowledge base.

Conditioning is an important task for updating the current uncertain information when a new sure piece of information is received. A conditioning operator is designed to satisfy some desirable properties such as giving priority to the new information and ensuring minimal change while transforming an initial distribution into a conditional one. This paper deals with conditioning in a possibility theory and possibilistic logic frameworks [8, 14, 19, 13]. Conditioning in standard (single-valued) possibility theory has been addressed in many works [24, 27, 18, 23, 17, 3]. There are two major definitions of possibility theory: min-based (or qualitative) possibility theory and product-based (or quantitative) possibility theory. At the semantic level, these two theories share the same definitions, including the concepts of possibility distributions, necessity measures, possibility measures and the definition of normalization condition. However, they differ in the way they define possibilistic conditioning. This paper focuses on a so-called min-based conditioning [24] (or qualitative-based conditioning) which is appropriate in situations where only the ordering between events is important. In this case, the unit interval $[0, 1]$ is viewed as an ordinal scale where only the minimum and the maximum operations are used for propagating and updating uncertainty degrees.

The first contribution of this paper concerns the definition of a set-valued possibility theory which generalizes both standard possibility theory and interval-based possibility theory [4]. The second contribution deals with conditioning in a set-valued possibility theory setting. We first propose three natural postulates for a set-valued conditioning. We show that any set-valued conditioning satisfying these postulates is necessarily based on applying min-based conditioning on each compatible standard possibility distribution. We also provide the exact set of possibility degrees associated with min-based conditioning a set-valued distribution. The last contribution concerns efficient and syntactic computations of conditioning set-valued knowledge bases.

The rest of this paper is organized as follows: Section 2 provides a brief refresher on the possibility theory and possibilistic logic settings. Section 3 presents set-valued possibility theory and set-valued possibilistic logic. In Section 4, we focus on set-valued conditioning while Section 5 provides a syntactic computing of set-valued conditioning. Section 6 provides concluding discussions.

2 BRIEF REMINDER ON POSSIBILITY THEORY

Possibility distributions: Possibility theory [29, 20] is a well-known uncertainty theory. It is based on the concept of possibility distribution π which associates every state ω of the world Ω (the universe of discourse) with a degree in the interval $[0, 1]$ expressing a partial

¹ Univ Lille Nord de France, F-59000 Lille, France UArtois, CRIL - CNRS UMR 8188, F-62300 Lens, France, email: {benferhat, levray, tabia}@cril.univ-artois.fr

² Department of Computer Science, University of Texas at El Paso, 500 W. University El Paso, Texas 79968, USA, email: {vladik@utep.edu}

knowledge over the world. In this paper, Ω denotes the set of propositional interpretations. $\omega \models \phi$ means that ω is a model of (or satisfies) ϕ in the sense of propositional logic. The degree $\pi(\omega)$ represents the degree of compatibility (or consistency) of the interpretation ω with the available knowledge. By convention, $\pi(\omega)=1$ means that ω is fully consistent with the available knowledge, while $\pi(\omega)=0$ means that ω is impossible. $\pi(\omega) > \pi(\omega')$ simply means that ω is more compatible than ω' . A possibility distribution π is said to be normalized if there exists an interpretation ω such that $\pi(\omega)=1$, it is said to be subnormalized otherwise.

As it is already mentioned in the introduction, possibility degrees are interpreted either i) *qualitatively* (in min-based possibility theory) where only the *ordering* of the values matters, or ii) *quantitatively* (in product-based possibility theory) where the possibilistic scale $[0, 1]$ is quantitative as in probability theory. Min-based or qualitative possibility theory refers to the possibilistic setting where only the ordering induced by possibility degrees is important. In this setting, only the max and min operators are used for the reasoning and updating tasks.

Min-based conditioning: In the standard possibilistic setting, conditioning comes down to updating a possibility distribution π encoding the current knowledge when a completely sure event called *evidence* or *observation*, denoted by $\phi \subseteq \Omega$ is received. This results in a conditional possibility distribution denoted by $\pi(\cdot|\phi)$. There are many definitions of conditioning operators in the standard possibilistic setting [24, 27, 18, 23, 17].

Hisdal [24] proposed that a definition of a conditioning operator in the qualitative setting should satisfy the condition:

$$\forall \omega \models \phi, \pi(\omega) = \min(\pi(\omega|\phi), \Pi(\phi)).$$

Where $\Pi(\phi)$ denotes the possibility measure of an event ϕ , defined by:

$$\Pi(\phi) = \max\{\pi(\omega) : \omega \in \Omega, \omega \models \phi\}.$$

Dubois and Prade [16] proposed to select the largest conditional possibility distribution satisfying this condition, leading to the following conditioning operator.

Definition 1 (min-based conditioning). *Let π be a possibility distribution, $\phi \subseteq \Omega$ be a sure event. min-based conditioning of π by ϕ , simply denoted by $\pi(\cdot|_m\phi)$, is defined as:*

$$\forall \omega \in \Omega, \pi(\omega|_m\phi) = \begin{cases} 1 & \text{if } \pi(\omega) = \Pi(\phi) \text{ and } \omega \in \phi; \\ \pi(\omega) & \text{if } \pi(\omega) < \Pi(\phi) \text{ and } \omega \in \phi; \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

When $\Pi(\phi)=0$, then by convention $\forall \omega \in \Omega, \pi(\omega|_m\phi)=1$.

Possibilistic knowledge bases: A possibilistic formula is a pair (φ, α) where φ is a propositional logic formula and $\alpha \in [0, 1]$ is a certainty degree associated with φ . The higher the certainty degree α is, the more important is the formula φ . A possibilistic base $K = \{(\varphi_i, \alpha_i), i = 1, \dots, n\}$ is simply a set of possibilistic formulas.

A possibilistic knowledge base is a well-known compact representations of a possibility distribution. Given a possibilistic base K , we can generate a unique possibility distribution where interpretations ω satisfying all propositional formulas in K have the highest possible degree $\pi(\omega)=1$ (since they are fully consistent), whereas the others are pre-ordered with respect to the highest formulas they falsify. More formally:

Definition 2. *Let K be a possibilistic knowledge base. Then, the corresponding possibility distribution π_K is given by: $\forall \omega \in \Omega,$*

$$\pi_K(\omega) = \begin{cases} 1 & \text{if } \forall (\varphi, \alpha) \in K, \omega \models \varphi \\ 1 - \max\{\alpha_i : (\varphi_i, \alpha_i) \in K, \omega \not\models \varphi_i\} & \text{otherwise.} \end{cases} \quad (2)$$

The following lemma will be helpful for establishing proofs of some propositions. It states that 'zero-weighted' formulas can be added or removed from possibilistic knowledge bases without changing their distributions.

Lemma 1. *Let K be a possibilistic knowledge base K such that $(\delta, 0) \in K$. Let $K' = K \setminus \{(\delta, 0)\}$. Then $\forall \omega \in \Omega, \pi_K(\omega) = \pi_{K'}(\omega)$.*

This lemma can be easily shown since if a formula δ has a certainty degree equal to 0, then if there is an interpretation ω that falsifies only the formula δ then, according to Definition 2, the possibility degree associated to ω will be $1-0=1$.

An important notion that plays a central role in the inference process and conditioning is the one of α -cut. Let α be a positive real number. An α -cut is a set of propositional formulas defined by $K_{\geq \alpha} = \{\varphi : (\varphi, \beta) \in K \text{ and } \beta \geq \alpha\}$.

The concept of α -cut can be used to provide the syntactic counterpart of conditioning a possibilistic knowledge base with a propositional formula:

Definition 3. *Let K be a possibilistic knowledge and ϕ be a sure piece of information. The result of conditioning K by ϕ , denoted K_ϕ is defined as follows:*

$$K_\phi = \{(\phi, 1)\} \cup \{(\varphi, \alpha) : (\varphi, \alpha) \in K \text{ and } K_{\geq \alpha} \wedge \phi \text{ is consistent.}\}$$

Namely, K_ϕ is obtained by considering ϕ with a certainty degree '1', plus weighted formulas (φ, α) of K such that their α -cut is consistent with ϕ . It can be checked that:

$$\forall \omega \in \Omega, \pi_{K_\phi}(\omega) = \pi_K(\omega|_m\phi),$$

where π_K and π_{K_ϕ} are given using Definition 2 and $\pi_K(\cdot|_m\phi)$ is obtained using Definition 1.

Next section generalizes standard possibility theory and possibilistic logic into a set-valued setting.

3 SET-VALUED POSSIBILITY THEORY AND SET-VALUED POSSIBILISTIC LOGIC

Let us first start with a short example to motivate our extension.

Example 1. *Suppose we are interested in the amenities and facilities of a hotel in Paris to organize a conference. For this, we posted a question on a specialized Internet platform. To simplify, the question was about the presence of a large conference room in the hotel (represented by the variable c) and if the hotel has a great restaurant (represented by the variable r) to host the gala dinner. We also asked people to specify how certain of the answers they are, using a unit scale $[0, 1]$. Assume that we got three answers of three people: p_1 is a former hotel employee, the second, p_2 , is an employee of the Paris tourism office and the third, p_3 , is a client of the hotel. The certainty levels of these people with respect to different scenarios³ are summarized as follows:*

³ In this example, the scenario cr means that the hotel has a conference room and has a great restaurant while the scenario $c \neg r$ means that the hotel has a conference room but does not have a great restaurant.

Table 1. Example of multiple sources information

	p_1	p_2	p_3
cr	1	1	1
$\neg cr$	1	1	1
$c\neg r$.3	.2	.4
$\neg c\neg r$.4	.4	.4

In this example, the confidence degrees provided by the responders can be viewed as possibility degrees. Now, suppose that we got hundreds or thousands of answers or suppose that there is a large number of variables, then it will be interesting to find a compact way to encode the obtained answers and more importantly to reason with them (answer any request of interest and update the available information when new sure information is obtained). Set-valued possibility theory is especially tailored to this type of information.

Let us now introduce the concept of set-valued possibility distribution.

3.1 Set-valued possibility distributions

In the set-valued possibilistic setting, the available knowledge is encoded by a set-valued possibility distribution $S\pi$ where each state ω is associated with a finite set $S\pi(\omega)$ of possible values of possibility degrees $\pi(\omega)$.

If S is a set, then we denote by \bar{S} and \underline{S} the maximum and minimum values of S respectively. When all S 's associated with interpretations (or formulas) are singletons (meaning that $\bar{S} = \underline{S}$), we refer to standard distributions (resp. standard possibilistic bases). Here, $\underline{S\pi}(\omega)$ (resp. $\bar{S\pi}(\omega)$) denotes the minimum (resp. maximum) of the possibility degrees of ω .

Clearly, set-valued possibility theory is also an extension of interval-based possibility theory [4], where the set is denoted as an interval of possible values. Therefore, we now consider sets of degrees and we define a set-valued possibility distribution as follows:

Definition 4 (Set-valued possibility distribution). *A set-valued possibility distribution $S\pi$ is a mapping $S\pi : \Omega \rightarrow \mathcal{S}$ from the universe of discourse Ω to the set \mathcal{S} of all sub-sets included in the interval $[0, 1]$, with the normalization property requiring that $\max_{\omega \in \Omega} \bar{S\pi}(\omega) = 1$.*

The information corresponding to Example 1 could be compactly encoded as follows:

Example 2. (Example 1 cont'd.) *Let us represent the available knowledge from Example 1 as a set-valued possibility distribution given in Table 2.*

Table 2. Set-valued distribution corresponding to the multiple source information of Table 1.

	$S\pi$
cr	$\{1\}$
$\neg cr$	$\{1\}$
$c\neg r$	$\{.2, .3, .4\}$
$\neg c\neg r$	$\{.4\}$

As in an interval-based possibility theory [4], we also interpret a set-valued possibility distribution as a family of compatible standard possibility distributions defined by:

Definition 5. *Let $S\pi$ be a set-valued possibility distribution. A normalized possibility distribution π is said to be compatible with $S\pi$ if and only if $\forall \omega \in \Omega, \pi(\omega) \in S\pi(\omega)$.*

As shown in Example 3, compatible distributions are not unique. We denote by $\mathcal{C}(S\pi)$ the set of all possibility distributions compatible with $S\pi$.

Example 3. *Let $S\pi$ be a set-valued possibility distribution described in the Table 3.*

Then following Definition 5, the possibility distributions π_1 and π_2 (from Table 3) are compatible with $S\pi$.

However, π_3 is not compatible with $S\pi$ since $\pi_3(cr) = .4 \notin S\pi(cr) = \{1\}$.

Table 3. Example of set-valued possibility distribution $S\pi$, compatible possibility distributions π_1 and π_2 and a non compatible one π_3 .

$\omega \in \Omega$	$S\pi$	$\omega \in \Omega$	π_1	π_2	π_3
cr	$\{1\}$	cr	1	1	.4
$\neg cr$	$\{1\}$	$\neg cr$	1	1	1
$c\neg r$	$\{.2, .3, .4\}$	$c\neg r$.3	.4	.2
$\neg c\neg r$	$\{.4\}$	$\neg c\neg r$.4	.4	.4

Let us now see how to generalize standard possibilistic logic into a set-valued setting.

3.2 Set-valued possibilistic logic

Contrary to standard possibilistic logic where the uncertainty is described with single values, set-valued possibilistic logic uses sets. The syntactic representation of set-valued possibilistic logic generalizes the notion of a possibilistic base to a set-valued possibilistic knowledge base as follows:

Definition 6. *A set-valued possibilistic knowledge base, denoted by SK , is a set of propositional formulas associated with sets:*

$$SK = \{(\varphi, S), \varphi \in \mathcal{L} \text{ and } S \text{ is a set of degrees in } [0, 1]\}$$

In Definition 6, $\varphi \in \mathcal{L}$ denotes again a formula of a propositional language \mathcal{L} .

A set-valued possibilistic base SK can be viewed as a family of standard possibilistic bases called compatible bases. More formally:

Definition 7 (Compatible possibilistic base). *A possibilistic base K is said to be compatible with a set-valued possibilistic base SK if and only if K is obtained from SK by replacing each set-valued formula (φ, S) by a standard possibilistic formula (φ, α) with $\alpha \in S$.*

In other words, each compatible possibilistic base is such that $K = \{(\varphi, \alpha) : (\varphi, S) \in SK \text{ and } \alpha \in S\}$.

We also denote by $\mathcal{C}(SK)$ the finite set of all compatible possibilistic bases associated with a set-valued possibilistic base SK .

Example 4. *In the following, we will use this set-valued possibilistic knowledge base to illustrate our propositions. Let SK be a set-valued possibilistic knowledge base such that:*

$$SK = \{(\neg c \vee r, \{.4, .7, .8\}), (r, \{.6\})\}.$$

An example of a compatible possibilistic knowledge base is:

$$K = \{(\neg c \vee r, .4), (r, .6)\}.$$

As in standard possibilistic logic, a set-valued knowledge base SK is also a compact representation of a set-valued possibility distribution $S\pi_{SK}$.

3.3 From set-valued possibilistic bases to set-valued possibility distributions

Let us go one step further with the contribution on how to compute the set-valued possibility distribution from a set-valued base.

Let $SK = \{(\varphi_i, S_i) : i=1, \dots, n\}$ be a set-valued possibilistic knowledge base. A natural way to define a set-valued possibility distribution, associated with SK and denoted by $S\pi_{SK}$, is to consider all standard possibility distributions associated with each compatible knowledge base. Namely:

Definition 8. Let SK be a set-valued possibilistic knowledge base. The set-valued possibility distribution $S\pi_{SK}$ associated with SK is defined by:

$$\forall \omega \in \Omega, S\pi_{SK}(\omega) = \{\pi_K(\omega) : K \in \mathcal{C}(SK)\}.$$

Recall that $\mathcal{C}(SK)$ is the set of compatible knowledge bases (given in Definition 7) and π_K is given by Definition 2.

Similar to the single valued possibilistic logic setting, we can get rid of some formulas of a set-valued knowledge base without any information loss. More precisely, we can ignore any formula of SK attached with only one certainty degree equal to zero, as stated in the following lemma.

Lemma 2. Let SK be a set-valued possibilistic base such that $(\delta, \{0\}) \in SK$. Let $SK' = SK \setminus \{(\delta, \{0\})\}$. Then $\forall \omega \in \Omega, S\pi_{SK}(\omega) = S\pi_{SK'}(\omega)$.

Lemma 2 is again useful for establishing proofs of some propositions. The idea behind this lemma stands in the definition of compatible bases and Lemma 1. Indeed, in the case where SK is such that $(\delta, \{0\}) \in SK$, then in every compatible base K , we have $(\delta, 0) \in K$, therefore, as stated in Lemma 1, the weighted formula $(\delta, 0)$ can be ignored from K without changing its associated distributions, and this can be generalized to the set-valued formula $(\delta, \{0\})$.

Let us now characterize $S\pi_{SK}$. The following proposition provides the conditions under which the highest possibility degree '1' belongs to $S\pi_{SK}(\omega)$:

Proposition 1. Let SK be a set-valued possibilistic knowledge base. Let ω be an interpretation. Then:

$$1 \in S\pi_{SK}(\omega) \text{ iff } \omega \models \bigwedge \{\varphi : (\varphi, S) \in SK \text{ and } \underline{S} > 0\}$$

Namely, $1 \in S\pi_{SK}(\omega)$ if and only if ω satisfies all formulas having a strictly positive certainty degree.

Proof. Recall that $1 \in S\pi_{SK}(\omega)$ means that there exists a compatible possibilistic base $K \in \mathcal{C}(SK)$ such that $\pi_K(\omega) = 1$. Now, formulas of K having a certainty degree equal to '0' can be removed, thanks to Lemma 1, without changing π_K . The fact that $\pi_K(\omega) = 1$ implies that ω is a model of $\{\varphi : (\varphi, \alpha) \in K, \alpha > 0\}$. This also means that ω is also a model of $\{\varphi : (\varphi, S) \in SK, \underline{S} > 0\}$.

Let us now show the converse. Assume that ω is a model of $\{\varphi : (\varphi, S) \in SK, \underline{S} > 0\}$. Let K be a compatible possibilistic knowledge base obtained from SK by replacing each set-valued S by its lower bound \underline{S} . Clearly, $\{\varphi : (\varphi, \underline{S}) \in K\}$ is satisfied by ω . Hence, $1 \in S\pi_{SK}(\omega)$. \square

Example 5. (Example 4 cont'd) Let us continue with the knowledge base from Example 4. Recall that

$$SK = \{(\neg c \vee r, \{.4, .7, .8\}), (r, \{.6\})\}$$

Following Proposition 1, interpretations cr and $\neg cr$ will have among their possibility degrees the degree 1 (namely $1 \in S\pi_{SK}(cr)$ and $1 \in S\pi_{SK}(\neg cr)$) since these interpretations are models of all the formulas of SK attached only to strictly positive degrees.

We now study under which conditions a possibility degree $(1-\alpha)$ belongs to $S\pi_{SK}(\omega)$, with $\alpha \in [0, 1]$. Clearly, if $(1-\alpha) \in S\pi(\omega)$ then there exists a compatible base K such that $\pi_K(\omega) = 1-\alpha$. Hence, there exists $(\varphi, \alpha) \in K$ such that $\omega \not\models \varphi$. Then there exists $(\varphi, S) \in SK$ such that $\omega \not\models \varphi$ and $\alpha \in S$.

To determine the possible values of $S\pi_{SK}(\omega)$, it is enough to browse all certainty degrees associated with formulas of SK falsified by ω and check whether their inverse will belong or not to $S\pi_{SK}(\omega)$.

This is precisely specified by the following proposition:

Proposition 2. Let ω be an interpretation. Let $A = \bigcup \{S : (\varphi, S) \in SK, \omega \not\models \varphi\}$. Let $a \in A \cup \{0\}$. Then,

$$(1-a) \in S\pi_{SK}(\omega) \text{ iff } \omega \models \{\varphi : (\varphi, S) \in SK, \underline{S} > a\}$$

Proof. Proposition 2 recovers Proposition 1 in case where $a=0$. Hence, we only focus on the case $a>0$. To see the proof, assume that $a>0$ and $(1-a) \in S\pi_{SK}(\omega)$. This means that there exists a compatible possibilistic knowledge base $K \in \mathcal{C}(SK)$, such that $\pi_K(\omega) = 1-a$.

This means that $\{\varphi : (\varphi, b), b > a\}$ is consistent and satisfied by ω . Since $\{\varphi : (\varphi, S), \underline{S} > a\} \subseteq \{\varphi : (\varphi, b), b > a\}$, this also means that $\{\varphi : (\varphi, S), \underline{S} > a\}$ is consistent and satisfied by ω .

Let us show the converse. Assume that $\omega \models \{\varphi : (\varphi, S), \underline{S} > a\} \wedge \omega$. Clearly, if $A = \emptyset$ (namely, $a=0$) or $A = \{0\}$ then whatever is the compatible base K , ω will satisfy each formula in K , hence $\pi_K(\omega) = 1$, and $(1-a) \in S\pi_{SK}(\omega)$. Assume that $a \in A$ and $a > 0$. Let (φ_1, S_1) be a formula of SK such that $a \in S_1$ and $\omega \not\models \varphi_1$. Let K be a compatible base defined by:

$$K = \{(\varphi, \underline{S}) : (\varphi, S) \in SK, \varphi \neq \varphi_1\} \cup \{(\varphi_1, a)\}.$$

Namely, K is obtained from SK by replacing S by \underline{S} for each formula in SK , except for φ_1 where a is used instead of \underline{S} . It is easy to see that K is compatible with SK , namely $K \in \mathcal{C}(SK)$. It is also easy to see that $\pi_K(\omega) = 1-a$, since $\{\varphi : (\varphi, b) \in K, b > a\}$ is satisfied by ω , $\{\varphi : (\varphi, b) \in K, b > a\} \cup \{(\varphi_1, a)\}$ is falsified by ω . Therefore $(1-a) \in S\pi_{SK}(\omega)$. \square

Let us continue our example, and illustrate Proposition 2.

Example 6. (Example 4 cont'd) We need to check which degrees belong to $S\pi_{SK}(\omega)$. For each interpretation, we first compute $A = \bigcup \{S : (\varphi, S) \in SK, \omega \not\models \varphi\}$. For instance, let us consider $\omega = c \neg r$ then $A = \{.4, .7, .8, .6\}$. Now, let us analyse each value a of $A \cup \{0\}$,

- For $a=0$, $c \neg r \not\models \{\neg c \vee r, r\}$, then $1 \notin S\pi_{SK}(c \neg r)$;
- For $a=.4$, $c \neg r \not\models \{r\}$, then $.6 \notin S\pi_{SK}(c \neg r)$;
- For $a=.7$, $\emptyset \wedge c \neg r$ is consistent, then $.3 \in S\pi_{SK}(c \neg r)$;
- For $a=.8$, $\emptyset \wedge c \neg r$ is consistent, then $.2 \in S\pi_{SK}(c \neg r)$;
- Finally, for $a=.6$, $\emptyset \wedge c \neg r$ is consistent, then $.4 \in S\pi_{SK}(c \neg r)$.

Then we can conclude that $S\pi_{SK}(c \neg r) = \{.2, .3, .4\}$.

Let us take another interpretation, for instance $\omega = \neg c \neg r$. Then $A = \{.6\}$ and for each $a \in A \cup \{0\}$,

- For $a=0$, $\neg c \rightarrow r \neq \{\neg c \vee r, r\}$, then $1 \notin S\pi_{SK}(\neg c \rightarrow r)$;
- And for $a=.6$, $\emptyset \wedge \neg c \rightarrow r$ is consistent, then $.4 \in S\pi_{SK}(\neg c \rightarrow r)$.

We can conclude that $S\pi_{SK}(\neg c \rightarrow r) = \{.4\}$.

The whole distribution is exactly the one given in Example 2.

Let us now deal with the issue of conditioning a set-valued possibilistic base. The following section extends min-based conditioning to set-valued possibility distributions.

4 CONDITIONING SET-VALUED POSSIBILISTIC INFORMATION

Before providing our extension of min-based conditioning to the set-valued setting, let us first focus on the natural properties that a set-valued conditioning operator should fulfill.

4.1 Three natural requirements for the set-valued conditioning

The first natural requirement (called recovering standard conditioning) is that in the *degenerate* case, namely when each set $S\pi(\omega)$ contains exactly one single degree $\pi(\omega)$, the result of the new conditioning procedure should coincide with the result $\pi(\cdot|_m\phi)$ of the original conditioning procedure (Definition 1). For each possibility distribution π , by $\{\pi(\omega)\}$ we denote its set-valued representation, i.e., a set-valued possibility distribution for which, for every $\omega \in \Omega$, we have $S\pi(\omega) = \{\pi(\omega)\}$. In these terms, the above requirement takes the following form:

S1. If for every $\omega \in \Omega$, we have $S\pi(\omega) = \{\pi(\omega)\}$, then $S\pi(\omega|\phi) = \{\pi(\omega|_m\phi)\}$ for all ω and ϕ .

The second requirement (called specificity) is related to the fact that we do not know the precise values $S\pi(\omega)$ since we only have partial information about them. In principle, if we can get some additional information about these values, then this would lead, in general, to narrower sets (indeed, the cardinality of a set captures the ignorance regarding the exact value of $\pi(\omega)$). Let us define the concepts of specificity between set-valued possibility distribution:

Definition 9. Let $S\pi$ and $S\pi'$ be two set-valued possibility distributions. Then $S\pi$ is said to be more specific than $S\pi'$, denoted $S\pi \subseteq S\pi'$, if $S\pi(\omega) \subseteq S\pi'(\omega)$ holds for all $\omega \in \Omega$.

S2. If $S\pi(\omega) \subseteq S\pi'(\omega)$ for all ω , then $S\pi(\omega|\phi) \subseteq S\pi'(\omega|\phi)$ for all ω .

Of course, these two postulates are not sufficient. For example, we can take $S\pi(\cdot|\phi) = \{\pi(\cdot|_m\phi)\}$ for degenerate set-valued possibility distributions and $S\pi(\omega|\phi) = [0, 1]$ for any other set-valued distribution $S\pi$. To avoid such extensions, it is reasonable to impose the following minimality condition:

S3. There does not exist a conditioning operation $'|_1'$ that satisfies both properties **S1–S2** and for which:

- $S\pi(\omega|_1\phi) \subseteq S\pi(\omega|\phi)$ for all $S\pi$, ω , and ϕ ,
- $S\pi(\omega|_1\phi) \neq S\pi(\omega|\phi)$ for some $S\pi$, ω , and ϕ .

S3 is called minimality condition. The following theorem provides one of our main results where we show that there is only one set-valued conditioning satisfying **S1–S3** and where the set conditional possibility degree $S\pi(\omega|\phi)$ is defined as the closure of the set of all $\pi(\cdot|_m\phi)$, where π is compatible with $S\pi$.

Theorem 1. There exists exactly one set-valued conditioning, also denoted by $S\pi(\cdot|\phi)$ for sake of simplicity, that satisfies the properties **S1–S3**, and which is defined by: $\forall \omega \in \Omega$,

$$S\pi(\omega|\phi) = \{\pi(\omega|_m\phi) : \pi \in \mathcal{C}(S\pi)\} \quad (3)$$

where $|_m$ is the min-based conditioning given in Definition 1.

Proof. 1°. Let us denote the corresponding set-based conditioning by $S\pi(\cdot|\phi)$. We need to prove:

- that this closure $S\pi(\cdot|\phi)$ satisfies the properties **S1–S3**, and
- that every operation $S\pi(\cdot|_1\phi)$ that satisfies the properties **S1–S3** coincides with the set-conditioning $S\pi(\cdot|\phi)$.

2°. One can easily see that the operation $S\pi(\cdot|\phi)$ satisfies the properties **S1–S2**.

3°. Let us now prove that if an operation $S\pi(\cdot|_1\phi)$ satisfies the properties **S1–S2**, then for every $S\pi$ and ϕ , we have $S\pi(\cdot|\phi) \subseteq S\pi(\cdot|_1\phi)$.

Then, for every distribution $\pi \in \mathcal{C}(S\pi)$, we have $\{\pi\} \subseteq S\pi$ and thus, due to the postulate **S2**, we have $\{\pi\}(\cdot|_1\phi) \subseteq S\pi(\cdot|\phi)$. By the property **S1**, we have $\{\pi\}(\omega|_1\phi) = \{\pi(\omega|_m\phi)\}$. Thus, the above inclusion means that $\pi(\cdot|_m\phi) \in S\pi(\cdot|_1\phi)$.

The set $S\pi(\omega|_1\phi)$ therefore contains all the values $\pi(\omega|_m\phi)$ corresponding to all possible $\pi \in \mathcal{C}(S\pi)$:

$$\{\pi(\omega|_m\phi) : \pi \in \mathcal{C}(S\pi)\} \subseteq S\pi(\omega|_1\phi).$$

Thus, we conclude that $S\pi(\omega|\phi) \subseteq S\pi(\omega|_1\phi)$ for all ω .

The statement is proven.

4°. We can now prove that $S\pi(\cdot|\phi)$ also satisfies the property **S3**.

Indeed, if there is some other operation $|_1$ that satisfies **S1** and **S2**, and for which $S\pi(\omega|_1\phi) \subseteq S\pi(\omega|\phi)$ for all ω , then, since we have already proven the opposite inclusion in Part 3 of this proof, we conclude that $S\pi(\omega|_1\phi) = S\pi(\omega|\phi)$ for all ω , so indeed no narrower conditioning operation is possible.

5°. To complete the proof, let us show that if some $S\pi(\cdot|_1\phi)$ satisfies the properties **S1–S3**, then it coincides with $S\pi(\cdot|\phi)$.

Indeed, by Part 3 of this proof, we have $S\pi(\omega|\phi) \subseteq S\pi(\omega|_1\phi)$ for all ω . If we had $S\pi(\omega|\phi) \neq S\pi(\omega|_1\phi)$ for some ω and ϕ , this would contradict the minimality property **S3**. Thus, indeed, $S\pi(\cdot|\phi) = S\pi(\cdot|_1\phi)$. Uniqueness is proven, and so is for the theorem. \square

4.2 Analyzing set-based conditioning

Now, we can go one step beyond Theorem 1 and provide the exact contents of the conditioned set $S\pi(\cdot|_m\phi)$. Let us first start with the following lemma which delimits the set of possible values associated with models of ϕ after the conditioning operation.

Lemma 3. Let $S\pi$ be a set-valued possibility distribution. Let $\phi \subseteq \Omega$. Then $\forall \omega \in \Omega$,

- If $\omega \neq \phi$, $S\pi(\omega|\phi) = \{0\}$,
- And if $\omega = \phi$, $S\pi(\omega|\phi) \subseteq S\pi(\omega) \cup \{1\}$.

The proof of this lemma is immediate. Indeed, if π is a standard possibility distribution, then by definition $\pi(\omega|_m\phi)$ is either equal to $\pi(\omega)$ or to 1 for models of ϕ . Hence, the only admissible values for $S\pi(\omega|\phi)$ are those in $S\pi(\omega)$ and the value 1. For counter-models of ϕ (namely, $\omega \neq \phi$), then clearly $S\pi(\omega|\phi) = \{0\}$ since $\pi(\omega|_m\phi) = 0$ for each compatible distributions π .

Given this lemma, we need to answer two questions:

- Under which conditions does the fully possibility degree 1 belong to $S\pi(\omega|\phi)$?
- Under which conditions will a given possibility degree $a \in S\pi(\omega)$ still belong to $S\pi(\omega|\phi)$?

The answer to these questions is given in the following proposition:

Proposition 3. *Let $S\pi$ be a set-valued possibility distribution. Let $\phi \subseteq \Omega$.*

- i) $1 \in S\pi(\omega|\phi)$ iff $\forall \omega' \neq \omega, \overline{S\pi}(\omega) \geq \underline{S\pi}(\omega')$.
- ii) Let $a \in S\pi(\omega)$ (with $a \neq 1$). Then $a \in S\pi(\omega|\phi)$ iff $\exists \omega' \neq \omega, \overline{S\pi}(\omega') > a$.

Proof. For item (i) assume that $1 \in S\pi(\omega|\phi)$. This means that there exists a compatible distribution π of $S\pi$ such that $\pi(\omega|\phi) = 1$. This also means that $\forall \omega' \neq \omega, \pi(\omega) \geq \pi(\omega')$. Since, $\overline{S\pi}(\omega) \geq \pi(\omega)$, and $\pi(\omega') \geq \underline{S\pi}(\omega')$, hence we have $\forall \omega' \neq \omega, \overline{S\pi}(\omega) \geq \underline{S\pi}(\omega')$. For the converse, assume that $\forall \omega', \overline{S\pi}(\omega) \geq \underline{S\pi}(\omega')$. Let π be a compatible distribution such that $\pi(\omega) = \overline{S\pi}(\omega)$ and $\forall \omega' \neq \omega, \pi(\omega') = \underline{S\pi}(\omega')$. Clearly, $\forall \omega' \neq \omega, \pi(\omega) > \pi(\omega')$. Hence $\pi(\omega|\phi) = 1$ and $1 \in S\pi(\omega|\phi)$.

For item (ii), let $a \in S\pi(\omega)$ where $a \neq 1$. Assume that $\exists \omega' \neq \omega$, such that $\overline{S\pi}(\omega') > a$. Consider a compatible distribution π where $\pi(\omega') = \overline{S\pi}(\omega')$ and $\pi(\omega) = a$. Then clearly, $\pi(\omega|\phi) = a \in S\pi(\omega|\phi)$. For the converse, assume that $a \in S\pi(\omega|\phi)$ and $a \neq 1$. This means that there exists a compatible distribution π such that $\pi(\omega|\phi) = a < 1$. Hence, $\exists \omega', \pi(\omega) = a < \pi(\omega')$. Since $\pi(\omega') \leq \overline{S\pi}(\omega')$ this means that $\overline{S\pi}(\omega') > a$. □

Example 7. *In this example, we deal with conditioning a set-valued possibility distribution. Therefore, let us continue Example 2 and assume that the manager of the hotel tells us that the restaurant of the hotel has closed down definitively a few weeks ago. Then we need to condition with the new piece of information $\phi = \neg r$. Let us run the conditioning operation step by step. For every interpretation model of ϕ ,*

- For $\omega = c \neg r$,
 - i) since, with $\omega' = \neg c \neg r, .4 \geq .4$, then $1 \in S\pi(c \neg r | \neg r)$;
 - ii) For $a = .2$, since, $\overline{S\pi}(\neg c \neg r) = .4 > .2$, then $.2 \in S\pi(c \neg r | \neg r)$.
For $a = .3$, since, $\overline{S\pi}(\neg c \neg r) = .4 > .2$, then $.3 \in S\pi(c \neg r | \neg r)$.
For $a = .4$, since, $\overline{S\pi}(\neg c \neg r) = .4 \not> .4$, then $.4 \notin S\pi(c \neg r | \neg r)$.
- For the interpretation $\omega = \neg c \neg r$, we follow the same computation steps.
- For counter-models of $\neg r$, we have $S\pi(\omega|\phi) = \{0\}$.

Given the distribution in Table 2, we sum up the result of conditioning this distribution in Table 4.

Table 4. Set-valued distribution $S\pi$ of Example 2 conditioned by $\phi = \neg r$.

	$S\pi(. \phi)$
cr	$\{0\}$
$\neg cr$	$\{0\}$
$c \neg r$	$\{.2, .3, 1\}$
$\neg c \neg r$	$\{1\}$

5 SYNTACTIC COUNTERPART OF SET-VALUED CONDITIONING

Let us first consider again conditioning a standard possibilistic knowledge base K and rewrite the result of conditioning K . Recall that $K_{\geq a} = \{\varphi : (\varphi, \alpha) \in K \text{ and } \alpha \geq a\}$ be a set of propositional formulas from K having a weight greater or equal to a . Then, the result of conditioning K by ϕ , denoted by K_ϕ , given by Definition 3 can be rewritten as:

$$K_\phi = \{(\phi, 1)\} \cup \{(\varphi, \alpha) : (\varphi, \alpha) \in K_{\geq \alpha} \wedge \phi \text{ is consistent}\} \cup \{(\varphi, 0) : (\varphi, \alpha) \in K_{\geq \alpha} \wedge \phi \text{ is inconsistent}\}.$$

The only difference with Definition 3 is that '0' weighted formulas have been added. This has no influence thanks to Lemma 1. Namely, K_ϕ is obtained from K by adding ϕ with a fully certainty degree and ignore some formulas from K . By ignoring some formulas, we mean the certainty degrees of these formulas are set to '0'.

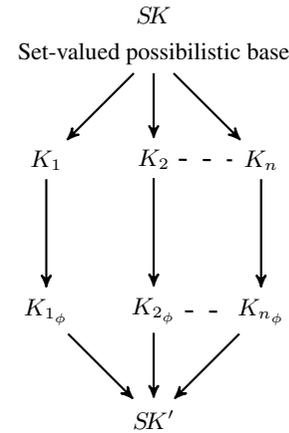


Figure 1. Compatible-based conditioning

The aim of this section is to provide syntactic computation of set-valued conditioning when set-valued possibility distributions are compactly represented by set-valued possibilistic knowledge bases. As illustrated in Figure 1, the input is an initial set-valued knowledge base SK and a formula ϕ . The output is a new set-valued knowledge base SK' that results from conditioning the set of all compatible bases of SK with ϕ . This new set-valued knowledge base SK' is obtained by considering the set of all compatible possibilistic knowledge bases, $K_i \in \mathcal{C}(SK)$. More precisely, it is done in three steps:

- First, from SK we generate the set of compatible bases K_1, K_2, \dots, K_n
- then, we condition each compatible base K_i with ϕ . The result is K_{i_ϕ} and obtained using Definition 3.
- Lastly, we define SK' by associating with each formula φ of SK the set of degrees present in at least one conditioned K_{i_ϕ} .

Namely: $SK' = \{(\varphi, S) : S = \bigcup \{\alpha_k : (\varphi, \alpha_k) \in K_\phi, K \in \mathcal{C}(SK)\}\}$.

Hence, a naive algorithm for computing SK' is given.

Algorithm 1 Naive computation of SK'

Input: SK : a set-valued knowledge base
 ϕ : a propositional formula
Output: SK' : the result of conditioning SK with ϕ

```

 $SK' \leftarrow \{(\phi, 1)\}$ 
foreach  $(\gamma, S) \in SK$  do
   $S' \leftarrow \emptyset$ 
  foreach  $K$  compatible with  $SK$  do
    Compute  $K_\phi$ 
     $S' \leftarrow S' \cup \{\alpha : (\gamma, \alpha) \in K_\phi\}$ 
  end foreach
   $SK' \leftarrow SK' \cup \{(\gamma, S')\}$ 
end foreach
return  $SK'$ 

```

Clearly, this algorithm is not satisfactory since the number of compatible bases may be exponential.

Our aim is then to equivalently compute SK' without exploiting the set of all compatible possibilistic knowledge bases.

It is easy to show that $\forall \omega \in \Omega, \pi_{K'}(\omega) = \pi_K(\omega | \phi)$. Now, in the set-valued setting, conditioning SK comes down first to apply standard conditioning on each compatible base then gathering all certainty degrees. Clearly, SK' is obtained from SK by ignoring some weight. The conditions under which a weight should be ignored is given by the following proposition:

Proposition 4. *Let SK be a set-valued knowledge base, ϕ be a propositional formula. Let $(\gamma, S) \in SK$ and $a \in S$. Let S' be the new set associated with γ in SK' . Then:*

$$a \in S' \text{ iff } \phi \wedge \{\varphi : (\varphi, S) \in SK, \underline{S} \geq a\} \wedge \gamma \text{ is consistent.}$$

Proof. The proof is as follows. Assume that $a \in S'$. This means that there exists a compatible base K such that $(\gamma, a) \in K'$. Since $\{\varphi : (\varphi, \alpha) \in K'\}$ is consistent, and $(\gamma, a) \in K'$ and $(\phi, 1) \in K'$ then trivially $\phi \wedge \gamma \wedge \{\varphi : (\varphi, b) \in K'\}$ is consistent. Hence, $\phi \wedge \gamma \wedge \{\varphi : (\varphi, b) \in K', b \geq a\}$ is consistent and $\phi \wedge \gamma \wedge \{\varphi : (\varphi, S) \in SK, \underline{S} \geq a\}$ is consistent.

Now, assume that $\phi \wedge \gamma \wedge \{\varphi : (\varphi, S) \in SK, \underline{S} \geq a\}$ is consistent. Let K be a compatible base, where each (φ, S) such that $\varphi \neq \gamma$ is replaced by (φ, \underline{S}) and (γ, S) is replaced by (γ, a) . Clearly, K is a compatible. Besides, $(\gamma, a) \in K'$ since $K_{\geq a} \wedge \phi$ is consistent. Hence, $a \in S'$. \square

Based on the above propositions, we propose an algorithm (Algorithm 2) to compute the result of conditioning SK with ϕ . It consists in browsing all the degrees of SK and checking whether each degree should be replaced by 0 or not.

In Algorithm 2, the costly task is checking consistency of the statement marked by (#). Hence, the complexity of computing SK' is $O(|SK| * n * SAT)$ where n is the number of different certainty levels in SK (namely, $n = |\bigcup\{S : (\varphi, S) \in SK\}|$). This is stated in the following proposition.

Proposition 5. *Let SK be a set-valued possibilistic knowledge base and ϕ be the new evidence. Let SK' be a set-valued possibilistic knowledge base computed using Algorithm 2. Then computing SK_ϕ is in $O(|SK| * n * SAT)$ where SAT is a satisfiability test of a set propositional clauses and n is the number of different weights in SK .*

Example 8. *Let us illustrate Algorithm 2. To do so, we continue Example 4 where $SK = \{(\neg c \vee r, \{.4, .7, .8\}), (r, \{.6\})\}$ and with the new information $\phi = \neg r$. For each pair (φ, S) ,*

Algorithm 2 Syntactic set-valued conditioning

Input: SK : a set-valued knowledge base
 ϕ : a propositional formula
Output: SK' : the result of conditioning SK with ϕ

```

 $SK' \leftarrow \{(\phi, 1)\}$ 
foreach  $(\gamma, S) \in SK$  do
   $S' \leftarrow \emptyset$ 
  foreach  $a \in S$  do
    if (#)  $\phi \wedge \gamma \wedge \{\varphi : (\varphi, S) \in SK, \underline{S} \geq a\}$  is consistent then
       $S' \leftarrow S' \cup \{a\}$ 
    else
       $S' \leftarrow S' \cup \{0\}$ 
    end if
     $SK' \leftarrow SK' \cup \{(\gamma, S')\}$ 
  end foreach
end foreach
return  $SK'$ 

```

- First let us take $(\neg c \vee r, \{.4, .7, .8\})$ then:
 - For $a = .4$, $\{r, \neg c \vee r\} \wedge \{\neg r\} \wedge \{\neg c \vee r\}$ is not consistent then, $0 \in S'$;
 - For $a = .7$, $\emptyset \wedge \{\neg r\} \wedge \{\neg c \vee r\}$ is consistent then, $.7 \in S'$;
 - We use the same reasoning for $a = .8$, then, $.8 \in S'$.
- Now for the second pair $(r, \{.6\})$ we have:
 - For $a = .6$, $\{r\} \wedge \{\neg r\} \wedge \{r\}$ is not consistent so $0 \in S'$;

The new base is $SK' = \{(\neg r, \{1\}), (\neg c \vee r, \{0, .7, .8\}), (r, \{0\})\}$. Thanks to Lemma 2, we can exclude the pair $(r, \{0\})$, this is our new base: $SK' = \{(\neg r, \{1\}), (\neg c \vee r, \{0, .7, .8\})\}$. The corresponding set-valued possibility distribution according Definition 8 is given in Table 5.

Table 5. Set-valued distribution corresponding to set-valued knowledge base SK' .

	$S\pi_{SK'}$
cr	$\{0\}$
$\neg cr$	$\{0\}$
$c\neg r$	$\{.2, .3, 1\}$
$\neg c\neg r$	$\{1\}$

6 RELATED WORKS AND DISCUSSIONS

This paper dealt with representing and reasoning with qualitative information in a possibilistic setting and it provided three main contributions:

- The first one is a new extension of possibilistic logic called set-valued possibilistic logic particularly suited for reasoning with qualitative and multiple source information. We provided a natural semantics in terms of compatible possibilistic bases and compatible possibility distributions.
- The second main contribution deals with a generalization of the well-known min-based or qualitative conditioning to the new set-valued setting. The paper proposes three natural postulates ensuring that any set-valued conditioning satisfying these three postulates is necessarily based on the set of compatible standard possibility distributions.

- The third main contribution concerns the syntactic characterization of set-valued conditioning. Efficient procedures are proposed to compute the exact set-valued possibility distributions and their syntactic counterparts. Interestingly enough, the proposed setting generalizes standard possibilistic and conditioning does not require extra computational cost with respect to the standard single valued possibilistic setting. We provide an algorithm which does not generate explicitly the set of all compatible possibilistic knowledge bases.

Many extensions have been proposed to generalize possibilistic logic. The closest one to set-valued possibilistic logic, proposed in this paper, is interval-based possibilistic logic [4, 11, 5]. The two settings view a knowledge base (resp. possibility distribution) as a family of compatible bases (resp. distributions). Of course, intervals are particular sets. However, in [5] conditioning operator deals only with quantitative interpretation of possibility theory [5] while set-valued possibilistic logic deals with qualitative possibility theory. Besides, the rational postulates given in [5] does not characterise the uniqueness of conditioning operator while in this paper, this three postulates **S1**, **S2**, and **S3** guarantee the uniqueness of the conditioning operation.

Among the other extensions, symbolic possibilistic logic [6, 7] deals with a special type of uncertainty where the available uncertain information is in the form of partial knowledge on the relative certainty degrees (symbolic weights) associated with formulas. In [2], a multiple agent extension of possibilistic logic is proposed. This extension associates sets of agents to sets of possibilistic logic formulas and aims to reason on the individual and mutual beliefs of the agents. Note that no form of conditioning the whole knowledge is proposed for this setting.

Note that the idea of compatible-based conditioning in the interval-based possibilistic setting is somehow similar to conditioning in credal sets [1, 26] and credal networks [12] where the concept of convex set refers to the set of compatible probability distributions composing the credal set. Regarding the computational cost, conditioning in credal sets is done on the set of extreme points (edges of the polytope representing the credal set) but their number can reach $N!$ where N is the number of interpretations [28]. In this paper, our set-valued conditioning operator has a complexity close to the one of standard possibilistic knowledge base.

Clearly, many of the qualitative extensions of possibilistic logic mentioned in this section could benefit from our conditioning operators as far as they can be encoded as set-valued possibilistic bases. This will be our main track for future works.

REFERENCES

- [1] T Augustin, G Walter, and F P. A. Coolen, *Statistical inference*, John Wiley & Sons, Ltd, 2014.
- [2] A Belhadji, D Dubois, F Khellaf-Haned, and H Prade, 'Multiple agent possibilistic logic', *Journal of Applied Non-Classical Logics*, **23**(4), 299–320, (2013).
- [3] S Benferhat, C Da Costa Pereira, and A Tettamanzi, 'Syntactic Computation of Hybrid Possibilistic Conditioning under Uncertain Inputs', in *IJCAI*, p. 6822, Beijing, China, (August 2013). AAAI.
- [4] S Benferhat, J Hué, S Lagrue, and J Rossit, 'Interval-based possibilistic logic', in *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pp. 750–755, (2011).
- [5] S Benferhat, A Levray, K Tabia, and V Kreinovich, 'Compatible-based conditioning in interval-based possibilistic logic', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pp. 2777–2783, (2015).
- [6] S Benferhat and H Prade, 'Encoding formulas with partially constrained weights in a possibilistic-like many-sorted propositional logic', in *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*, pp. 1281–1286, (2005).
- [7] C Cayrol, D Dubois, and F Touazi, 'Symbolic Possibilistic Logic: Completeness and Inference Methods (regular paper)', in *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU), Compigne, 15/07/2015-17/07/2015*, eds., T Denoeux and S Destercke, number 9161 in LNAI, pp. 485–495. Springer, (juillet 2015).
- [8] S De Clercq, S Schockaert, A Nowé, and M De Cock, 'Multilateral negotiation in boolean games with incomplete information using generalized possibilistic logic', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pp. 2890–2896, (2015).
- [9] G Coletti and D Petturiti, 'Finitely maxitive conditional possibilities, bayesian-like inference, disintegrability and conglomerability', *Fuzzy Sets and Systems*, **284**, 31–55, (2016).
- [10] G Coletti and D Petturiti, 'Finitely maxitive t-conditional possibility theory: Coherence and extension', *Int. J. Approx. Reasoning*, **71**, 64–88, (2016).
- [11] G Coletti, D Petturiti, and B Vantaggi, 'Coherent t-conditional possibility envelopes and nonmonotonic reasoning', in *Information Processing and Management of Uncertainty in Knowledge-Based Systems - 15th International Conference, IPMU 2014, Montpellier, France, July 15-19, 2014, Proceedings, Part III*, pp. 446–455, (2014).
- [12] F. G. Cozman, 'Credal networks', *Artificial Intelligence*, **120**(2), 199 – 233, (2000).
- [13] P Dellunde, L Godo, and E Marchioni, 'Extending possibilistic logic over gödel logic', *Int. J. Approx. Reasoning*, **52**(1), 63–75, (2011).
- [14] D Dubois, 'On various ways of tackling incomplete information in statistics', *Int. J. Approx. Reasoning*, **55**(7), 1570–1574, (October 2014).
- [15] D Dubois, J Lang, and H Prade, 'Timed possibilistic logic', *Fundam. Inform.*, **15**(3-4), 211–234, (1991).
- [16] D Dubois and H Prade, 'The logical view of conditioning and its application to possibility and evidence theories', *Int. J. Approx. Reasoning*, **4**(1), 23–46, (1990).
- [17] D. Dubois and H. Prade, 'Bayesian conditioning in possibility theory', *Fuzzy Sets and Systems*, **92**(2), 223 – 240, (1997). Fuzzy Measures and Integrals.
- [18] D. Dubois and H. Prade, 'Possibility theory and its applications: a retrospective and prospective view', in *Decision Theory and Multi-Agent Planning*, volume 482, 89–109, Springer Vienna, (2006).
- [19] D Dubois and H Prade, 'Generalized possibilistic logic', in *Scalable Uncertainty Management*, eds., S Benferhat and J Grant, volume 6929 of *Lecture Notes in Computer Science*, 428–432, Springer Berlin Heidelberg, (2011).
- [20] D Dubois and H Prade, 'Possibility theory and its applications: Where do we stand?', in *Springer Handbook of Computational Intelligence*, eds., Janusz Kacprzyk and Witold Pedrycz, 31–60, Springer Berlin Heidelberg, (2015).
- [21] D Dubois, H Prade, and S Schockaert, 'Stable models in generalized possibilistic logic', in *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*, (2012).
- [22] T-F Fan and C-J Liao, 'A logic for reasoning about justified uncertain beliefs', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pp. 2948–2954, (2015).
- [23] P. Fonck, 'A comparative study of possibilistic conditional independence and lack of interaction', *International Journal of Approximate Reasoning*, **16**(2), 149–171, (1997).
- [24] E. Hisdal, 'Conditional possibilities independence and non interaction', *Fuzzy Sets and Systems*, 283–297, (1978).
- [25] J Lang, 'Possibilistic logic: complexity and algorithms', in *Algorithms for Uncertainty and Defeasible Reasoning*, eds., Jrg Kohlas and Serafin Moral, volume 5, 179–220, Kluwer Academic Publishers, (2001).
- [26] I Levi, *The Enterprise of Knowledge: An Essay on Knowledge, Credal Probability, and Chance*, volume 92, The Mit Press, 1980.

- [27] J.F. Huete L.M. De Campos and S. Moral, 'Possibilistic independence', *Proceedings of EUFIT 95*, **1**, 69–73, (1995).
- [28] A Wallner, 'Extreme points of coherent probabilities in finite spaces', *International Journal of Approximate Reasoning*, **44**(3), 339 – 357, (2007). Reasoning with Imprecise Probabilities.
- [29] L. A. Zadeh, 'Fuzzy sets as a basis for a theory of possibility', *Fuzzy Sets Syst.*, **100**, 9–34, (April 1999).

An Improved CNF Encoding Scheme for Probabilistic Inference

Anicet Bart¹ and Frédéric Koriche and Jean-Marie Lagniez and Pierre Marquis²

Abstract. We present and evaluate a new CNF encoding scheme for reducing probabilistic inference from a graphical model to weighted model counting. This new encoding scheme elaborates on the CNF encoding scheme **ENC4** introduced by Chavira and Darwiche, and improves it by taking advantage of log encodings of the elementary variable/value assignments and of the implicit encoding of the most frequent probability value per conditional probability table. From the theory side, we show that our encoding scheme is faithful, and that for each input network, the CNF formula it leads to contains less variables and less clauses than the CNF formula obtained using **ENC4**. From the practical side, we show that the *C2D* compiler empowered by our encoding scheme performs in many cases significantly better than when **ENC4** is used, or when the state-of-the-art *ACE* compiler is considered instead.

1 INTRODUCTION

A number of approaches have been developed during the past fifteen years for improving probabilistic inference, by taking advantage of the local structure (contextual independence and determinism) which may occur in the input graphical model (a weighted constraint network, representing typically a Bayesian network or a Markov network) [4, 34, 22, 3, 28, 18, 9, 16, 23, 36]. Among them are approaches which consist in associating with the input graphical model a weighted propositional formula via a polynomial-time translation [14, 33, 7, 37, 10, 11]. Once the translation has been applied, the problem of computing the probability (or more generally, the weight) of a given piece of evidence (assignments of values to some variables) mainly amounts to solving an instance of the (weighted) model counting problem.

While this problem is still $\#P$ -complete, it has received much attention in the past few years, both in theory and in practice; thus, many algorithms have been designed for solving the model counting problem $\#SAT$, either exactly or approximately (see e.g., [2, 19, 20, 31, 5]); search-based model counters (like *Cachet* [32] and *sharpSAT* [35]) and preprocessing techniques for $\#SAT$ [21] have been developed and evaluated. Propositional languages supporting the (weighted) model counting query in polynomial time have been defined and investigated, paving the way to compilation-based model counters (i.e., when the propositional encoding of the model is first turned into a compiled representation). The most prominent ones are the language *Decision-DNNF* of decision-based decomposable negation normal form formulas [12] and the language *SDD* consisting

of sentential decision diagrams – a subset of *d-DNNF* – [17]. Compilers targeting those languages have been developed (many of them are available on line); let us mention the top-down compilers *C2D* and *Dsharp* targeting the *Decision-DNNF* language [12, 15, 25], and the top-down compiler and the bottom-up compiler targeting the *SDD* language [27, 17].

In the following, we present and evaluate a new CNF encoding scheme for weighted constraint networks. This new encoding scheme elaborates on the CNF encoding scheme **ENC4** introduced in [10], and improves it by taking advantage of two supplementary “ideas”: the log encodings of the elementary variable/value assignments and of the implicit encoding of the most frequent probability value per table. While log encodings of variables is a simple idea which has been considered before for credal networks (and defined as “binarization”) [1], as far as we know, it had never been tested before for encoding weighted constraint networks into CNF. Furthermore, using an implicit encoding of the most frequent probability value per table seems brand new in this context.

Interestingly, unlike the formulae obtained using **ENC4**, the CNF formulae generated by our encoding scheme can be compiled into *Decision-DNNF* representations which do not need to be minimized (thanks to a specific handling of the weights given to the negative parameter literals). As such, they can also be exploited directly by any weighted model counter. From the theory side, we show that our encoding scheme is faithful, and that for each weighted constraint network, the CNF formula it leads to contains less variables and less clauses than the CNF formula obtained using **ENC4**. From the practical side, we performed a large-scale evaluation by compiling 1452 weighted constraint networks from 6 data sets. This evaluation shows our encoding scheme valuable in practice. More in detail, we have compared the compilation times and the sizes of the compiled representations produced by *C2D* (reasoning.cs.ucla.edu/c2d/) when using our encoding scheme, with the corresponding measures when **ENC4** is considered instead. Our encoding scheme appeared as a better performer than **ENC4** since it led most of the time to improved compilation times and improved compilation sizes. We have also done a differential evaluation which has revealed that each of the two “ideas” considered in our encoding is computationally fruitful. We finally compared the performance of *C2D* empowered by our encoding scheme with those of *ACE*, a state-of-the-art compiler for Bayesian networks based on **ENC4**, see <http://reasoning.cs.ucla.edu/ace>. Again, our empirical investigation also showed that *C2D* equipped with our encoding scheme challenges *ACE* in many cases. More precisely, it was able to compile more instances given the time and memory resources allocated in our experiments, and it led often to compiled representations significantly smaller than the ones computed using *ACE*.

¹ LINA-CNRS-INRIA and Ecole des Mines de Nantes, F-44000 Nantes, France, email: anicet.bart@univ-nantes.fr

² CRIL, Univ. Artois and CNRS, F-62300 Lens, France, email: {koriche, lagniez, marquis}@cril.univ-artois.fr

2 FORMAL PRELIMINARIES

A (finite-domain) *weighted constraint network* (alias WCN) is a triple $WCN = (\mathcal{X}, \mathcal{D}, \mathcal{R})$ where $\mathcal{X} = \{X_1, \dots, X_n\}$ is a finite set of variables; each variable X from \mathcal{X} is associated with a finite set, its domain D_X , and \mathcal{D} is the set of all domains of the variables from \mathcal{X} ; $\mathcal{R} = \{R_1, \dots, R_m\}$ is a finite set of (possibly partial) functions over the reals; with each R in \mathcal{R} is associated a subset $scope(R)$ of \mathcal{X} , called the scope of R and gathering the variables involved in R ; each R is a mapping from its domain $Dom(R)$, a subset of the Cartesian product D_R of the domains of the variables of $scope(R)$, to \mathbb{R} ; the cardinality of $scope(R)$ is the arity of R . In the following, each function R is supposed to be represented in extension (i.e., as a table associating weights with assignments).

An *assignment* \mathbf{a} of WCN over a subset \mathcal{S} of \mathcal{X} is a set $\mathbf{a} = \{(X, d) \mid X \in \mathcal{S}, d \in D_X\}$ of elementary assignments (X, d) , where for each $X \in \mathcal{S}$ there exists a unique pair of the form (X, d) in \mathbf{a} . If \mathbf{a} is an assignment of WCN over \mathcal{S} and $\mathcal{T} \subseteq \mathcal{S}$, then the restriction $\mathbf{a}[\mathcal{T}]$ of \mathbf{a} over \mathcal{T} is given by $\mathbf{a}[\mathcal{T}] = \{(X, d) \in \mathbf{a} \mid X \in \mathcal{T}\}$. Given two subsets \mathcal{S} and \mathcal{T} of \mathcal{X} such that $\mathcal{T} \subseteq \mathcal{S}$, an assignment \mathbf{a}_1 of WCN over \mathcal{S} is said to *extend* an assignment \mathbf{a}_2 of WCN over \mathcal{T} when $\mathbf{a}_1[\mathcal{T}] = \mathbf{a}_2$. A *full assignment* of WCN is an assignment of WCN over \mathcal{X} .

A full assignment \mathbf{s} of WCN is a *solution* of WCN if and only if for every $R \in \mathcal{R}$, we have $\mathbf{s}[scope(R)] \in Dom(R)$. The *weight* of a full assignment \mathbf{s} of WCN is $w_{WCN}(\mathbf{s}) = 0$ when \mathbf{s} is not a solution of WCN ; otherwise, $w_{WCN}(\mathbf{s}) = \prod_{R \in \mathcal{R}} R(\mathbf{s}[scope(R)])$. Finally, the *weight* $w_{WCN}(\mathbf{a})$ of an assignment \mathbf{a} of WCN over \mathcal{S} is the sum over all full assignments \mathbf{s} extending \mathbf{a} of the values $w_{WCN}(\mathbf{s})$.

Example 1 Let us consider as a running example the following "toy" $WCN = (\mathcal{X} = \{X_1, X_2\}, \mathcal{D} = \{D_{X_1}, D_{X_2}\}, \mathcal{R} = \{R\})$, where $D_{X_1} = \{0, 1, 2\}$, $D_{X_2} = \{0, 1\}$, and R such that $scope(R) = \{X_1, X_2\}$ is given by Table 1.

X_1	X_2	R
0	0	0
0	1	$8/30$
1	0	$1/10$
1	1	$1/10$
2	0	$8/30$
2	1	$8/30$

Table 1: A tabular representation of R .

$\mathbf{a} = \{(X_2, 1)\}$ is an assignment of WCN over $\mathcal{S} = \{X_2\}$. We have $w_{WCN}(\mathbf{a}) = 8/30 + 1/10 + 8/30 = 19/30$.

3 ON CNF ENCODING SCHEMES

Our objective is to be able to compute the *weight* of any assignment \mathbf{a} of a given WCN . Typically, the WCN under consideration will be derived without any heavy computational effort (i.e., in linear time) from a given random Markov field or a Bayesian network, and the assignment \mathbf{a} under consideration will represent some available piece of evidence. In such a case, $w(\mathbf{a})$ simply is the probability of this piece of evidence.

In order to achieve this goal, an approach consists in translating first the input $WCN = (\mathcal{X}, \mathcal{D}, \mathcal{R})$ into a *weighted propositional formula* $WPROF = (\Sigma, w, w_0)$. In such a triple, Σ is a propositional representation built up from a finite set of propositional variables PS , w is a weight distribution over the literals over PS , i.e., a mapping from $L_{PS} = \{x, \neg x \mid x \in PS\}$ to \mathbb{R} , and

w_0 is a real number (a scaling factor). The *weight* of an interpretation ω over PS given $WPROF$ is defined as $w_{WPROF}(\omega) = w_0 \times \prod_{x \in L_{PS} \mid \omega(x)=1} w(x) \times \prod_{\neg x \in L_{PS} \mid \omega(x)=0} w(\neg x)$ if ω is a model of Σ , and $w_{WPROF}(\omega) = 0$ in the remaining case. Furthermore, the *weight* of any consistent term γ over PS given $WPROF$ is given by $w_{WPROF}(\gamma) = \sum_{\omega \models \gamma} w_{WPROF}(\omega)$. Given a CNF formula Σ , we denote by $\#var(\Sigma)$ the number of variables occurring in Σ , and by $\#cl(\Sigma)$, the number of clauses in Σ . Finally, a canonical term over a subset V of PS is a consistent term where each variable of V occurs.

Computing $w(\gamma)$ from a consistent term γ and a $WPROF = (\Sigma, w, w_0)$ is a computationally hard problem in general (it is $\#P$ -complete). Weighted model counters (such as *Cachet* [32]) can be used in order to perform such a computation when Σ is a CNF formula. Reductions from the weighted model counting problem to the (unweighted) model counting problem, as the one reported in [6], can also be exploited, rendering possible the use of (unweighted) model counter, like *sharpSAT* [35]. Interestingly, when Σ has been compiled first into a Decision-DNNF representation (and more generally into a d -DNNF representation³), the computation of $w(\gamma)$ can be done in time linear in the size of the input, i.e., the size of γ , plus the size of the explicit representation of the weight distribution w over L_{PS} , the size of the representation of w_0 , and the size of the d -DNNF representation of Σ . Stated otherwise, the problem of computing $w(\gamma)$ from a consistent term γ and a $WPROF = (\Sigma, w, w_0)$ where Σ is a d -DNNF representation can be solved efficiently.

Whatever the targeted model counter (direct or compilation-based), the approach requires a notion of translation function (the formal counterpart of an encoding scheme):

Definition 1 (translation function) A mapping τ associating any $WCN = (\mathcal{X}, \mathcal{D}, \mathcal{R})$ with a weighted propositional formula $\tau(WCN) = (\Sigma, w, w_0)$ and any assignment \mathbf{a} of WCN over a subset \mathcal{S} of \mathcal{X} with a term $\tau(\mathbf{a})$ over the set of propositional variables PS on which Σ is built, is a translation function.

Valuable translation functions are those for which the encoding scheme is correct. We say that they are faithful:

Definition 2 (faithful translation) A translation function τ is faithful when it is such that for any $WCN = (\mathcal{X}, \mathcal{D}, \mathcal{R})$ and any assignment \mathbf{a} of WCN over a subset \mathcal{S} of \mathcal{X} , $w_{WCN}(\mathbf{a}) = w_{\tau(WCN)}(\tau(\mathbf{a}))$.

Some faithful translation functions have already been identified in the literature, see [13, 33, 8, 10, 11]. Typically, the set PS of propositional variables used in the translation is partitioned into two subsets: a set of indicator variables λ_i used to encode the assignments, and a set of parameter variables θ_j used to encode the weights. Formally let us denote by Λ_X the set of indicator variables used to encode assignments of variable $X \in \mathcal{X}$ and Θ_R be the set of parameter variables introduced in the encoding of $R \in \mathcal{R}$. Every literal l over all those variables has weight 1 (i.e., $w_1(l) = w_4(l) = 1$), except for the (positive) literals θ_j . Translations functions are typically *modular ones*, where "modular" means that the representation Σ to be generated is the conjunction of the representations $\tau(X)$ corresponding to the encoding of the domain D_X of each $X \in \mathcal{X}$, with the representations $\tau(R)$ corresponding to each mapping R in \mathcal{R} :

$$\Sigma = \bigwedge_{X \in \mathcal{X}} \tau(X) \wedge \bigwedge_{R \in \mathcal{R}} \tau(R).$$

³ But existing d -DNNF compilers actually target the Decision-DNNF language [26].

As a matter of example, let us consider the translation functions τ_1 and τ_4 associated respectively with the encoding schemes **ENC1** [13] and **ENC4** reported in [8]. In **ENC1** and **ENC4**, direct encoding is used for the representation of elementary assignments (X, d) . This means that every (X, d) is associated by τ_1 (and similarly by τ_4) in a bijective way with an indicator variable $\tau_1((X, d)) = \tau_4((X, d))$, and every assignment \mathbf{a} is associated with the term $\tau_1(\mathbf{a}) = \tau_4(\mathbf{a})$ which is the conjunction of the indicator variables $\tau_1((X, d))$ for each $(X, d) \in \mathbf{a}$. The encoding $\tau_1(X) = \tau_4(X)$ consists of the following CNF formula:

$$\left(\bigvee_{d \in D_X} \tau_1((X, d)) \right) \wedge \left(\bigwedge_{d_1, d_2 \in D_X \mid d_1 \neq d_2} \neg \tau_1((X, d_1)) \vee \neg \tau_1((X, d_2)) \right).$$

Finally, in τ_1 and τ_4 , the scaling factor $(w_1)_0 = (w_4)_0$ is 1.

Contrastingly, **ENC1** and **ENC4** differ in the way mappings R are encoded. In **ENC1**, each $\tau_1(R)$ is a CNF formula, consisting for each $\mathbf{a} \in \text{Dom}(R)$ of the following CNF formulae: $(\bigvee_{(X,d) \in \mathbf{a}} \neg \tau_1((X, d)) \vee \theta_{\mathbf{a}}) \wedge \bigwedge_{(X,d) \in \mathbf{a}} (\tau_1((X, d)) \vee \neg \theta_{\mathbf{a}})$. This formula contains $c \times (a + 1)$ clauses where c is the cardinality of $\text{Dom}(R)$ and a is the arity of R . Here, $\theta_{\mathbf{a}}$ is a parameter variable which is specific to \mathbf{a} . For each \mathbf{a} , the corresponding CNF formula actually states an equivalence between $\tau_1(\mathbf{a})$ and $\theta_{\mathbf{a}}$. Finally, $w_1(\theta_{\mathbf{a}}) = R(\mathbf{a})$.

In **ENC4**, for each $R \in \mathcal{R}$, one parameter variable θ_j per non-null weight in R is introduced, only. Thus, no parameter variable is introduced for the $\mathbf{a} \in \text{Dom}(R)$ such that $R(\mathbf{a}) = 0$. Furthermore, all the assignments $\mathbf{a} \in \text{Dom}(R)$ which are associated with the same value $R(\mathbf{a})$ are associated with the same parameter variable θ_j which is such that $w_4(\theta_j) = R(\mathbf{a})$. Each $\tau_4(R)$ is a CNF formula, obtained first by computing a compressed representation of R in a way similar to the way a simplification of a Boolean function f is computed using Quine/McCluskey algorithm, i.e., as a minimal number of prime implicants of f the disjunction of which being equivalent to f (see [29, 30, 24] and [10] for details). Once R has been compressed, $\tau_4(R)$ is computed as the conjunction for each $\mathbf{a} \in \text{Dom}(R)$ of the following clauses:

$$\begin{aligned} & \bigvee_{(X,d) \in \mathbf{a}} \neg \tau_4((X, d)) \text{ if } R(\mathbf{a}) = 0, \\ & \bigvee_{(X,d) \in \mathbf{a}} \neg \tau_4((X, d)) \vee \theta_j \text{ if } R(\mathbf{a}) \neq 0. \end{aligned}$$

Note that τ_4 by itself is not a faithful translation: the generated formula Σ_4 (the conjunction of all $\tau_4(X)$ for $X \in \mathcal{X}$ and of all $\tau_4(R)$ for $R \in \mathcal{R}$) must be *minimized* first w.r.t. its parameter variables in order to get a faithful translation. Such a "cardinality minimization", noted $\text{min}_{\theta}(\Sigma_4)$, leads to a strengthening of Σ_4 , obtained by removing every model of it assigning to true more than one parameter variable associated with a given R . Now, for each variable $X \in \mathcal{X}$, given the CNF formula $\tau_4(X)$, exactly one of the indicator variables $\tau_4((X, d))$ for $d \in D_X$ can be set to true in a model of Σ_4 . Accordingly, the "global cardinality minimization" $\text{min}(\Sigma_4)$ (i.e., when "cardinality minimization" is w.r.t. *all* the variables) can be done instead, since we have $\text{min}(\Sigma_4) = \text{min}_{\theta}(\Sigma_4)$. The main point is that the mapping τ_4^{min} associating $\mathcal{WCN} = (\mathcal{X}, \mathcal{D}, \mathcal{R})$ with the WPROP $(\text{min}(\Sigma_4), w_4, (w_4)_0)$ is faithful. Interestingly, when Σ_4 has been turned first into an equivalent d-DNNF representation, such a "global cardinality minimization" process leading to a minimized d-DNNF representation $\text{min}(\Sigma_4)$ can be achieved in linear time [12].

Example 2 (Example 1 continued) *As a matter of illustration, let us present the encodings obtained by applying τ_1 and τ_4 to our running example. τ_1 and τ_4 are based on the same set consisting of 5*

indicator variables, λ_i^j , where λ_i^j corresponds to the elementary assignment (X_i, j) , and on the same set of indicator clauses:

$$\begin{array}{lll} \lambda_1^0 \vee \lambda_1^1 \vee \lambda_1^2, & \neg \lambda_1^0 \vee \neg \lambda_1^2, & \lambda_2^0 \vee \lambda_2^1, \\ \neg \lambda_1^0 \vee \neg \lambda_1^1, & \neg \lambda_1^1 \vee \neg \lambda_1^2, & \neg \lambda_2^0 \vee \neg \lambda_2^1. \end{array}$$

τ_1 and τ_4 differ as to their parameter variables, and as to their parameter clauses. For τ_1 , one parameter variable per element of $\text{Dom}(R)$ (hence per line in Table 1) is introduced: each θ_i corresponds to line i , thus 6 variables are introduced. For τ_4 , one parameter variable per non-null value taken by R is considered, hence two parameter variables θ_1 (corresponding to $1/10$) and θ_2 (corresponding to $8/30$) are introduced. On this ground, $\tau_1(R)$ consists of the following parameter clauses:

$$\begin{array}{lll} \neg \lambda_1^0 \vee \neg \lambda_2^0 \vee \theta_1, & \neg \lambda_1^1 \vee \neg \lambda_2^0 \vee \theta_3, & \neg \lambda_1^2 \vee \neg \lambda_2^0 \vee \theta_5, \\ \lambda_1^0 \vee \neg \theta_1, & \lambda_1^1 \vee \neg \theta_3, & \lambda_2^1 \vee \neg \theta_5, \\ \lambda_2^0 \vee \neg \theta_1, & \lambda_2^0 \vee \neg \theta_3, & \lambda_2^0 \vee \neg \theta_5, \\ \neg \lambda_1^0 \vee \neg \lambda_2^1 \vee \theta_2, & \neg \lambda_1^1 \vee \neg \lambda_2^1 \vee \theta_4, & \neg \lambda_2^1 \vee \neg \lambda_2^1 \vee \theta_6, \\ \lambda_1^0 \vee \neg \theta_2, & \lambda_1^1 \vee \neg \theta_4, & \lambda_2^1 \vee \neg \theta_6, \\ \lambda_2^1 \vee \neg \theta_2, & \lambda_2^1 \vee \neg \theta_4, & \lambda_2^1 \vee \neg \theta_6, \end{array}$$

with $w_1(\theta_1) = 0$, $w_1(\theta_2) = w_1(\theta_5) = w_1(\theta_6) = 8/30$, $w_1(\theta_3) = w_1(\theta_4) = 1/10$, and every other literal has weight 1. Σ_1 contains 24 clauses, over 11 variables.

Contrastingly, with τ_4 , R is first compressed into

X_1	X_2	R
0	0	0
0	1	$8/30$
1		$1/10$
2		$8/30$

As a consequence, $\tau_4(R)$ consists of the following parameter clauses:

$$\begin{array}{ll} \neg \lambda_1^0 \vee \neg \lambda_2^0, & \neg \lambda_1^1 \vee \theta_1, \\ \neg \lambda_1^0 \vee \neg \lambda_2^1 \vee \theta_2, & \neg \lambda_1^2 \vee \theta_2, \end{array}$$

with $w_4(\theta_1) = 1/10$, $w_4(\theta_2) = 8/30$, and every other literal has weight 1. Σ_4 contains 10 clauses, over 7 variables.

4 A NEW, IMPROVED CNF ENCODING SCHEME

We present a new translation function $\tau_{4\text{linp}}$, which is modular as τ_1 and τ_4 . $\tau_{4\text{linp}}$ elaborates on τ_4 in two directions: the way elementary assignments are encoded, and the implicit handling of one parameter variable per mapping R .

Thus, within the translation function $\tau_{4\text{linp}}$, *log encoding* (aka bit-wise encoding) is used for the representation of elementary assignments (X, d) . The corresponding $\tau_{4\text{linp}}(X)$ CNF formula aims at forbidding the interpretations which do not correspond to any elementary assignment. Thus, there is no such constraint (i.e., it is equivalent to \top) when the cardinality of the domain of X is a power of 2.

As to the parameter variables and the parameter clauses, our translation function $\tau_{4\text{linp}}$ is reminiscent to τ_4 . However, there are some important differences. First, log encoding is used to define the indicator variables within the parameter clauses. Second, one parameter

variable θ_R per R is kept *implicit* once R has been compressed; it is selected as one of those θ_j such that $w_4(\theta_j) \neq 0$ is one of the most frequent weight in R once compressed. Then we take the scaling factor $(w_{4\text{linp}})_0$ to be equal to the product of all the weights $w_4(\theta_R)$ when R varies in \mathcal{R} , and we replace the weight $w_4(\theta_j)$ of all the remaining parameter variables θ_j associated with R by $w_{4\text{linp}}(\theta_j) = w_4(\theta_j)/w_4(\theta_R)$. The benefits achieved by this scaling come from the fact that there is no need to add any clause into $\tau_{4\text{linp}}(R)$ for the assignments \mathbf{a} such that $R(\mathbf{a}) = w_4(\theta_R)$. More formally, for each $R \in \mathcal{R}$, R is first compressed as in **ENC4**; then we define $\tau_{4\text{linp}}(R)$ as a CNF formula, consisting of the conjunction for each $\mathbf{a} \in \text{Dom}(R)$ such that $R(\mathbf{a}) \neq w_4(\theta_R)$ of the following clauses:

$$\bigvee_{(X,d) \in \mathbf{a}} \neg \tau_{4\text{linp}}((X,d)) \text{ if } R(\mathbf{a}) = 0, \\ \bigvee_{(X,d) \in \mathbf{a}} \neg \tau_{4\text{linp}}((X,d)) \vee \theta_j \text{ if } R(\mathbf{a}) \neq 0.$$

Here $\neg \tau_{4\text{linp}}((X,d))$ is the clause which is obtained as the disjunction of the negations of all literals occurring in $\tau_{4\text{linp}}((X,d))$.

Finally, considering the same weight distribution $w_{4\text{linp}} = w_4$ as the one considered in **ENC4** would not make the translation faithful; in order to ensure it, we now assign a specific weight to the negative parameter literals, so that $w_{4\text{linp}}(\neg \theta_j) = 1 - w_{4\text{linp}}(\theta_j)$ for every parameter variable θ_j considered in the parameter clauses of R , for every $R \in \mathcal{R}$.

As we will show it later, no minimization step is mandatory with $\tau_{4\text{linp}}$; furthermore, this translation is modular (like τ_4 but unlike τ_4^{min}); more importantly, we obtain as a side effect that any weighted model counter can be considered downstream (unlike τ_4 , which requires a minimization step).

Example 3 (Example 1 continued) For the WCN considered in Example 1, one just needs to consider two indicator variables for encoding the elementary assignments associated with X_1 (let us say, λ_1^0 and λ_1^1) and one indicator variable for encoding the elementary assignments associated with X_2 (λ_2). The correspondances between elementary assignments and their representation as terms over the indicator variables are as follows for X_1 :

X_1	λ_1^1	λ_1^0
0	0	0
1	0	1
2	1	0

and for X_2 , λ_2 corresponds to $(X_2, 1)$ (thus, $\neg \lambda_2$ corresponds to $(X_2, 0)$). We have $\tau_{4\text{linp}}(X_1) = \neg \lambda_1^1 \vee \neg \lambda_1^0$ and $\tau(X_2) = \top$. The most frequent value achieved by $R(\mathbf{a})$ is $w_4(\theta_R) = 8/30$. Since $\mathcal{R} = \{R\}$, we get that $(w_{4\text{linp}})_0 = 8/30$. $\tau_{4\text{linp}}(R)$ consists of the two following clauses:

$$\lambda_1^0 \vee \lambda_1^1 \vee \lambda_2, \quad \lambda_1^1 \vee \neg \lambda_1^0 \vee \theta_1.$$

The first clause aims at ensuring that the weight corresponding to the full assignment $\{(X_1, 0), (X_2, 0)\}$ is 0. The purpose of the second clause is to enforce the parameter variable θ_1 to be true when any assignment extending $\{(X_1, 1)\}$ is considered. We finally have $w_{4\text{linp}}(\theta_1) = 3/8$ and $w_{4\text{linp}}(\neg \theta_1) = 5/8$, while every other literal has weight 1. $\Sigma_{4\text{linp}}$ contains only 3 clauses, over 4 variables.

Table 2 makes precise for each interpretation over the variables $\lambda_1^0, \lambda_1^1, \lambda_2$, and θ_1 the corresponding full assignment of WCN over $\{X_1, X_2\}$ (if any) and the associated weight $w_{4\text{linp}}$.

Proposition 1 $\tau_{4\text{linp}}$ is faithful.

λ_1^1	λ_1^0	λ_2	θ_1	X_1	X_2	$w_{4\text{linp}}$
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	1	1/6
0	0	1	1	0	1	1/10
0	1	0	0	1	0	0
0	1	0	1	1	0	1/10
0	1	1	0	1	1	0
0	1	1	1	1	1	1/10
1	0	0	0	2	0	1/6
1	0	0	1	2	0	1/10
1	0	1	0	2	1	1/6
1	0	1	1	2	1	1/10
1	1	0	0	-	-	0
1	1	0	1	-	-	0
1	1	1	0	-	-	0
1	1	1	1	-	-	0

Table 2: The full assignment of WCN over $\{X_1, X_2\}$ and the the weight $w_{4\text{linp}}$ associated with each interpretation over the variables of $\Sigma_{4\text{linp}}$.

Proof: By definition of log encoding, every (partial) assignment \mathbf{a} of WCN over a subset \mathcal{S} of \mathcal{X} is associated with a term $\tau_{4\text{linp}}(\mathbf{a})$ over $\Lambda_{\text{WCN}} = \bigcup_{X \in \mathcal{X}} \Lambda_X$. Furthermore, every full assignment \mathbf{s} of WCN is associated in a bijective way with a term $\tau_{4\text{linp}}(\mathbf{s})$ over Λ_{WCN} which implies $\bigwedge_{X \in \mathcal{X}} \tau_{4\text{linp}}(X)$.

Let us recall that the weight of any full assignment \mathbf{s} is $w_{\text{WCN}}(\mathbf{s}) = 0$ when \mathbf{s} is not a solution of WCN and is $w_{\text{WCN}}(\mathbf{s}) = \prod_{R \in \mathcal{R}} R(\mathbf{s}[\text{scope}(R)])$ otherwise.

Assume first that $w_{\text{WCN}}(\mathbf{s}) = 0$. Then either \mathbf{s} is not a solution of WCN or \mathbf{s} is such that $R(\mathbf{s}[\text{scope}(R)]) = 0$ for at least one $R \in \mathcal{R}$. Hence there exists $R \in \mathcal{R}$ such that either $\mathbf{s}[\text{scope}(R)] \notin \text{Dom}(R)$ or $R(\mathbf{s}[\text{scope}(R)]) = 0$. Subsequently, there exists a clause in $\Sigma_{4\text{linp}}$ such that $\tau_{4\text{linp}}(\mathbf{s})$ falsifies it. This implies that every interpretation over $\Lambda_{\text{WCN}} \cup \Theta_R$ which extends $\tau_{4\text{linp}}(\mathbf{s})$ falsifies $\Sigma_{4\text{linp}}$. Accordingly, $w_{4\text{linp}}(\tau_{4\text{linp}}(\mathbf{s})) = 0$ as expected.

Assume now that \mathbf{s} is such that $w_{\text{WCN}}(\mathbf{s}) \neq 0$. By construction, for every $R \in \mathcal{R}$, the contribution of R to $w_{\text{WCN}}(\mathbf{s})$ is equal to the factor $R(\mathbf{s}[\text{scope}(R)])$. Suppose that k parameter variables $\theta_1, \dots, \theta_k$ have been introduced in $\tau_{4\text{linp}}(R)$. Then there are two cases to be considered: (1) $R(\mathbf{s}[\text{scope}(R)]) = w_4(\theta_R)$ and (2) $R(\mathbf{s}[\text{scope}(R)]) \neq w_4(\theta_R)$.

In case (1), by construction, $\tau_{4\text{linp}}(\mathbf{s})$ satisfies every clause of $\tau_{4\text{linp}}(R)$. Hence each of the 2^k canonical terms extending $\tau_{4\text{linp}}(\mathbf{s})$ over the k parameter variables implies $\tau_{4\text{linp}}(R)$. Therefore, the contribution of R to $w_{4\text{linp}}(\tau_{4\text{linp}}(\mathbf{s}))$ is equal to the sum, for each canonical term, of the products of the parameter literals occurring in it. But this sum is also equal to $\prod_{i=1}^k (w_{4\text{linp}}(\theta_i) + w_{4\text{linp}}(\neg \theta_i)) = 1 = w_4(\theta_R)/w_4(\theta_R) = R(\mathbf{s}[\text{scope}(R)])/w_4(\theta_R)$.

In case (2), by construction, there is a clause $\neg \tau_{4\text{linp}}(\mathbf{s}[\text{scope}(R)]) \vee \theta_j$ in $\tau_{4\text{linp}}(R)$, so that the parameter variable θ_j is set to true in every model of $\Sigma_{4\text{linp}}$ extending $\tau_{4\text{linp}}(\mathbf{s})$. As above, each of the 2^{k-1} canonical terms extending $\tau_{4\text{linp}}(\mathbf{s})$ over the $k-1$ remaining parameter variables (i.e., all of them but θ_j) implies $\tau_{4\text{linp}}(R)$. Therefore, the contribution of R to $w_{4\text{linp}}(\tau_{4\text{linp}}(\mathbf{s}))$ is equal to the sum, for each canonical term, of the products of the parameter literals occurring in it. But this sum is also equal to $R(\mathbf{s}[\text{scope}(R)])/w_4(\theta_R) \times \prod_{i=1, i \neq j}^k (w_{4\text{linp}}(\theta_i) + w_{4\text{linp}}(\neg \theta_i)) = R(\mathbf{s}[\text{scope}(R)])/w_4(\theta_R)$.

Whatever the case (1) or (2), since $(w_{4\text{linp}})_0$ is equal to $\prod_{R \in \mathcal{R}} w_4(\theta_R)$, the factor $w_4(\theta_R)$ of this product balances the denominator of the ratio $w_4(\theta_R)/w_4(\theta_R)$, so that finally, $w_{4\text{linp}}(\tau_{4\text{linp}}(\mathbf{s})) = \prod_{R \in \mathcal{R}} R(\mathbf{s}[\text{scope}(R)]) = w_{\text{WCN}}(\mathbf{s})$ as expected. \blacksquare

Our purpose was also to compare the efficiency of $\tau_{4\text{linp}}$ w.r.t. the

efficiency of τ_4 , where the efficiency is measured as the number of variables and/or as the number of clauses in the corresponding CNF encodings $\Sigma_{4\text{linp}}$ and Σ_4 . We obtained that $\tau_{4\text{linp}}$ is more efficient than τ_4 for both measures:

Proposition 2 *Given $\mathcal{WCN} = (\mathcal{X}, \mathcal{D}, \mathcal{R})$, let $\tau_4(\mathcal{WCN}) = (\Sigma_4, w_4, (w_4)_0)$, and $\tau_{4\text{linp}}(\mathcal{WCN}) = (\Sigma_{4\text{linp}}, w_{4\text{linp}}, (w_{4\text{linp}})_0)$. Then we have:*

- $\#var(\Sigma_{4\text{linp}}) < \#var(\Sigma_4)$,
- $\#cl(\Sigma_{4\text{linp}}) < \#cl(\Sigma_4)$.

Proof:

- $\#var$. When the cardinality of D_X is k , $\tau_4(X)$ uses k indicator variables, while $\tau_{4\text{linp}}(X)$ requires only $\lceil \log_2(k) \rceil$ indicator variables. As to the parameter variables, by construction, $\tau_{4\text{linp}}(R)$ requires one variable less than $\tau_4(R)$.
- $\#cl$. By construction, $\tau_4(X)$ contains $k \times (k-1) + 1$ clauses when k is the cardinality of D_X . Contrastingly, $\tau_{4\text{linp}}(X)$ contains at most $k - 2$ "blocking clauses" (this worst case is obtained when $k = 2^l + 1$ for some l). Hence, the number of clauses in $\tau_{4\text{linp}}(X)$ is strictly lower than the number of clauses in $\tau_4(X)$. Furthermore, by construction, $\tau_{4\text{linp}}(R)$ contains at least one clause less than $\tau_4(R)$ (this worst case situation is obtained when all the values $w(\theta_j) \neq 0$ of the parameter variables θ_j considered by $\tau_4(R)$ are distinct). ■

5 EXPERIMENTS

Our benchmarks consist of 1452 WCNs downloaded from <http://www.hlt.utdallas.edu/~vgogate/uail4-competition/index.html> and <http://reasoning.cs.ucla.edu/ace/>. Those instances correspond to Bayesian networks or random Markov fields in the UAI competition format. They are gathered into 6 data sets, as follows: Diagnose (100), UAI (377), Grids (320), Pedigree (22), Promedas (238), Relational (395).

We translated each input WCN into a WPROP, using both the τ_4 and the $\tau_{4\text{linp}}$ translation function. Downstream to the encoding, we took advantage of the C2D compiler which targets the Decision-DNNF language [12, 15] to compute, for each instance, a minimized Decision-DNNF representation of the CNF formula generated by τ_4 , and a Decision-DNNF representation of the CNF formula generated by $\tau_{4\text{linp}}$. C2D has been run with its default parameters. Note that we could also consider a model counter (like Cachet, which supports weights) downstream to the CNF encoding produced by $\tau_{4\text{linp}}$. For space reasons, we refrain from reporting the corresponding empirical results here because C2D performs often much better than Cachet on CNF instances issued from graphical models (the dtree computed to guide the Decision-DNNF computation achieved by C2D has a major positive impact on the process).

Our experiments have been conducted on a Quad-core Intel XEON X5550 with 32GiB of memory. A time limit of 900s for the compilation phase (including the translation time and the minimization time when τ_4 has been used⁴), and a total amount of 8GiB of memory for storing the resulting Decision-DNNF representations have been

⁴ Minimization can be achieved in linear time on d-DNNF representations [12]. It may have a valuable reduction effect on the size of the compiled form.

considered for each instance. Both the instances used in our experiments, the run-time code of our translator `bn2Cnf` implementing the τ_4 encoding scheme and the $\tau_{4\text{linp}}$ encoding scheme, and some detailed empirical results are available on line from <http://www.cril.fr/KC>.

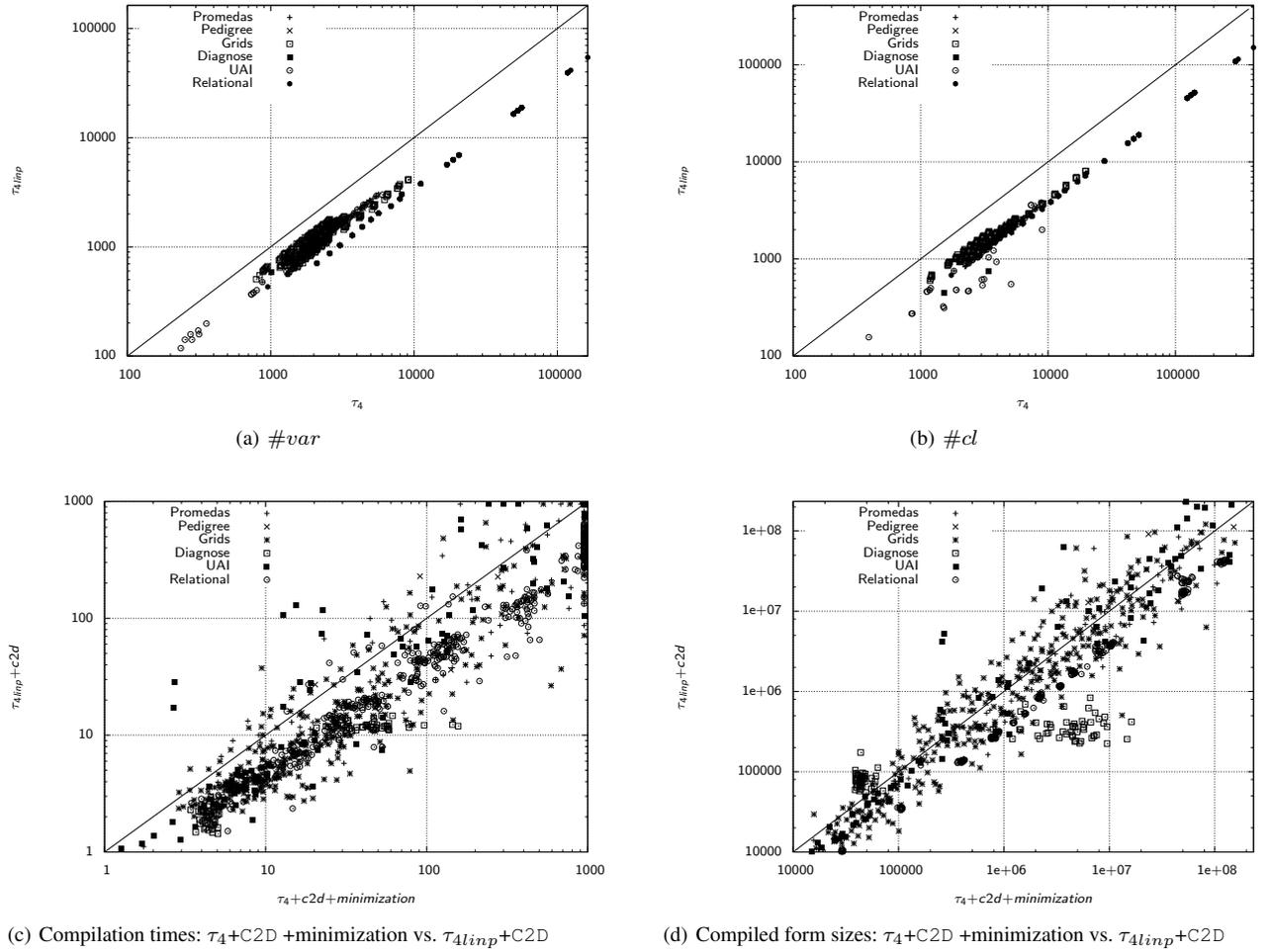
In order to figure out the reductions in the number of variables and in the number of clauses done by $\tau_{4\text{linp}}$ compared to τ_4 , we computed the number of variables $\#var$ and the number of clauses $\#cl$ of $\Sigma_{4\text{linp}}$ and Σ_4 for each instance. Some of our empirical results are depicted using scatter plots with logarithmic scales. Thus, the scatter plots (a) and (b) from Figure 1 report respectively the relative performances of τ_4 and $\tau_{4\text{linp}}$ w.r.t. the measurements $\#var$ and $\#cl$. They cohere with Proposition 2 and show that $\tau_{4\text{linp}}$ leads in practice to CNF encodings which are exponentially smaller w.r.t. both the number of variables and the number of clauses than those produced by τ_4 .

The two scatter plots (c) and (d) from Figure 1 report respectively the CPU times (in seconds) needed to compute the Decision-DNNF representations associated with the input WCNs (for each of the two encoding schemes τ_4 and $\tau_{4\text{linp}}$) and make precise the sizes (in number of arcs) of those Decision-DNNF representations.

Table 3 presents a selection of the results available from <http://www.cril.fr/KC> and used in the scatter plots from Figure 1. The columns of the table make precise, from the leftmost one to the rightmost one:

- data about the input instance, namely:
 - the family of the input WCN, among the six families considered in the experiments;
 - the type of the instance (Bayes net or Markov net);
 - the name of the instance;
 - the number of variables of the instance;
 - the number of tables of the instance;
 - the cardinality of (one of) the largest domain(s) of a variable of the instance;
 - the arity of (one of) the relations of the instance, of largest arity;
 - the total number of tuples in the instance (i.e., the sum of the cardinalities of the relations);
 - the sum of the cardinalities of the domains of the variables;
- and for each of the two encoding schemes τ_4 and $\tau_{4\text{linp}}$ under consideration:
 - the number of variables in the CNF encoding of the instance;
 - the number of clauses in the CNF encoding of the instance;
 - the time (in seconds) required to generate the CNF encoding, plus the time needed by C2D to generate a Decision-DNNF representation from it (and to minimize it when τ_4 has been used);
 - the size (in number of arcs) of the resulting Decision-DNNF representation produced by C2D (after minimization when τ_4 has been used).

Clearly enough, the scatter plots (c) and (d) from Figure 1 as well as Table 3 illustrate the benefits that can be achieved by considering $\tau_{4\text{linp}}$ instead of τ_4 when C2D is used downstream. Indeed, $\tau_{4\text{linp}}$ led most of the time to improved compilation times and improved compilation sizes. To be more precise, as to the compilation times, $\tau_{4\text{linp}}$ proved strictly better than τ_4 for 911 instances (while τ_4 proved strictly better than $\tau_{4\text{linp}}$ for 87 instances). As to the sizes

Figure 1: Comparing τ_{Alinp} with τ_4 .

Family	Name	Instance						#var	#cl	τ_4 time	C2D size	C2D size	#var	#cl	τ_{Alinp} time	C2D size	C2D size
		Type	#var	#Rel	max dom.	max ari.	#tuples										
Promedas	or_chain_96.fg	MARKOV	719	719	2	3	4260	1438	3058	4663	216.4	3058	1620	1942	111.5	1620	
Promedas	or_chain_223.fg	MARKOV	988	988	2	3	5754	1976	?	?	?	?	2268	2639	1427.4	2268	
Promedas	or_chain_178.fg	MARKOV	1021	1021	2	3	5936	2042	?	?	?	?	2314	2715	816.3	2314	
Promedas	or_chain_132.fg	MARKOV	723	723	2	3	4058	1446	3009	4522	95.6	3009	1563	1818	199.6	1563	
Promedas	or_chain_86.fg	MARKOV	892	891	2	3	5020	1784	3789	5602	499.6	3789	?	?	?	?	
Pedigree	pedigree23	MARKOV	402	402	5	4	5025	784	1479	2933	525.4	1479	737	1276	174.4	737	
Pedigree	pedigree30	MARKOV	1289	1289	5	5	12819	2491	4802	8860	1836.2	4802	2468	3802	1282.6	2468	
Pedigree	pedigree18	MARKOV	1184	1184	5	5	12198	2291	4407	8252	927.7	4407	2262	3560	1140.9	2262	
Grids	50-20-8	BAYES	400	400	2	3	3042	800	2556	3428	872.8	2556	1756	1887	1073.0	1756	
Grids	90-46-1	BAYES	2116	2116	2	3	16562	4232	?	?	?	?	3503	6727	87.9	3503	
Grids	90-42-2	BAYES	1764	1764	2	3	13778	3528	6228	13756	57.2	6228	2700	5513	48.6	2700	
Grids	90-50-7	BAYES	2500	2500	2	3	19602	5000	9222	19846	420.3	9222	?	?	?	?	
Grids	90-50-8	BAYES	2500	2500	2	3	19602	5000	?	?	?	?	4131	8048	145.7	4131	
Grids	75-26-4	BAYES	676	676	2	3	5202	1352	3020	5446	496.7	3020	1668	2468	376.4	1668	
Diagnose	3073	BAYES	329	329	6	12	34704	763	1695	3436	151.5	1695	1020	741	27.0	1020	
UAI	404.wcsp	MARKOV	100	710	4	3	4538	258	1678	3421	1653.5	1678	839	1037	777.1	839	
UAI	moissac4.pre	BAYES	462	462	3	3	7308	1386	2593	7338	39.7	2593	1669	3585	32.5	1669	
UAI	linkage_21	MARKOV	437	437	5	4	6698	941	1722	3638	1136.4	1722	?	?	?	?	
UAI	prob005.pddl	MARKOV	2701	29534	2	6	125726	5402	?	?	?	?	2701	29534	249.7	2701	
UAI	log-1	MARKOV	939	3785	2	5	16266	1878	5663	13393	45.0	5663	939	3785	11.4	939	
UAI	CSP_13	MARKOV	100	710	4	3	4538	258	?	?	?	?	839	1037	468.9	839	
Relational	blockmap_15.03-0003	BAYES	18787	18787	2	3	132436	37574	56451	141138	473.2	56451	18877	51827	152.4	18877	
Relational	blockmap_20.01-0009	BAYES	39297	39297	2	3	278138	78594	?	?	?	?	39334	108649	303.5	39334	
Relational	blockmap_22.02-0006	BAYES	56873	56873	2	3	405240	113746	?	?	?	?	56955	157979	625.8	56955	
Relational	mastermind_10.08.03-0004	BAYES	2606	2606	2	3	18658	5212	8250	19699	277.7	8250	3038	7446	176.5	3038	
Relational	blockmap_20.01-0008	BAYES	39297	39297	2	3	278138	78594	?	?	?	?	39334	108649	364.7	39334	
Relational	blockmap_22.03-0008	BAYES	59404	59404	2	3	423452	118808	?	?	?	?	59533	165085	490.0	59533	

Table 3: Comparing τ_{Alinp} with τ_4 . Each '?' means that the process aborted with a time-out or a memory-out.

of the compiled representations, τ_{Alinp} proved strictly better than τ_4 for 759 instances (while τ_4 proved strictly better than τ_{Alinp} for 239 instances). Using the τ_4 encoding scheme, C2D has been able to generate a Decision-DNNF for 903 instances over 1452 within the time and memory limits. Contrastingly, when equipped with τ_{Alinp} , C2D has been able to generate a Decision-DNNF for 1007 instances using the same computational resource bounds.

In order to evaluate the impact of the two "ideas" used in our encoding, we also performed a differential evaluation. Table 4 reports the number of instances for which the whole process – encoding+compilation+minimization (when needed) – terminated before the time limit, when the input encoding scheme is, respectively, τ_4 , τ_{4l} (log encoding of the indicator variables), τ_{4inp} (implicit encoding of the most frequent probability value per table), and τ_{Alinp} .

τ_4	903
τ_{4l}	975
τ_{4inp}	982
τ_{Alinp}	1007

Table 4: Number of instances compiled within a time limit of 900s.

The cactus plot given at Figure 2 makes precise for each of those four encodings, the number of instances processed successfully depending on the allocated time. Both Table 4 and Figure 2 show that each of the two "ideas" used in our encoding has a positive influence on the time needed to "compile" the input WCN.⁵

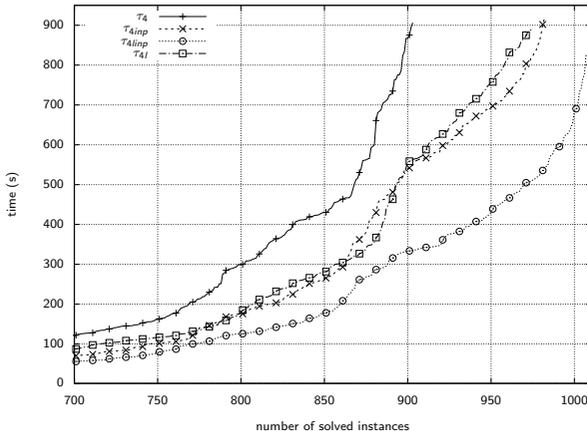


Figure 2: Number of instances compiled depending on the allocated time.

Finally, we also compared the performance of C2D empowered by our encoding scheme with those of ACE (version 3.0), a package that compiles a graphical model into an arithmetic circuit (AC) and then uses the AC to answer multiple queries with respect to the model, see <http://reasoning.cs.ucla.edu/ace>. In our experiments, logical model counting is used as a basis for compilation (we used the

⁵ The computation times reported for τ_{4l} are lower bounds, since they do not include the times required for achieving the minimization step w.r.t. the parameter variables. Indeed, this step has not been implemented. Especially, given that $\min(\Sigma_{4l}) \neq \min_{\theta}(\Sigma_{4l})$, it was not possible to take advantage of the "global cardinality minimization" functionality offered by C2D to compute $\min_{\theta}(\Sigma_{4l})$. Nevertheless, since cardinality minimization of a specific subset of variables is feasible efficiently from a Decision-DNNF representation, the approximation done does not question the conclusions drawn about the impact of the two "ideas" used in our encoding.

–forceC2d option of ACE for ensuring it). In this case, compilation proceeds by encoding the model into a propositional formula, compiling it into Decision-DNNF (using the C2D knowledge compiler), and extracting the AC from the compiled Decision-DNNF.

ACE is mainly based on **ENC4**, but incorporates several improvements; thus, *exactly_one* constraints (alias Eclauses) are generated in the encoding used by C2D (so that this encoding is not exactly a CNF encoding); such constraints are useful for representing the domains of the variables (they can replace the indicator clauses); furthermore, no parameter variable and no parameter clause are introduced for the $a \in \text{Dom}(R)$ such that $R(a) = 1$.

Like in the previous experiments reported in the paper, the comparison between $\tau_{Alinp}+C2D$ and ACE –forceC2d mainly concerns the generation (using C2D) of a Decision-DNNF representation from an input WCN. However, there is a fundamental difference: in the previous experiments, nothing changed but the encoding under consideration; for this reason, it was possible to draw firm conclusions about the relative efficiency of the encodings; in the comparison with ACE, the situation is different because the input of C2D when run within ACE does not simply consist of the encoding of the given WCN. Indeed, a dtree *derived from the input WCN* (using the well-known minfill heuristic) is considered as well for guiding the compilation process. This dtree may easily be distinct from the one considered by C2D when computed from the encoding, only, and may lead to improved compilation times and compilation sizes. Accordingly, one must keep in mind that the empirical protocol used for comparing $\tau_{Alinp}+C2D$ with ACE –forceC2d is not favorable to $\tau_{Alinp}+C2D$.

The scatter plots (a) and (b) from Figure 3 show respectively the compilation times and the compiled form sizes obtained by using $\tau_{Alinp}+C2D$ on the one hand, and ACE –forceC2d on the other hand. As to the compilation times, $\tau_{Alinp}+C2D$ proved strictly better than ACE –forceC2d for 335 instances (while ACE –forceC2d proved strictly better than $\tau_{Alinp}+C2D$ for 667 instances). As to the sizes of the compiled representations, $\tau_{Alinp}+C2D$ proved strictly better than ACE –forceC2d for 676 instances (while ACE –forceC2d proved strictly better than $\tau_{Alinp}+C2D$ for 326 instances). Overall, ACE –forceC2d has been able to generate a Decision-DNNF for 922 instances over 1452 within the time and memory limits. Contrastingly, $\tau_{Alinp}+C2D$ has been able to generate a Decision-DNNF for 1007 instances using the same computational resource bounds.

Empirically, ACE –forceC2d appeared as a better performer than $\tau_{Alinp}+C2D$ w.r.t. the compilation times. Here are two possible explanations for it. Firstly, the dtree computed derived from the input WCN can lead to a better decomposition, as explained above (this looks particularly salient for instances from the "Relational" family). Secondly, there are numerous instances for which ACE –forceC2d terminated within 10s, while $\tau_{Alinp}+C2D$ did not. This can be explained by the fact that each reported time actually covers all the computation time required by the process starting from the input WCN and finishing with the generation of the resulting Decision-DNNF representation. Especially, it includes the time required to generate the dtree used by C2D, and this dtree generation time can be much smaller when the generation process exploits the structure of the given WCN than when its input is just an encoding of the WCN. On the other hand, the combination $\tau_{Alinp}+C2D$ solved more instances than ACE –forceC2d within the time and memory limits and led to significantly smaller compiled representations in many cases. This is a further illustration of the practical benefits which can be achieved by taking advantage of our encoding τ_{Alinp} .

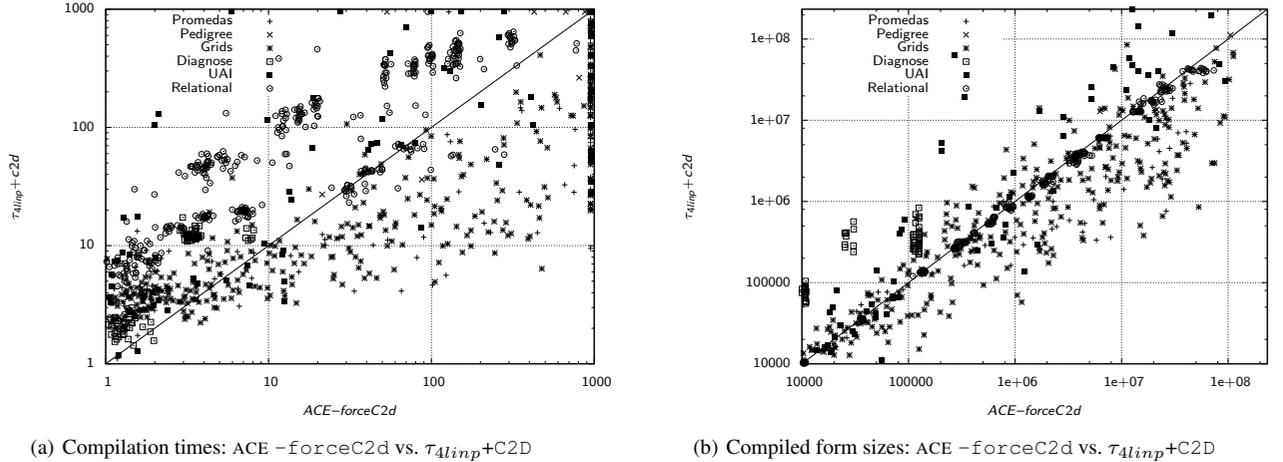


Figure 3: Comparing ACE-forceC2d vs. $\tau_{Alinp}+C2D$.

6 OTHER RELATED WORK

Interestingly, the key ideas used in τ_{Alinp} are not specific to the CNF encoding pointed out, but could also be exploited to define a CDNF encoding (i.e., a conjunction of DNF representations), which can serve as an input to the bottom-up SDD compiler [17]. This can prove useful since bypassing intermediate representations in CNF can lead in some cases to a more efficient compilation algorithm (sometimes by orders of magnitude) [11].

Let $\tau_{Alinp-sdd}$ be the translation leading to the WPROP $(\Sigma_{Alinp-sdd}, w_{Alinp-sdd}, (w_{Alinp-sdd})_0)$ where $\Sigma_{Alinp-sdd} = \bigwedge_{X \in \mathcal{X}} \tau_{Alinp-sdd}(X) \wedge \bigwedge_{R \in \mathcal{R}} \tau_{Alinp-sdd}(R)$. We define $\tau_{Alinp-sdd}(X) = \tau_{Alinp}(X)$ for every $X \in \mathcal{X}$. Then for every $R \in \mathcal{R}$, $\tau_{Alinp-sdd}(R)$ is a simplified DNF formula computed from the compressed representation of R as the disjunction of all terms $\tau_{Alinp-sdd}(\mathbf{a})$ for $\mathbf{a} \in Dom(R)$ such that $R(\mathbf{a}) = w(\theta_R)$, and all terms $\tau_{Alinp-sdd}(\mathbf{a}) \wedge \theta_j$ for $\mathbf{a} \in Dom(R)$ such that $R(\mathbf{a}) \neq 0$ and $R(\mathbf{a}) \neq w(\theta_R)$. The simplification step is achieved using Quine/McCluskey algorithm.⁶ Let us finally define $w_{Alinp-sdd}$ as w_{Alinp} (and $(w_{Alinp-sdd})_0 = (w_{Alinp})_0$).

Proposition 3 $\tau_{Alinp-sdd}$ is faithful.

Proof: The result comes easily from the fact that τ_{Alinp} is faithful and that under $\bigwedge_{X \in \mathcal{X}} \tau_{Alinp-sdd}(X)$ (equivalent to $\bigwedge_{X \in \mathcal{X}} \tau_{Alinp}(X)$), each DNF formula $\tau_{Alinp-sdd}(R)$ is equivalent to the CNF formula $\tau_{Alinp}(R)$. ■

Example 4 (Example 1 continued) $\tau_{Alinp-sdd}(R)$ is computed by considering first the DNF representation reported in the next table (left part), where the last line corresponds to a don't care.

λ_1^1	λ_1^0	λ_2	θ_1
0	0	1	
1	0		
0	1		1
1	1		

λ_1^1	λ_1^0	λ_2	θ_1
1	0	1	
	1		1

This DNF representation is then simplified, leading to the DNF representation reported in the table (right part), equivalent to

$$\lambda_1^1 \vee (\neg \lambda_1^0 \wedge \lambda_2) \vee (\lambda_1^0 \wedge \theta_1).$$

⁶ Terms conflicting with $\bigwedge_{X \in scope(R)} \tau_{Alinp-sdd}(X)$ can also be added as don't cares prior to the simplification step; this may lead to smaller DNF representations.

This DNF representation is also equivalent under $\bigwedge_{X \in \mathcal{X}} \tau_{Alinp}(X) = \neg \lambda_1^1 \vee \neg \lambda_1^0$ to

$$\tau_{Alinp}(R) = (\lambda_1^0 \vee \lambda_1^1 \vee \lambda_2) \wedge (\lambda_1^1 \vee \neg \lambda_1^0 \vee \theta_1).$$

7 CONCLUSION

We have presented a new CNF encoding scheme τ_{Alinp} for reducing probabilistic inference from a graphical model to weighted model counting. This scheme takes advantage of log encodings of the elementary variable/value assignments and of the implicit encoding of the most frequent probability value per conditional probability table. We have proved that τ_{Alinp} is faithful. Experiments have shown that τ_{Alinp} can be useful in practice; especially, the C2D compiler empowered by it performs in many cases significantly better than when ENC4 is used, or when ACE is considered instead.

This work opens several perspectives for further research. From the practical side, we set a time limit to 900s in our experiments and we did not repeat the computations with C2D because the number of instances considered (1452) was large. However, default settings of C2D uses randomization to generate dtrees, which guide the compilation process and may have a huge impact on the total process. Thus we plan to repeat the experiments a few times with a greater time limit, averaging the obtained results to minimize the effect of randomization. On a different, yet empirical perspective, we plan also to compare the performances of $\tau_{Alinp}+C2D$ with those of ACE-forceC2d, when C2D is guided in both cases by a dtree derived from the input network.

On the other hand, instead of associating specific weights with the negative parameter literals, it would be enough to ask (via the introduction of a further constraint) that at most one parameter variable for any relation $R \in \mathcal{R}$ is set to true. Our preliminary investigation showed that, empirically, this approach is less efficient than τ_{Alinp} when one considers the compilation times obtained by C2D used downstream, but also that it leads to compiled representations which are typically of smaller sizes. It would be interesting to look for a trade-off by taking advantage of the two approaches (introducing specific weights for negative parameter literals for some R and introducing *at most one* constraints for other R). In the future, we plan also to evaluate in practice the benefits offered by such approaches when Decision-DNNF is targeted, and by the $\tau_{Alinp-sdd}$ translation when SDD is targeted.

REFERENCES

- [1] A. Antonucci, Y. Sun, C. Polpo de Campos, and M. Zaffalon, 'Generalized loopy 2u: A new algorithm for approximate inference in credal networks', *Int. J. Approx. Reasoning*, **51**(5), 474–484, (2010).
- [2] F. Bacchus, S. Dalmao, and T. Pitassi, 'Algorithms and complexity results for #sat and Bayesian inference', in *Proc. of FOCS'03*, pp. 340–351, (2003).
- [3] F. Bacchus, S. Dalmao, and T. Pitassi, 'Value elimination: Bayesian inference via backtracking search', in *Proc. of UAI'03*, pp. 20–28, (2003).
- [4] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller, 'Context-specific independence in bayesian networks', in *Proc. of UAI'96*, pp. 115–123, (1996).
- [5] S. Chakraborty, D. J. Fremont, K. S. Meel, S. A. Seshia, and M. Y. Vardi, 'Distribution-aware sampling and weighted model counting for SAT', in *Proc. of AAAI'14*, pp. 1722–1730, (2014).
- [6] S. Chakraborty, D. Fried, K.S. Meel, and M.Y. Vardi, 'From weighted to unweighted model counting', in *Proc. of IJCAI'15*, pp. 689–695, (2015).
- [7] M. Chavira and A. Darwiche, 'Compiling bayesian networks with local structure', in *Proc. of IJCAI'05*, pp. 1306–1312, (2005).
- [8] M. Chavira and A. Darwiche, 'Encoding CNFs to empower component analysis', in *Proc. of SAT'06*, pp. 61–74, (2006).
- [9] M. Chavira and A. Darwiche, 'Compiling Bayesian networks using variable elimination', in *Proc. of IJCAI'07*, pp. 2443–2449, (2007).
- [10] M. Chavira and A. Darwiche, 'On probabilistic inference by weighted model counting', *Artificial Intelligence*, **172**(6-7), 772–799, (2008).
- [11] A. Choi, D. Kisa, and A. Darwiche, 'Compiling probabilistic graphical models using sentential decision diagrams', in *Proc. of ECSQARU'13*, pp. 121–132, (2013).
- [12] A. Darwiche, 'Decomposable negation normal form', *Journal of the ACM*, **48**(4), 608–647, (2001).
- [13] A. Darwiche, 'A compiler for deterministic decomposable negation normal form', in *AAAI'02*, pp. 627–634, (2002).
- [14] A. Darwiche, 'A logical approach to factoring belief networks', in *Proc. of KR'02*, pp. 409–420, (2002).
- [15] A. Darwiche, 'New advances in compiling CNF into decomposable negation normal form', in *Proc. of ECAI'04*, pp. 328–332, (2004).
- [16] A. Darwiche, *Modeling and Reasoning with Bayesian Networks*, Cambridge University Press, 2009.
- [17] A. Darwiche, 'SDD: A new canonical representation of propositional knowledge bases', in *Proc. of IJCAI'11*, pp. 819–826, (2011).
- [18] F. J. Díez and S. F. Galán, 'Efficient computation for the noisy MAX', *Int. J. of Intelligent Systems*, **18**(2), 165–177, (2003).
- [19] C. P. Gomes, J. Hoffmann, A. Sabharwal, and B. Selman, 'From sampling to model counting', in *Proc. of IJCAI'07*, pp. 2293–2299, (2007).
- [20] C. P. Gomes, A. Sabharwal, and B. Selman, 'Model counting', in *Handbook of Satisfiability*, 633–654, (2009).
- [21] J.-M. Lagniez and P. Marquis, 'Preprocessing for propositional model counting', in *Proc. of AAAI'14*, pp. 2688–2694, (2014).
- [22] D. Larkin and R. Dechter, 'Bayesian inference in the presence of determinism', in *Proc. of AISTATS'03*, (2003).
- [23] W. Li, P. Poupart, and P. van Beek, 'Exploiting structure in weighted model counting approaches to probabilistic inference', *J. of Artificial Intelligence Research*, **40**, 729–765, (2011).
- [24] E.J. McCluskey, 'Minimization of Boolean functions', *Bell System Technical Journal*, **35**(6), 1417–1444, (1956).
- [25] Ch.J. Muise, Sh.A. McIlraith, J.Ch. Beck, and E.I. Hsu, 'Dsharp: Fast d-DNNF compilation with sharpSAT', in *Proc. of AI'12*, pp. 356–361, (2012).
- [26] U. Oztok and A. Darwiche, 'On compiling CNF into Decision-DNNF', in *Proc. of CP'14*, pp. 42–57, (2014).
- [27] U. Oztok and A. Darwiche, 'A top-down compiler for sentential decision diagrams', in *Proc. of IJCAI'15*, pp. 3141–3148, (2015).
- [28] D. Poole and N.L. Zhang, 'Exploiting contextual independence in probabilistic inference', *J. of Artificial Intelligence Research*, **18**, 263–313, (2003).
- [29] W.V.O. Quine, 'The problem of simplifying truth functions', *American Mathematical Monthly*, **59**, 521–531, (1952).
- [30] W.V.O. Quine, 'A way to simplify truth functions', *American Mathematical Monthly*, **62**, 627–631, (1955).
- [31] M. Samer and S. Szeider, 'Algorithms for propositional model counting', *J. Discrete Algorithms*, **8**(1), 50–64, (2010).
- [32] T. Sang, F. Bacchus, P. Beame, H.A. Kautz, and T. Pitassi, 'Combining component caching and clause learning for effective model counting', in *Proc. of SAT'04*, (2004).
- [33] T. Sang, P. Beame, and H. A. Kautz, 'Performing Bayesian inference by weighted model counting', in *Proc. of AAAI'05*, pp. 475–482, (2005).
- [34] M. Takikawa and B. D'Ambrosio, 'Multiplicative factorization of noisy-max', in *Proc. of UAI'99*, pp. 622–630, (1999).
- [35] M. Thurley, 'sharpSAT - counting models with advanced component caching and implicit BCP', in *Proc. of SAT'06*, pp. 424–429, (2006).
- [36] J. Vomlel and P. Tichavský, 'Probabilistic inference in BN2T models by weighted model counting', in *Proc. of SCAI'13*, pp. 275–284, (2013).
- [37] M. Wachter and R. Haenni, 'Logical compilation of Bayesian networks with discrete variables', in *Proc. of ECSQARU'07*, pp. 536–547, (2007).

A Bayesian Approach to Norm Identification

Stephen Cranefield¹ and Felipe Meneguzzi² and Nir Oren³ and Bastin Tony Roy Savarimuthu⁴

Abstract. When entering a system, an agent should be aware of the obligations and prohibitions (collectively *norms*) that affect it. Existing solutions to this *norm identification* problem make use of observations of either norm compliant, or norm violating, behaviour. Thus, they assume an extreme situation where norms are typically violated, or complied with. In this paper we propose a Bayesian approach to norm identification which operates by learning from both norm compliant and norm violating behaviour. We evaluate our approach's effectiveness empirically and compare its accuracy to existing approaches. By utilising both types of behaviour, we not only overcome a major limitation of such approaches, but also obtain improved performance over the state of the art, allowing norms to be learned with fewer observations.

1 Introduction

Norms, as instantiated through obligations, permissions and prohibitions, are a popular approach to declarative behaviour specification within multi-agent systems [25, 21, 7, 1]. Such norms describe the expected behaviour of agents, but can be violated in exceptional circumstances. A large body of work exists on how agents should behave in the presence of norms [10, 3, 14, 15, 13]. Recently, work has emerged addressing *norm identification*—how an agent can identify norms already present in an environment. This problem is important in open, dynamic multi-agent systems, where agents can enter and leave the system at any time, and no assumption regarding norm knowledge can be made. While it is often assumed that norms can be communicated to agents when they enter a system [20], factors such as limited bandwidth, implicit norms (in some systems), lack of a shared ontology, malicious behaviour and changing norms can invalidate this assumption, instead requiring that agents be able to identify norms dynamically.

Previous work on dynamic norms often focuses on the consequences of norm emergence to society [17], that is, evaluating what happens to a society when norms change. However, only recently have researchers started to investigate practical approaches to the problem posed to individual agents of *inferring* new norms as they emerge. Such work often makes a combination of assumptions regarding what available evidence can actually be used to identify new norms. The work of Savarimuthu et al. [18, 19], is a typical example of an existing approach to norm recognition, based on the detection of a *sanctioning signal*—an action responding to an agent's norm violation that may (possibly) be performed by a peer of the agent or an institutional authority. Crucially, it is assumed that these signals may be recognised as conveying some negative emotional or institutional force, even before details of the specific norms in the society have

been inferred⁵. By learning the situations in which such sanctioning signals arise, agents are able to infer their triggering norms. However, while such an approach works well when sanctioning signals are common, it is more difficult to apply in systems where agents (largely) comply with norms.

To address this difficulty, Oren and Meneguzzi [16] introduced a plan recognition based mechanism for norm identification. In their approach, by observing the behaviour of agents, and identifying what states these agents avoid or always achieve, prohibitions and obligations can be identified. However, in its simplest form, this approach must assume fully norm-compliant behaviour. An adaptation suggested by Oren and Meneguzzi overcomes this limitation, but is too memory intensive to be practical in any reasonably sized domain.

There have been other works in the realm of norm identification [6, 2, 11]. The work done in the EMIL project by Campenni et al. [6] infers norms using observed behavioural patterns based on a threshold-based approach, where the observations could be from a range of sources: deontic commands, evaluative statements and assertions made by agents that are being observed. Based on aggregating this information, an agent could infer potential norms. For example if behaviour A is more prevalent than behaviour B in a given context, then A is considered to be a norm. This work assumes that an observer already knows how to interpret the normative statements, and hence has an implicit notion of a norm. However, in real life an observer new to a society may not have prior knowledge of what the norms might be and why an agent is being sanctioned. The work of Alrawagfeh et al. [2] aims to extract permission norms similarly to that of Campenni et al. [6], and prohibitions similarly to the work of Savarimuthu et al. [19, 18], thus suffering from the limitations of these two approaches. Additionally, this work does not infer obligations. The work of Mahmoud et al. [11], like the work of Savarimuthu et al. [19, 18], requires a sanctioning signal in order to function.

Existing work on norm identification therefore assumes that norms are almost always either complied with or violated [11, 19, 18, 16, 2], and is not appropriate in less extreme (and more realistic) cases. The core contribution of this paper is *an approach to norm identification that operates well in domains where both norm compliance and violation can regularly occur*. Thus we relax the strong assumptions of all existing work and develop an algorithm that can infer norms using a variety of possible sources of evidence. Our approach, described in Section 2, uses Bayesian inference to compute for each candidate norm the odds that it is an established norm, compared to the null hypothesis that there are no norms, given observations of other agents' behaviour. To act in a norm compliant way, an agent uses these odds to select which norms should be followed. We report on an empirical evaluation of our approach in Section 3, which

¹ University of Otago, New Zealand, email: stephen.cranefield@otago.ac.nz

² Pontifical Catholic University of Rio Grande do Sul, Brazil, email: felipe.meneguzzi@puers.br

³ University of Aberdeen, UK, email: n.oren@abdn.ac.uk

⁴ University of Otago, New Zealand, email: tony.savarimuthu@otago.ac.nz

⁵ For example, in human society, observing someone shouting or gesturing angrily at another can be understood as a message of displeasure, even without overhearing their conversation. Similarly, if we observe a policeman issuing an infringement notice to another, we are aware that a violation has been detected, even if we do not know the details.

shows two key results: first, that norm-compliant behaviour is possible after relatively few observations of the actions of others; and second, that our approach outperforms existing approaches to norm identification. Finally, we contextualise our work and point to future research directions in Section 4.

An extended abstract of this paper has been published previously [8]. That did not provide details of the approach, and reported on some preliminary experimental results. In this paper we provide a full account of the models and algorithms used in the work, and discuss a new experimental evaluation.

2 The Model and Approach

The first question that we must address is how our normative system should be encoded. There is a long history of utilising transition systems to model agents within a multi-agent system. These transition systems model the state space as nodes in a graph, and actions are encoded as edges allowing transitions between nodes (states). Therefore, following Oren and Meneguzzi [16], we consider an abstract normative environment where norms govern motion through such a graph, and seek to identify legal and illegal paths within this graph. As in transition systems, nodes in the graph represent individual states, while edges represent transitions through the space due to agent actions. However, unlike the graphs used in transition systems, we abstract away from the interpretation function used to associate values with variables in each state, and instead consider only motion through the graph itself. Therefore, a path within such a graph represents the actions of an agent following a plan to transition from some initial state (its start node) to a goal state (its destination node). One such graph is shown in Figure 1.

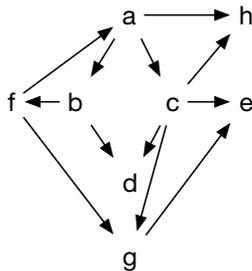


Figure 1: A sample graph

We conceptualise plans as sequences of nodes, and make no assumptions about the source of those plans: they may be generated dynamically given a goal and a set of possible actions, or they may come from a plan library, such as a BDI agent program. Our norm identification mechanism is based on the assumption that the observed agents' plan libraries (or available actions and planning mechanism) are known to the observing agent, at least at some level of abstraction. This would be the case if all agents share the same plan library, if their possible plans can be inferred from public knowledge about the problem domain, e.g. public transport routes and timetables, or (as assumed by Oren and Meneguzzi [16]) if the observing agent has a plan recognition mechanism. Alternatively, in the absence of any other information, an agent may have no other option than to simply assume that other agents are like itself, in order to gain some traction on the norm identification problem. It is important to note that for the purposes of norm identification, agents only need to infer the plans of other agents that govern their *publicly observable* behaviour.

Identifying norms then involves observing the movements of others through the graph to identify their goals and the paths that cor-

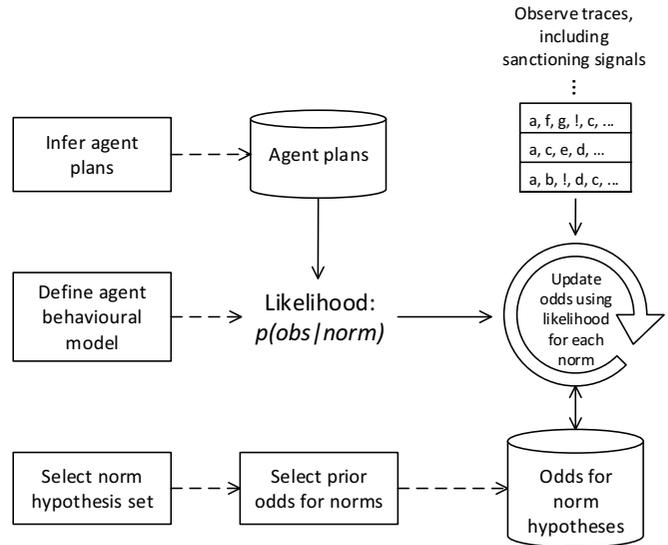


Figure 2: Overview of the approach

respond to plan executions. Given a set of norm hypotheses, we use observations as evidence to compute the odds of each hypothesis being a norm compared to the null hypothesis that there are no norms. In addition, we assume that when violations of norms occur, other agents may choose to sanction the offending agents, that these sanctioning actions can be recognised through sanctioning signals, and that these actions are spontaneously performed by agents rather than being generated by the plans they follow. These signals are another source of evidence that can be used to update the odds of the norm hypotheses.

Figure 2 illustrates our approach to norm identification. The left-most boxes in the diagram illustrate the inputs of our approach. The top left shows that, as discussed above, we must obtain an approximation (at least) of the plans that observed agents use to generate their publicly observable behaviour. We must also choose a set of candidate norms, as shown at the bottom left of the diagram. These are the hypotheses for which we iteratively compute their odds of being norms in the agent society (compared to the null hypothesis that there are no norms), as observations of agent behaviour are made. Section 2.1 describes our normative language and the instances of this language that form our hypothesis set in the graph traversal domain.⁶ The middle left of Figure 2 shows one other requirement of our approach. Our Bayesian approach to norm identification involves computing the likelihood of observed behaviour, given each candidate norm and the null hypothesis. As we are assuming that agent behaviour is generated by plans, we need a model explaining how agents choose which plans to follow in the presence of norms. This is discussed in Section 2.5. In addition, we assume that agents may choose to sanction others if they have violated a norm, and that this sanctioning behaviour is not part of the plan execution process, but rather a reactive process that runs in parallel with plan execution. We model this by the use of parameters specifying society-wide probabilities of observing and then choosing to sanction norm violations. We also consider the possibility of agents choosing to punish other agents for their own (non-normative) reasons, and model the chance of this occurring using another parameter. These parameters and their use in computing the likelihood of observations are discussed in Section 2.4.

⁶ The largest set of norm hypotheses arises if we consider all possible norm formulas generated by the normative language, which is what we use for our experiments.

The right hand side of Figure 2 shows the run-time Bayesian inference that updates the odds of the norm hypotheses as observations are made of agent behaviour. These observations are traces of agent movement on the graph, annotated with any sanctioning signals observed (denoted ‘!’ in the figure). Given prior odds for each norm hypothesis (we currently use a uniform prior distribution), Bayes’ Theorem explains how to update the odds for the norm hypotheses after making a new observation, by computing the likelihood of the observation under each hypothesis. This is discussed in Section 2.2.

2.1 Normative language

Our norm hypothesis space is defined by the following subset of linear temporal logic (LTL), where cn and n range over the labels of nodes in the graph⁷, \top denotes *true*, and \diamond and \circ denote “eventually” and “in the next state”, respectively.

$$\begin{aligned} \text{NORM} = & [\neg]\diamond n \mid cn \wedge \circ\top \rightarrow [\neg]\circ n \\ & \mid cn \wedge \circ\top \rightarrow [\neg]\circ\diamond n \end{aligned}$$

These norms are interpreted as obligations or prohibitions constraining the agents’ motion through the graph.

These three norm types, with and without the optional negation, are abbreviated and interpreted as follows (in the order shown above):

1. *eventually*(n) / *never*(n): These unconditional norms constrain a plan execution to include or exclude node n , and correspond to the obligation that n eventually occurs, or (respectively) that state n is prohibited.
2. *next*(cn, n) / *not.next*(cn, n): These are conditional obligations and prohibitions, triggered whenever the agent reaches node cn (we refer to this as the “condition node” for the norm) and the end state has not been reached⁸. In this case the norm states that node n must be (or, respectively, must not be) the next node reached. We restrict our norm hypotheses to only include *next* and *not.next* formulae for which there is an edge from cn to n in the graph.
3. *eventually*(cn, n) / *never*(cn, n): These are also conditional norms, expressing that *beginning from the node after the condition node*, node n must be eventually reached or (respectively) never reached.

Since *eventually* and *next* norms are obligations and *never* and *not.next* are prohibitions, they could alternatively be expressed using explicit deontic modalities, with temporal logic semantics for each modality that specify the traces in which future violations are deemed to occur (e.g. see the approach of Broersen et al. [5]). However, for our purpose in this paper, the syntax above and the semantics of violation given (later) in Table 1 are sufficient.

2.2 Bayesian updating

Bayesian approaches to machine learning make use of Bayes’ Theorem, which in its *diachronic interpretation* states how the probability of a hypothesis H should be updated in the light of new data D .

$$p(H|D) = \frac{p(H)p(D|H)}{p(D)}$$

The probability $p(H)$ is known as the *prior* probability of hypothesis H , $p(D|H)$ is the *likelihood* of the observed data D given the

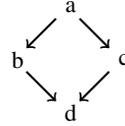
hypothesis, and $p(H|D)$ is the *posterior* probability of H given D . The denominator $p(D)$ is the probability of the data being observed under any hypothesis, and is a normalising term for the probabilities $p(H|D)$.

The calculation can be repeated as further data is observed by replacing the prior with the previously calculated posterior and using Bayes’ Theorem again to compute an updated posterior. This process is known as *Bayesian updating*.

If \mathbf{H} is a mutually exclusive and collectively exhaustive set of hypotheses, the denominator can be expanded as follows.

$$p(H|D) = \frac{p(H)p(D|H)}{\sum_{H' \in \mathbf{H}} p(H')p(D|H')}$$

However, the hypotheses of interest in a problem domain may not be mutually exclusive (independent of each other), and/or we may not be able to enumerate a finite set of hypotheses. This is the case when the hypotheses are norms that may hold in a society. Norms may not be independent of each other, and this can depend on the environment, for example, given the graph below and the goal of travelling from node a to node d , a norm prohibiting movement to node b after visiting node a has precisely the same effect as a norm obliging travel to c after visiting a .



2.3 Updating the odds of norms

When the normalising term $p(D)$ cannot be easily computed, e.g. because the hypotheses are not mutually exclusive and collectively exhaustive, an alternative approach to using Bayes’ Theorem is to work with *odds*. The odds of hypothesis H_1 over hypothesis H_2 , given some observed data D , is denoted $O(H_1:H_2|D)$ and is defined as follows:

$$\begin{aligned} O(H_1:H_2|D) &= \frac{p(H_1|D)}{p(H_2|D)} = \frac{p(H_1)p(D|H_1)/p(D)}{p(H_2)p(D|H_2)/p(D)} \\ &= O(H_1:H_2) \frac{p(D|H_1)}{p(D|H_2)} \end{aligned}$$

where $O(H_1:H_2) = \frac{p(H_1)}{p(H_2)}$ denotes the prior odds of H_1 with respect to H_2 .

In this formulation the normalising constant $p(D)$ cancels out and the odds of two competing hypotheses given new data can be computed using only the prior odds and the likelihoods of the two hypotheses. The probabilities of all other hypotheses do not need to be considered. In this paper we consider the odds of our hypotheses of interest (norms) compared to a specific null hypothesis: the hypothesis that there are no norms, written H_\emptyset . We write $O_\emptyset(H) = O(H:H_\emptyset)$ for the prior odds of H and $O_\emptyset(H|D) = O(H:H_\emptyset|D)$ for the posterior odds of H given D . By definition, $O_\emptyset(H_\emptyset) = 1$. For other norm hypotheses we set the prior odds uniformly to an arbitrary value less than one. The precise values of prior odds are unimportant for our work as we are interested in finding the norms with the *maximum* odds compared to the null hypothesis.

Whenever new data D is observed, we can then update the posterior odds for each norm hypothesis H by multiplying them by the ratio of the likelihoods of D given H and H_\emptyset . We consider two sources of evidence for norms. For each observation, we separately compute its likelihood based on (a) the observed sanctioning signals, and (b)

⁷ Technically, cn and n are *nominals* from Hybrid Logic [4, p.435]: propositional symbols that are constrained to be true in exactly one state.

⁸ Formally, the end state of a trace can be identified as the one in which $\circ\top$ is false.

```

procedure update-odds( $p, s, g, P, \mathbf{H}$ )
begin
  likelihood $_{H_0}^{sig} = p^{sig}(p, s | H_0)$ 
  likelihood $_{H_0}^{plans} = p^{plans}(p | H_0, g, P)$ 
  for  $n \in \mathbf{H}$ 
     $O_\emptyset(n) = O_\emptyset(n) * p^{sig}(p, s | n) / \text{likelihood}_{H_0}^{sig}$ 
     $O_\emptyset(n) = O_\emptyset(n) * p^{plans}(p | n, g, P) / \text{likelihood}_{H_0}^{plans}$ 
  end
end

```

Figure 3: The procedure for updating odds

Table 1: Violation indices $v_n(p)$ for the six norm types, given the path $p = \langle p_1, \dots, p_\ell \rangle$

Norm type	Violation indices
$eventually(n)$	$\{\ell\}$ if $\forall i \in \{1, \dots, \ell\} p_i \neq n$, else \emptyset
$never(n)$	$\{i : p_i = n\}$
$next(cn, n)$	$\{i+1 : 1 \leq i < \ell \wedge p_i = cn \wedge p_{i+1} \neq n\}$
$not.next(cn, n)$	$\{i+1 : 1 \leq i < \ell \wedge p_i = cn \wedge p_{i+1} = n\}$
$eventually(cn, n)$	$\{\ell\}$ if $\exists i \in \{1, \dots, \ell\} (p_i = cn \wedge \forall j \in \{i+1, \dots, \ell\} p_j \neq n)$ else \emptyset
$never(cn, n)$	\emptyset if $\forall i \in \{1, \dots, \ell\}, p_i \neq n$, else $\{j : \min(\{i : p_i = cn\}) < j \leq \ell \wedge p_j = n\}$

a plan-based approach, and update the odds based on each of these. Each observation consists of a path p and a set s of path indices at which sanctioning signals were observed. For our norm hypotheses we only consider a single norm at a time⁹, i.e., our hypothesis set \mathbf{H} consists of all norms from the language defined in Section 2.1.

The procedure for updating the odds¹⁰ for all norm hypotheses in the hypothesis set \mathbf{H} , given a new observation $\langle p, s \rangle$ is shown in Figure 3, where p^{sig} and p^{plans} are as defined in the following sections. The parameters passed to the update-odds function are the observed path and sanctioning signals, a goal g and set P of plans used by the plan-based likelihood computation, and the norm hypothesis set.

2.4 The likelihood of observed sanctions

We assume that agents may (sometimes) observe paths traversed by other agents in the graph. The observed paths represent possibly partial traversals of the graph by the other agents: they may be segments of longer paths traversed, but there are no unobserved nodes internal to the paths. Violations are detected through *signalling actions* (also known as *signals*) that indicate sanctioning of the observed agent [18, 19]. Such a signalling action could occur due to norm violation, or due to the sanctioner sanctioning the observed agent improperly (e.g., due to maliciousness, or a violation of the sanctioner’s personal values). We model the latter case by assuming there is a small population-wide probability p_{pun} of a non-normative punishment signal being observed after any step of an observed path. We also assume there are fixed probabilities of norm violations being observed (p_{obs}) and of observed violations being sanctioned (p_{sanc}). We model all signalling actions by a single symbol—we do not assume that sanctions are specific to particular norms, nor that agents can distinguish normative sanctions from non-normative punishments.

⁹ Our approach can be extended to consider hypotheses that are non-singleton norm sets, but we leave this for future work. If more than one norm can hold, it is still useful to identify the norms with the highest individual relative odds, before considering which sets of norms to add to the hypothesis space.

¹⁰ In our implementation, we work with log odds.

```

function choose-plan(goal, plans, norm)
1. poss-plans = plans(goal, plans)
2. Decide whether to be norm-compliant
3a. if norm-compliant
    nvp = non-viol-plans(poss-plans, norm)
    if nvp =  $\emptyset$ 
      return null
    else
      return random-weighted-choice(nvp)
3b. else
    if poss-plans =  $\emptyset$ 
      return null
    else
      return random-weighted-choice(poss-plans)

```

Figure 4: Model for an agent’s choice of plan

Given an observed trace annotated with sanctioning actions, we can compute the likelihood of this observation given a hypothesized norm as follows.

Let $p = \langle p_1, \dots, p_\ell \rangle$ be an observed path and the set s be a record of the indices of the path at which signals were observed.¹¹ The signal is represented by including index i in set s . We define $v_n(p)$ as the set of indices of the path p at which violations of the norm n occurred, defined in Table 1. The occurrence of $i \in v_n(p)$ is interpreted as the violation occurring *after* the action to move to node p_i , and may be a result either of that move or of the path ending if the destination node has been reached and an *eventually* norm is violated.

Given a norm hypothesis n , the likelihood of observing trace $p = \langle p_1, \dots, p_\ell \rangle$, where set s contains the indices at which sanction or punishment signals were observed, is then:

$$p^{sig}(p, s | n) = \prod_{1 \leq i \leq \ell} p_i^{sig}(i \in v_n(p), i \in s | n)$$

where p_i^{sig} , which takes two Boolean arguments, denotes the likelihood of the observation at path index i , as defined by the following table.

	$i \in s$	$i \notin s$
$i \in v_n(p)$	$p_{pun} + ((1 - p_{pun}) \cdot p_{obs} \cdot p_{sanc})$	$(1 - p_{pun}) \cdot (1 - p_{obs} \cdot p_{sanc})$
$i \notin v_n(p)$	p_{pun}	$1 - p_{pun}$

The first row of the table is for the case when a violation occurs at index i . If a signal is observed at i , then this is either a non-normative punishment or the violation was observed and sanctioned. If no signal is observed, then there is no punishment and the violation has not been both observed and sanctioned. When there is no violation at i (second row), a signal can only be a non-normative punishment, so the likelihood of a signal occurring (or not) is the probability of the punishment occurring (or not).

2.5 Likelihood using knowledge of agent plans

Following the approach of Oren and Meneguzzi [16] we can use knowledge of agent plans (e.g. through plan recognition [22]) to compute the likelihood of an observed path through the graph (ignoring any sanction or punishment signals). We assume that all agents

¹¹ We currently assume that sanctions are applied (if at all) *immediately* after a movement to a node p_i in the path causes a norm to be violated. Relaxing this assumption would require a more complex likelihood function that considers the possible matches of signals with possible past violations.

$$\begin{aligned}
p^{plans}(o | n, g, P) &= p_{comp} \left(\sum_{\pi \in \text{non-viol-plans}(\text{plans}(g, P), n)} \frac{\text{weight}(\pi)}{\sum_{\pi' \in \text{non-viol-plans}(\text{plans}(g, P), n)} \text{weight}(\pi')} p(o | \pi) \right) \\
&\quad + (1 - p_{comp}) \left(\sum_{\pi \in \text{plans}(g, P)} \frac{\text{weight}(\pi)}{\sum_{\pi' \in \text{plans}(g, P)} \text{weight}(\pi')} p(o | \pi) \right) \\
&= p_{comp} \frac{\sum_{\substack{\pi \in \text{non-viol-plans}(\text{plans}(g, P), n) \\ \cap \text{plans-containing}(\text{plans}(g, P), o)}} \text{weight}(\pi)}{\sum_{\pi \in \text{non-viol-plans}(\text{plans}(g, P), n)} \text{weight}(\pi)} + (1 - p_{comp}) \frac{\sum_{\substack{\pi \in \text{plans}(g, P) \\ \cap \text{plans-containing}(\text{plans}(g, P), o)}} \text{weight}(\pi)}{\sum_{\pi \in \text{plans}(g, P)} \text{weight}(\pi)}
\end{aligned}$$

Figure 5: Likelihood of an observed path using knowledge of agent plans

share the same set of possible plans (choices of paths in the graph), and that the observing agent can infer the observed agent’s goal (comprising starting and destination nodes).

To define the likelihood of an observed path given a norm hypothesis, a goal and a plan library, we require a model for the decision-making process of the observed agents, which must choose and execute plans to achieve their goals in the possible presence of a norm. The analysis in this section is based on the decision-making model shown in Figure 4.

In this model, the agent first generates all plans for the goal. The returned plans may be weighted, (e.g., to indicate agent preferences or execution costs), but our examples use equal weights for simplicity. Next, the agent decides whether it will act in a norm-compliant manner. If so, it filters the possible plans to keep only those that do not violate the norm, and chooses a plan using a random weighted choice. Otherwise, it makes a random weighted choice from the full set of plans for the goal. Note that this is intended to be a simple abstract model for the purpose of defining a likelihood function in the absence of any information about an observed agent. We do not claim that this is, or should be, the exact control mechanism used in any agent implementation.

We define the likelihood, based on knowledge of agent plans, of an observed path o on the graph, given a norm hypothesis n , an inferred goal g and a set of plans P , as shown in Figure 5. We write p_{comp} for the rate of norm compliance in the society. The first two lines of the figure multiply the probability of choosing a plan π and the probability $p(o | \pi)$ that the plan contains the observed path, for the norm-compliance and non-norm-compliance cases. As $p(o | \pi)$ is either 1 or 0, the last two lines replace this factor with a union in the limits of the sum. The function `non-viol-plans` filters out plans that cause violations, using the violation indices function v_n (Table 1).

There are two cases when the formula in Figure 5 cannot be evaluated due to zero values in the denominator of a fraction: when there are no plans for the inferred goal, and when there are no norm-compliant plans for the goal. The former case invalidates our assumption that the observed behaviour is generated using a plan taken from a known set of plans to fulfil the inferred goal, and we abandon the odds update based on plan knowledge for the current observation. In the latter case we replace the first addend in the last line of the figure with 0. This represents the assumption that a norm compliant agent would have abandoned its goal in this case.

3 Experiments

We have performed a set of experiments to validate and evaluate our approach to norm detection. These experiments use a random graph [9] containing 35 nodes, representing states in a state space,

and edges connecting these nodes represent actions available to an agent. The algorithm we used to generate observations works exactly like an agent randomly choosing possible plans to reach a goal state, subject to the normative constraints, with a certain probability. Table 2 summarises the parameters common to all experiments.

Table 2: Parameters used across experiments.

p_{pun}	0.01	p_{sanc}	0.99
p_{obs}	0.99	Prior odds $O_{\emptyset}(n)$	0.5

Here, we differentiate the norms used to generate the observations, which we call N_g , from the norms inferred by our norm detection approach, which we call N_d . Norm likelihood is estimated using log odds against there being no norm (rather than a probability), and we infer the norms from a set of observations by ranking the odds of each norm (against no norm), and consider a given number of the norms with the highest odds to be the norms in the society.

In the experiments, we ran an agent fully aware of the norms to generate a random set of observations following the algorithm of Figure 4 (i.e. random, but norm-compliant behaviour), while allowing for the possibility of occasional non-compliant behaviour with a probability set at 1% unless otherwise noted. Thus $(1 - p_{comp}) = 0.01$. We then applied our norm identification method to these observations in order to compute the odds of all possible norm hypotheses. Our experiments consisted of submitting a sequence of 100 observations to an agent and measuring its ability to produce norm-compliant behaviour after increasing numbers of observations. Each experiment was repeated 50 times and the results were averaged to reduce the impact of any particularly informative or uninformative sequence of observations.

Our aim is to allow an agent to undertake norm-compliant behaviour, even without an exact model of the norms. One approach to doing this is to consider the T most likely norms, even if they are less likely than the null hypothesis (i.e., they may not exist since they are less likely than there being no norm). In such a situation, the agent can be thought of as acting conservatively, as it may avoid potentially permitted courses of action. Another approach is to have the agent consider only norms that are more likely than the null hypothesis¹². In the remainder of this work, we consider the functioning of an agent utilising norm identification and acting using the conservative approach, and evaluate its effectiveness in generating norm compliant behaviour.

Our experiments measure precision and recall as a function of the number of observations supplied to the detection mechanism.

¹² However, this is dependent on the prior odds chosen.

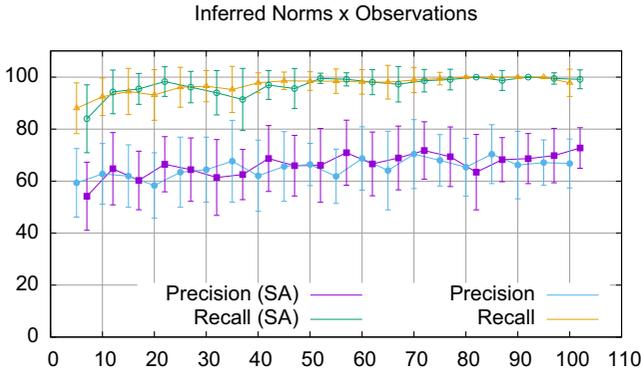


Figure 6: Inferred norms

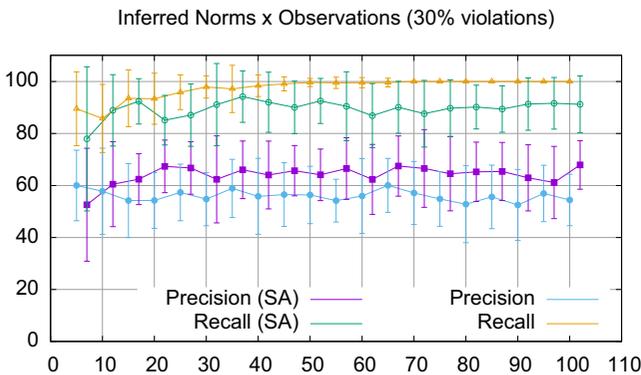


Figure 7: Inferred norms, with 30% violations

As norms may subsume each other, we measured precision and recall in terms of compliant behaviour rather than the exact norms inferred. Thus, in our evaluation, precision means the fraction of norm-compliant plan executions generated by an agent using a sample of the inferred norms N_d to drive its behaviour, and recall is the fraction of plans that are compliant with the true underlying norms N_g that are sampled by the agent. Specifically, we compute precision by updating the odds of each norm hypothesis after an observation and choosing the set N_d with the top T norms.

Our experiments are illustrated in the graphs of Figures 6–8. Our first set of experiments, illustrated in Figure 6, shows precision and recall as a function of the number of observations with error bars denoting standard deviation for these measures. Each experiment was conducted in the presence and absence of sanctioning actions—“SA” in the figure legends indicate the use of sanctioning actions.¹³ The results indicate that precision varies from a starting point of around 55 without sanctioning actions and 60 with them. Precision and recall increase rapidly in the first 10 observations and then tend to slowly increase as more observations are taken into consideration, while variance in precision diminishes as more observations are made.

Moreover, as we increase the amount of non-compliant behaviour in the observations, illustrated in Figure 7, the effect of sanctioning actions becomes more pronounced. Precision improves, while recall gets worse, as potential norm compliant plans are filtered out (possibly due to non-normative punishment actions) while fewer non-compliant plans are executed. Thus, as long as the agent has choices

¹³ Note that the values for the sanctioning action (SA) case are shifted two points to the right to prevent error bars from overlapping.

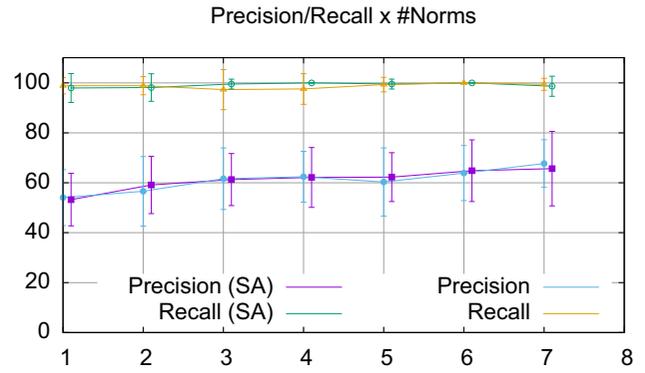


Figure 8: Precision and recall as a function of norms

available to itself with regards to plans which it can execute, sanctioning actions appear to be beneficial. It is also important to note that while recall is lower in the presence of sanctioning actions, it still remains relatively high.

We also computed precision and recall as a function of the number of norms in the system. In this experiment, we randomly selected elements of the power set $\mathcal{P}(N_g)$ of the set of seven norms in the scenario and, for each size of subset of $\mathcal{P}(N_g)$, we determined precision and recall after 100 observations. The results, shown in Figure 8, indicate a slight increase in precision and recall as more norms are introduced into the system. However, the highly overlapping error bars prevent us from asserting that the number of norms present in the system has an impact on the ability of the approach to infer norms.

Finally, we ran experiments comparing the effectiveness of our approach against the previous state-of-the-art approaches, each of which makes different assumptions about the behaviour of the underlying agents. We compared the Bayesian approach to the approaches of Savarimuthu et al. [18, 19], which assume that observed agents generate non-compliant behaviour for norms to be inferred, and that of Oren and Meneguzzi [16], which assumes that observed agents mostly comply with the norms¹⁴. In all of these comparative experiments we generated 20 observations and included sanctioning actions, since the approaches of Savarimuthu et al. relied on these signals. Like our previous experiment, we performed 50 repetitions to smooth out random variations in the observations. Unlike our previous experiments, which used 100 observations, we stopped at 20 observations, since by the 20th observation precision and recall had dropped to 0 for all comparable approaches. We conducted experiments with compliance ranging from almost full (a 0.01% probability of violations) to none (a 100% probability of violations). The results are presented in Table 3, which shows, for each approach, the mean of the following measures after a given number of observations (indicated by the #Obs column) have been made: the number of norms inferred, precision and recall. Standard deviations for each measure are shown in parentheses. Here, the Bayesian approach clearly outperforms previous approaches after a very small number of observations. For all competing approaches, precision and recall tend to drop towards 0 as the number of observations increase. Since these approaches infer norms more aggressively, they tend to over-constrain the agent’s behaviour, inferring many more norms than the Bayesian approach. As a result, the competing approaches tend to generate no “true positive” plan evaluations, that is, for plans that are compliant with the real underlying norms, these approaches tend to erroneously evaluate them as being non-compliant.

¹⁴ In the comparison we describe, a threshold of 0.5 was used in Oren and Meneguzzi’s second algorithm.

Table 3: Comparison of the Bayesian approach with the data mining [19, 18] and plan recognition [16] approaches. In each line, the best results are shown in bold.

#Obs	#Norms (σ)	Precision (σ)	Recall (σ)	#Obs	#Norms (σ)	Precision (σ)	Recall (σ)	#Obs	#Norms (σ)	Precision (σ)	Recall (σ)
Bayesian Approach				Data Mining Approach [19, 18]				Plan Recognition Approach [16]			
Probability of violation=0.01				Probability of violation=0.01				Probability of violation=0.01			
1	18.00 (0.00)	62.63 (27.50)	43.16 (28.08)	1	7.90 (2.14)	60.00 (48.98)	9.56 (9.60)	1	30.00 (0.00)	60.00 (48.98)	11.43 (10.46)
5	18.00 (0.00)	59.59 (15.91)	86.31 (12.81)	5	10.55 (2.97)	25.00 (43.30)	3.16 (5.88)	5	28.25 (1.84)	0.00 (0.00)	0.00 (0.00)
10	18.00 (0.00)	63.58 (8.94)	91.87 (7.22)	10	8.20 (2.42)	50.00 (50.00)	9.40 (11.07)	10	27.95 (1.85)	0.00 (0.00)	0.00 (0.00)
15	18.00 (0.00)	59.97 (11.64)	90.17 (12.48)	15	7.35 (1.15)	55.00 (49.74)	6.44 (6.28)	15	27.50 (2.71)	0.00 (0.00)	0.00 (0.00)
20	18.00 (0.00)	64.76 (12.24)	95.54 (7.16)	20	7.95 (2.29)	45.00 (49.74)	5.22 (6.47)	20	27.00 (2.38)	0.00 (0.00)	0.00 (0.00)
Probability of violation=0.3				Probability of violation=0.3				Probability of violation=0.3			
1	18.00 (0.00)	40.43 (32.15)	31.48 (29.33)	1	7.30 (2.12)	30.00 (45.82)	5.22 (9.02)	1	30.00 (0.00)	40.00 (48.98)	6.51 (10.20)
5	18.00 (0.00)	52.59 (21.72)	77.94 (27.72)	5	11.75 (3.25)	15.00 (35.70)	1.37 (3.27)	5	27.25 (2.38)	0.00 (0.00)	0.00 (0.00)
10	18.00 (0.00)	60.50 (16.34)	88.92 (13.73)	10	13.45 (3.84)	10.00 (30.00)	1.26 (4.18)	10	26.50 (2.01)	0.00 (0.00)	0.00 (0.00)
15	18.00 (0.00)	62.42 (9.77)	92.37 (8.74)	15	15.20 (4.28)	5.00 (21.79)	0.38 (1.67)	15	26.15 (2.72)	0.00 (0.00)	0.00 (0.00)
20	17.95 (0.21)	67.36 (10.12)	85.14 (9.54)	20	16.45 (3.33)	5.00 (21.79)	0.90 (3.96)	20	25.15 (2.39)	0.00 (0.00)	0.00 (0.00)
Probability of violation=0.6				Probability of violation=0.6				Probability of violation=0.6			
1	18.00 (0.00)	58.48 (34.09)	34.77 (30.56)	1	6.40 (2.53)	70.00 (45.82)	10.62 (9.87)	1	30.00 (0.00)	80.00 (40.00)	14.82 (11.65)
5	18.00 (0.00)	54.58 (14.59)	82.34 (15.77)	5	13.50 (3.78)	15.00 (35.70)	1.59 (3.95)	5	27.20 (1.99)	0.00 (0.00)	0.00 (0.00)
10	18.00 (0.00)	55.63 (11.26)	82.41 (16.93)	10	13.20 (3.52)	15.00 (35.70)	2.85 (8.91)	10	27.35 (2.57)	0.00 (0.00)	0.00 (0.00)
15	18.00 (0.00)	58.57 (13.48)	87.97 (9.58)	15	15.75 (4.49)	0.00 (0.00)	0.00 (0.00)	15	26.45 (2.20)	0.00 (0.00)	0.00 (0.00)
20	18.00 (0.00)	62.34 (10.90)	81.03 (12.50)	20	17.05 (4.42)	0.00 (0.00)	0.00 (0.00)	20	25.70 (2.23)	0.00 (0.00)	0.00 (0.00)
Probability of violation=1				Probability of violation=1				Probability of violation=1			
1	18.00 (0.00)	35.13 (35.01)	22.97 (23.26)	1	6.10 (2.56)	45.00 (49.74)	8.80 (11.05)	1	30.00 (0.00)	60.00 (48.98)	8.93 (9.38)
5	17.95 (0.21)	61.71 (26.40)	50.87 (30.00)	5	11.40 (5.04)	5.00 (21.79)	0.38 (1.67)	5	27.45 (1.59)	0.00 (0.00)	0.00 (0.00)
10	18.00 (0.00)	52.05 (20.25)	69.37 (28.50)	10	12.15 (4.87)	0.00 (0.00)	0.00 (0.00)	10	27.15 (1.45)	0.00 (0.00)	0.00 (0.00)
15	18.00 (0.00)	63.59 (20.83)	73.97 (24.45)	15	17.25 (6.54)	0.00 (0.00)	0.00 (0.00)	15	26.50 (1.96)	0.00 (0.00)	0.00 (0.00)
20	17.95 (0.21)	56.83 (19.22)	74.22 (21.27)	20	16.90 (5.65)	0.00 (0.00)	0.00 (0.00)	20	26.15 (1.98)	0.00 (0.00)	0.00 (0.00)

We note that, given the different expressivity of the competing approaches, and the fact that our experiments used norms expressed in the temporal modalities used in our approach, there may be a mismatch in the detection capabilities within the experiments. Nevertheless, since our experiments measured precision and recall in terms of *compliant behaviours* rather than the specific norms, we believe that our analysis is valid.

4 Discussion and Conclusions

The Bayesian approach presented in this paper combines ideas from the sanctioning action observation [19, 18] and plan recognition [16] norm identification approaches to create a powerful new mechanism. As our experiments indicate, we generate norm-compliant behaviour in a norm-identifying agent approximately 60% of the time across a range of violation likelihoods, and show that the presence of sanctioning actions substantially improves recall.

As mentioned in Section 2, we assume that we can determine an agent's starting point and goal. If we consider AgentSpeak(L) style agents, then the identification of a plan, and from this its context and guard conditions, would allow an agent to determine the observed agent's start point in many situations. Furthermore, once a plan has been identified, its goal can be trivially determined. While we assume that sanctioning actions (signals) are observable, we do not assume that it is possible to associate specific norms with specific signals. We also assume that one cannot differentiate between sanction and punishment signals. Lifting such restrictions would improve the learning rate, but is not realistic.

We intend to pursue several avenues of future work. First, while our model permits it, we have not evaluated the effects of conflicting norms on the norm identification process or on the conservative strategy described in this paper, and we intend to investigate what additional mechanisms must be created to function in such domains. Given that norms can subsume others, we believe the use of a subsumption-based norm conflict resolution mechanism [24] would result in an agent with an enhanced ability to identify norms and act in a norm compliant manner. We also plan to investigate weighting mechanisms and their effects on norm identification. Such mechanisms could, for example, originate from a trust and reputation model [12]. Here, highly trusted agents could be assumed to (normally) act in a norm-compliant manner, while less trustworthy agents would be expected to trigger more sanctioning actions. The addition of trust information could allow us to consider which agents are performing

the sanctioning actions. An agent often signalling that a trustworthy agent is acting in a norm-violating manner could have its opinion discounted, while those signalling that untrustworthy agents are violating norms could have their opinion strengthened. The addition of such a mechanism should improve the performance of our model, and given the Bayesian underpinnings of many trust systems [23], should be a relatively straightforward addition.

Another source of weightings we could exploit within the model originates from the plans themselves. In this work, we assumed that all plans to achieve some goal are equally likely to be used by an agent. However, some of these plans could be more expensive (e.g. from a resource utilisation point of view) than others, and a utility maximising agent would be expected to select cheaper plans (subject to normative constraints). We believe that the use of such weights would increase the rate at which norms are identified, and also increase the precision and recall of our approach.

ACKNOWLEDGEMENTS

F. Meneguzzi thanks Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) through the Universal Grant (Grant ref. 482156/2013-9) and PQ fellowship (Grant ref. 306864/2013-4).

REFERENCES

- [1] Huib Aldewereld, Frank Dignum, Andrés García-Camino, Pablo Noriega, Juan A. Rodríguez-Aguilar, and Carles Sierra, 'Operationalisation of norms for usage in electronic institutions', in *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 223–225. ACM, 2006.
- [2] Wagdi Alrawagfeh, Edward Brown, and Manrique Mata-Montero, 'Norms of behaviour and their identification and verification in open multi-agent societies', *International Journal of Agent Technologies and Systems*, 3(3), 1–16, 2011.
- [3] *Normative Multi-Agent Systems*, eds., Giulia Andrighetto, Guido Governatori, Pablo Noriega, and Leendert W. N. van der Torre, volume 4 of *Dagstuhl Follow-Ups*, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2013.
- [4] P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*, Cambridge University Press, 2001.
- [5] Jan Broersen, Frank Dignum, Virginia Dignum, and John-Jules Ch. Meyer, 'Designing a deontic logic of deadlines', in *Deontic Logic in Computer Science*, volume 3065 of *Lecture Notes in Computer Science*, 43–56, Springer, 2004.
- [6] Marco Campenní, Giulia Andrighetto, Federico Cecconi, and Rosaria Conte, 'Normal = Normative? The role of intelligent agents in norm innovation', *Mind & Society*, 8(2), 153–172, 2009.

- [7] Cristiano Castelfranchi, Frank Dignum, Catholijn M. Jonker, and Jan Treur, 'Deliberative normative agents: Principles and architecture', in *Intelligent Agents VI. Agent Theories, Architectures, and Languages*, 364–378, Springer, 2000.
- [8] Stephen Cranefield, Tony Savarimuthu, Felipe Meneguzzi, and Nir Oren, 'A Bayesian approach to norm identification (extended abstract)', in *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1743–1744. IFAAMAS, 2015.
- [9] P. Erdős and A. Rény, 'On random graphs', *Publicationes Mathematicae*, **6**, 290–297, 1959.
- [10] Michael Luck, Samhar Mahmoud, Felipe Meneguzzi, Martin Kollingbaum, Timothy J. Norman, Natalia Criado, and Moser Silva Fagundes, 'Normative agents', in *Agreement Technologies*, ed., Sascha Ossowski, volume 8 of *Law, Governance and Technology Series*, 209–220, Springer, 2013.
- [11] Moamin A. Mahmoud, Mohd Sharifuddin Ahmad, Azhana Ahmad, Mohd Zaliman Mohd Yusoff, and Aida Mustapha, 'The semantics of norms mining in multi-agent systems', in *Computational Collective Intelligence. Technologies and Applications*, volume 7653 of *Lecture Notes in Computer Science*, 425–435, Springer, 2012.
- [12] S. Marsh, *Formalising trust as a computational concept*, Ph.D. dissertation, University of Stirling, 1994.
- [13] Felipe Meneguzzi, Odinaldo Rodrigues, Nir Oren, Wamberto W. Vasconcelos, and Michael Luck, 'BDI reasoning with normative considerations', *Engineering Applications of Artificial Intelligence*, **43**, 127–146, 2015.
- [14] Felipe Rech Meneguzzi and Michael Luck, 'Norm-based behaviour modification in BDI agents', in *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, pp. 177–184. IFAAMAS, 2009.
- [15] Sanjay Modgil, Nir Oren, Noura Faci, Felipe Meneguzzi, Simon Miles, and Michael Luck, 'Monitoring compliance with e-contracts and norms', *Artificial Intelligence and Law*, **23**(2), 161–196, 2015.
- [16] Nir Oren and Felipe Meneguzzi. Norm identification through plan recognition. Proceedings of the 15th International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems, <http://www.staff.science.uu.nl/~dignu101/coin2013/papers/20130161.pdf>, 2013.
- [17] Bastin Tony Roy Savarimuthu, Stephen Cranefield, Martin K Purvis, and Maryam A Purvis, 'Norm emergence in agent societies formed by dynamically changing networks', *Web Intelligence and Agent Systems*, **7**(3), 223–232, 2009.
- [18] Bastin Tony Roy Savarimuthu, Stephen Cranefield, Maryam A. Purvis, and Martin K. Purvis, 'Obligation norm identification in agent societies', *Journal of Artificial Societies and Social Simulation*, **13**(4), 3, 2010.
- [19] Bastin Tony Roy Savarimuthu, Stephen Cranefield, Maryam A. Purvis, and Martin K. Purvis, 'Identifying prohibition norms in agent societies', *Artificial Intelligence and Law*, **21**(1), 1–46, 2013.
- [20] Constantin Serban and Naftaly H. Minsky, 'In vivo evolution of policies that govern a distributed system', in *Proceedings of the 10th IEEE International Conference on Policies for Distributed Systems and Networks*, pp. 134–141, 2009.
- [21] Yoav Shoham and Moshe Tennenholtz, 'Emergent conventions in multi-agent systems: Initial experimental results and observations', in *Proceedings of the Third International Conference on the Principles of Knowledge Representation and Reasoning*, pp. 225–231. Morgan Kaufmann, 1992.
- [22] Gita Sukthankar, Robert P. Goldman, Christopher Geib, David V. Pynadath, and Hung Hai Bui, *Plan, Activity, and Intent Recognition: Theory and Practice*, Elsevier, 2014.
- [23] W.T. Luke Teacy, Michael Luck, Alex Rogers, and Nicholas R. Jennings, 'An efficient and versatile approach to trust and reputation using hierarchical Bayesian modelling', *Artificial Intelligence*, **193**, 149–185, 2012.
- [24] Wamberto Vasconcelos, Martin J. Kollingbaum, and Timothy J. Norman, 'Resolving conflict and inconsistency in norm-regulated virtual organizations', in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 632–639. ACM, 2007.
- [25] Georg Henrik von Wright, *Norm and action: a logical enquiry*, Routledge & Kegan Paul, 1963.

Subsumed Label Elimination for Maximum Satisfiability

Jeremias Berg and Paul Saikko and Matti Järvisalo¹

Abstract. We propose *subsumed label elimination* (SLE), a so-called *label-based* preprocessing technique for the Boolean optimization paradigm of maximum satisfiability (MaxSAT). We formally show that SLE is orthogonal to previously proposed SAT-based preprocessing techniques for MaxSAT in that it can simplify the underlying minimal unsatisfiable core structure of MaxSAT instances. We also formally show that SLE can considerably reduce the number of internal SAT solver calls within modern core-guided MaxSAT solvers. Empirically, we show that combining SLE with SAT-based preprocessing improves the performance of various state-of-the-art MaxSAT solvers on standard industrial weighted partial MaxSAT benchmarks.

1 INTRODUCTION

Maximum satisfiability (MaxSAT), the optimization counterpart of Boolean satisfiability (SAT), is becoming a competitive approach to solving hard optimization problems due to recent advances in MaxSAT solving [2, 38]. As MaxSAT is finding an increasing number of applications in solving real-world optimization problems—ranging from, e.g., inconsistency analysis, diagnosis, design debugging, and fault localization [15, 14, 4, 32, 44, 30, 39, 27, 35] to further applications in AI, combinatorics, data analysis, and bioinformatics [41, 23, 43, 3, 10, 21, 8, 9, 42]—there is a high demand for new techniques for speeding up MaxSAT solving further.

This paper focuses on improving the efficiency of solving real-world MaxSAT instances via preprocessing the instances before calling a state-of-the-art MaxSAT solver. In particular, effective preprocessing techniques for MaxSAT have the promise of providing *solver-independent* speed-ups to overall solving times, similarly to SAT where preprocessing is today an integral part of the solving process [20, 29]. This motivates work on MaxSAT-level preprocessing, in hope of bridging the gap between highly successful SAT preprocessing and the currently less studied and understood role of preprocessing for MaxSAT [7, 11, 31, 5, 13].

One approach to MaxSAT preprocessing is to lift commonly applied SAT preprocessing techniques, such as bounded variable elimination [20], self-subsuming resolution, and forms of clause elimination [26], to MaxSAT. Direct applications of such SAT preprocessing techniques are not correct w.r.t. preserving the optimal solutions of MaxSAT instances [7]. However, correct liftings to MaxSAT are enabled by the so-called *labelled conjunctive normal form* (LCNF) representation [7, 6].

A natural next goal for MaxSAT preprocessing is to go beyond lifting well-known SAT preprocessing techniques, by developing novel MaxSAT-specific LCNF-level preprocessing techniques that

can be applied in conjunction with SAT-based preprocessing techniques, ideally with orthogonal simplification properties. In this paper, we address this challenge by proposing *label-based preprocessing* as a form of native LCNF-level MaxSAT preprocessing. In particular, we propose the preprocessing technique of *subsumed label elimination* (SLE). The main aim of SLE is, working in conjunction with SAT-based preprocessing on labelled MaxSAT instances, to detect and eliminate *redundant labels*, i.e., auxiliary variables that are first added to maintain correctness under SAT-based preprocessing, but which can be inferred to be redundant by a simple polynomial-time deduction rule that SLE implements. Arising from deduction rules proposed in the nineties for the so-called *binate covering problem* [17, 16], a key insight of SLE is that redundant labels can be eliminated by comparing the label-sets L of clauses C^L on the LCNF level, i.e., *regardless of the contents of C* . While SLE is based on a relatively simple observation, it significantly differs from the earlier proposed SAT-based preprocessing techniques for MaxSAT. In practice it also tends to provide further speed-ups to the MaxSAT solving process for several state-of-the-art MaxSAT solvers.

In more detail, we analyze how known LCNF-lifted SAT preprocessing techniques and SLE modify key properties of MaxSAT instances: the (labelled) minimal unsatisfiable cores (LMUSes) and (labelled) minimal correction sets (LMCSes). We show that SLE is fundamentally different from LCNF-lifted SAT preprocessing. In contrast to SAT preprocessing which is unable to simplify LMUSes and LMCSes, SLE can effectively remove labels from LMUSes. Via a straightforward translation of LCNFs to standard MaxSAT, this implies that SLE can reduce the number of standard MUSes in the resulting MaxSAT instance. This can improve the performance of so-called *core-guided MaxSAT solvers*, such as [22, 25, 40, 36, 37], as well as those based on the implicit hitting set approach [18, 19, 11]. Giving a concrete witnessing family of LCNF-MaxSAT instances, we show that SLE has the potential to drastically decrease the number of iterations performed by various core-guided MaxSAT solvers. Complementing the theoretical analysis, we show empirically that by combining SLE with LCNF-lifted SAT preprocessing, noticeably more labels (i.e. redundancies) are eliminated than without SLE on weighted partial MaxSAT instances of the industrial track of MaxSAT Evaluation 2015. Further, we show that the additional simplifications translate into runtime improvements for various state-of-the-art MaxSAT solvers on industrial weighted partial instances.

This paper is organized as follows. After preliminaries on labelled CNFs and SAT-based preprocessing for MaxSAT (Section 2), we detail subsumed label elimination (Section 3), and provide a theoretical analysis of SLE both in terms of its effects on the core structure of MaxSAT instances (Section 4) and its potential to speed-up MaxSAT solving (Section 5). Empirical results on simplifications provided by SLE and the impact of SLE on the performance of MaxSAT solvers are provided in Section 6.

¹ Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Finland

2 PRELIMINARIES

Throughout this paper, we work with *labelled CNFs* (LCNFs) [7, 6] which allow for generalizing MaxSAT and provide a convenient formalism for describing correct liftings of SAT preprocessing techniques to MaxSAT. For an intuitive reading, in LCNF a set of *labels* is associated with each clause. An empty label-set denotes that the corresponding clause is hard, while a non-empty label-set implies that the corresponding clause is soft. Furthermore, key concepts such as maximum satisfiability, minimal unsatisfiable subsets and minimal correction sets, are defined over the *label-sets* L of LCNF clauses C^L instead of the clauses C .

Before the formal definitions, consider the MaxSAT instance with three unweighted soft clauses shown in Figure 1 (1). As argued in [7], in order to apply e.g. bounded variable elimination (VE) [20] and still maintain the set of optimal solutions, each soft clause C_i needs to be attached an auxiliary fresh variable \mathbf{l}_i , resulting in the instance (Figure 1, 2a). On the level of LCNFs [6], the resulting instance is shown in Figure 1 (2b). Restricting VE from eliminating any of the added variables allows for sound application of most SAT preprocessing techniques in terms of MaxSAT. As an example, first eliminating the variable x and then y gives (possibly among others; here \bowtie_x denotes resolving on x) the clause shown in Figure 1 (3a). Notice how the original one-to-one mapping between the clauses and labels vanishes, as after VE a clause may contain multiple labels. To solve the MaxSAT instance after preprocessing, the clauses obtained by preprocessing are then considered hard, and for each \mathbf{l}_i the unit soft clause $(\neg \mathbf{l}_i)$ with weight inherited from C_i is added into the instance. On the LCNF level, *labelled* VE [7] results equivalently in the LCNF instance (3b), explicitly separating original variables and the labels in each of the clauses.

(1) MaxSAT instance:

$$C_1 = (x \vee y \vee z), C_2 = (\neg x \vee \neg a \vee y), C_3 = (\neg y \vee \neg a \vee \neg b)$$

(2a) After adding labels: $C_1 = (x \vee y \vee z \vee \mathbf{l}_1)$ $C_2 = (\neg x \vee \neg a \vee y \vee \mathbf{l}_2)$ $C_3 = (\neg y \vee \neg a \vee \neg b \vee \mathbf{l}_3)$...	(3a) After variable eliminating x and y : $((C_1 \bowtie_x C_2) \bowtie_y C_3)$ $= (\neg a \vee \neg b \vee z \vee \mathbf{l}_1 \vee \mathbf{l}_2 \vee \mathbf{l}_3)$...
(2b) LCNF representation: $C_1^{\{\mathbf{l}_1\}}$ $C_2^{\{\mathbf{l}_2\}}$ $C_3^{\{\mathbf{l}_3\}}$...	(3b) After labelled variable elimination on x and y : $((C_1^{\{\mathbf{l}_1\}} \bowtie_x C_2^{\{\mathbf{l}_2\}}) \bowtie_y C_3^{\{\mathbf{l}_3\}})$ $= (\neg a \vee \neg b \vee z)^{\{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3\}}$...

Figure 1: Example of SAT-based preprocessing on the CNF and LCNF level.

2.1 Labelled CNFs and MaxSAT

Assume a countable set Lbl of labels. A labelled clause C^L consists of a clause C and a (possibly empty) set $L \subseteq Lbl$ of labels. A LCNF formula Φ is a set of labelled clauses. $Cl(\Phi)$ and $Lbls(\Phi)$ denote the set of clauses and labels of Φ , respectively, and $LCl(\Phi, l) = \{C^L \mid C^L \in \Phi, l \in L\}$ the set of labelled clauses in Φ that have l in their label-set. A LCNF formula is satisfiable iff $Cl(\Phi)$ (a CNF formula) is satisfiable.

Given a LCNF formula Φ and a subset $M \subseteq Lbls(\Phi)$ of its labels, the subformula $\Phi|_M$ of Φ induced by M is $\{C^L \in \Phi \mid L \subseteq M\}$, i.e., the LCNF formula obtained by removing from Φ all labelled

clauses with at least one label not in M ; notice that $\Phi|_{Lbls(\Phi) \setminus M} = \{C^L \in \Phi \mid L \cap M = \emptyset\}$. The *removal* $\text{REMOVE}(\Phi, K)$ of the label-set $K \subseteq Lbls(\Phi)$ from Φ gives $\{C^{L \setminus K} \mid C^L \in \Phi\}$, i.e. the LCNF formula obtained by removing all labels from Φ that are in K (note that removal does not remove clauses).

A (labelled) *unsatisfiable core* of an unsatisfiable LCNF formula Φ is a label-set $L \subseteq Lbls(\Phi)$ such that $\Phi|_L$ is unsatisfiable. An unsatisfiable core L is minimal (a LMUS) iff $\Phi|_{L'}$ is satisfiable for all $L' \subset L$. We denote the set of minimal unsatisfiable cores of Φ by $\text{LMUS}(\Phi)$. A (labelled) *minimal correction subset* (LMCS) of Φ is a label-set $R \subseteq Lbls(\Phi)$ such that (i) $\Phi|_{Lbls(\Phi) \setminus R}$ is satisfiable, and (ii) $\Phi|_{Lbls(\Phi) \setminus R'}$ is unsatisfiable for all $R' \subset R$. We denote the set of LMCSes of Φ by $\text{LMCS}(\Phi)$. Hitting set duality, formalizing a connection between LMUSes and LMCSes, is useful in this work.

Theorem 1 (Hitting set duality [6]) *A label-set $R \subseteq Lbls(\Phi)$ of a LCNF formula Φ is a LMCS of Φ iff R is an irreducible hitting set over $\text{LMUS}(\Phi)$, i.e., iff R is a hitting set over $\text{LMUS}(\Phi)$ and no $R' \subset R$ is a hitting set of $\text{LMUS}(\Phi)$.*

A LCNF-MaxSAT instance consists of a LCNF formula Φ , and a weight function $w: Lbls(\Phi) \rightarrow \mathbb{N}$ assigning a positive weight $w(l)$ to each label $l \in Lbls(\Phi)$. The cost of a label-set $L \subseteq Lbls(\Phi)$ is the sum of the weights of the labels in L . Given a LCNF-MaxSAT instance Φ such that $\Phi|_\emptyset$ is satisfiable, any assignment τ that satisfies $\Phi|_\emptyset$ is a solution to the LCNF-MaxSAT instance. A solution τ is optimal if it satisfies $\Phi|_{Lbls(\Phi) \setminus R}$ for some minimum-cost LMCS R of Φ . The cost of τ is the cost of R . We treat the MaxSAT problem for LCNFs as the problem of computing R . In the rest of the text we will always assume that solutions to (Φ, w) exist, i.e., that $\Phi|_\emptyset$ is satisfiable.

A (standard/non-labelled) MaxSAT instance $F = (F_h, F_s, w)$ consists of a set F_h of hard and a set F_s of soft clauses, together with a function $w: F_s \rightarrow \mathbb{N}$ assigning a positive weight $w(C)$ to each soft clause $C \in F_s$. A (standard) minimal correction set (MCS) of F is a subset-minimal subset of F_s whose removal from F_s makes the instance satisfiable. Similarly, a (standard) minimal unsatisfiable core (MUS) of F is a subset-minimal subset F'_s for which $F_h \cup F'_s$ is an unsatisfiable set of clauses. Given a non-labelled MaxSAT instance F , any truth assignment τ satisfying all hard clauses is a solution to the instance. A solution τ is optimal if the sum of the weights of the soft clauses τ satisfies is the maximum over all solutions. Notice that the soft clauses falsified by an optimal solution form a minimum-cost MCS of F .

A MaxSAT instance $F = (F_h, F_s, w)$ can be viewed as a LCNF-MaxSAT instance (Φ_F, w) by introducing (i) for each hard clause $C \in F_h$ the labelled clause C^\emptyset , and (ii) for each soft clause $C \in F_s$ the labelled clause $C^{\{l_C\}}$, where l_C is a distinct label for C with weight $w(l_C) = w(C)$. It is easy to see that any optimal solution to Φ_F is an optimal solution to F , and vice versa. An essential intuition is that LMCSes of (Φ_F, w) correspond exactly to the MCSes of (F_h, F_s, w) in that for any MCS $\{C_1, \dots, C_k\}$ there is a corresponding LMCS $\{l_{C_1}, \dots, l_{C_k}\}$ (and vice versa). Similarly, LMUSes of (Φ_F, w) correspond to MUSes of (F_h, F_s, w) .

To the other direction, a LCNF-MaxSAT instance (Φ, w) can be viewed as a MaxSAT instance F_Φ by associating with each label $l_i \in Lbls(\Phi)$ a distinct variable a_i , and introducing (i) for each labelled clause $C^L \in \Phi$ a hard clause $C \vee \bigvee_{l_i \in L} a_i$, and (ii) for each $l_i \in Lbls(\Phi)$, a soft clause $(\neg a_i)$ with weight $w((\neg a_i)) = w(l_i)$, where $w(l_i)$ is the weight of the label l_i . Again, using this reduction, LMUSes and LMCSes of (Φ, w) correspond exactly to the MUSes

and MCSes of F_Φ . Importantly for this work, especially the discussion in Section 5, this reduction allows one to treat any standard MaxSAT solver as a LCNF-MaxSAT solver.

Example 2 Consider the MaxSAT instance $F_{\text{ex}} = (F_h, F_s, w)$ with $w(C) = 1$ for all $C \in F_s$, $F_h = \{(x \vee y), (\neg t \vee \neg z), (\neg z \vee y), (\neg y \vee z), (z \vee t)\}$, and $F_s = \{(\neg x), (x), (y \vee t), (z \vee t \vee x)\}$. The assignment τ for which $\tau(t) = \tau(x) = 0$ and $\tau(y) = \tau(z) = 1$ is an optimal solution to F_{ex} with cost 1. The LCNF-MaxSAT instance $\Phi_{F_{\text{ex}}}$ corresponding to F_{ex} is

$$\Phi_{F_{\text{ex}}} = \{(x \vee y)^\emptyset, (\neg t \vee \neg z)^\emptyset, (\neg z \vee y)^\emptyset, (\neg y \vee z)^\emptyset, (z \vee t)^\emptyset, (\neg x)^{\{l_1\}}, (x)^{\{l_2\}}, (y \vee t)^{\{l_3\}}, (z \vee t \vee x)^{\{l_4\}}\}$$

with $w(l_i) = 1$ for $i = 1..4$. Now $Cl(\Phi_{F_{\text{ex}}}) = F_h \cup F_s$ and $Lbls(\Phi_{F_{\text{ex}}}) = \{l_1, l_2, l_3, l_4\}$. The label-set $L = \{l_1, l_2\}$ is an LMUS of $\Phi_{F_{\text{ex}}}$ as

$$\Phi_{F_{\text{ex}}|_L} = \{(x \vee y)^\emptyset, (\neg t \vee \neg z)^\emptyset, (\neg z \vee y)^\emptyset, (\neg y \vee z)^\emptyset, (z \vee t)^\emptyset, (\neg x)^{\{l_1\}}, (x)^{\{l_2\}}\}$$

is unsatisfiable. The sets $R_1 = \{l_1\}$ and $R_2 = \{l_2\}$ are examples of (minimum-cost) LMCSes of $\Phi_{F_{\text{ex}}}$. The fact that τ is an optimal solution to the LCNF-MaxSAT instance $\Phi_{F_{\text{ex}}}$ can be verified by checking that τ satisfies $\Phi_{F_{\text{ex}}|_{Lbls(\Phi_{F_{\text{ex}}}) \setminus R_2}}$. Converting $\Phi_{F_{\text{ex}}}$ back to MaxSAT results in the instance $F' = (F'_h, F'_s, w)$ with

$$F'_h = \{(x \vee y), (\neg t \vee \neg z), (\neg z \vee y), (\neg y \vee z), (z \vee t), (\neg x \vee a_1), (x \vee a_2), (y \vee t \vee a_3), (z \vee t \vee x \vee a_4)\}$$

and $F'_s = \{(\neg a_1), (\neg a_2), (\neg a_3), (\neg a_4)\}$.

2.2 SAT-based Preprocessing for LCNFs

A motivation for viewing MaxSAT instances as LCNF in [7] was to develop sound applications of SAT preprocessing techniques for MaxSAT. Many important SAT preprocessing techniques, including bounded variable elimination (VE) [20], self-subsuming resolution (SSR), and subsumption elimination (SE), cannot be used directly on MaxSAT instances [7]. However, the techniques can be applied on LCNFs by taking into account the natural restrictions implied by the SAT-level techniques on the label-sets of labelled clauses. With this intuition, the following LCNF-liftings of VE, SSR, and SE were proposed [7].

- **LCNF-lifting of the resolution rule:** The resolvent of two labelled clauses $(x \vee A)^{L_1}$ and $(\neg x \vee B)^{L_2}$ w.r.t. x is $(x \vee A)^{L_1} \boxtimes_x (\neg x \vee B)^{L_2} = (A \vee B)^{L_1 \cup L_2}$.
- **LCNF-lifting of VE (LVE):** Let Φ_x and $\Phi_{\neg x}$, resp., denote the sets of labelled clauses that contain the literal x and the literal $\neg x$, resp. LVE allows for replacing $\Phi_x \cup \Phi_{\neg x}$ with $\Phi_x \boxtimes_x \Phi_{\neg x} = \{A^{L_1} \boxtimes_x B^{L_2} \mid A^{L_1} \in \Phi_x, B^{L_2} \in \Phi_{\neg x}, A \vee B \text{ non-tautological}\}$ given that $|\Phi_x \boxtimes_x \Phi_{\neg x}| \leq |\Phi_x \cup \Phi_{\neg x}|$.
- **LCNF-lifting of SE (LSE):** A labelled clause A^{L_1} subsumes B^{L_2} if $A \subseteq B$ and $L_1 \subseteq L_2$. LSE allows for removing subsumed clauses.
- **LCNF-lifting of SSR (LSSR):** Given labelled clauses $(l \vee A)^{L_1}$ and $(\neg l \vee B)^{L_2}$, if A^{L_1} subsumes B^{L_2} , LSSR allows for replacing $(\neg l \vee B)^{L_2}$ with B^{L_2} .

Blocked clause elimination (BCE) [28] is sound for MaxSAT [7], and could as such be directly applied on MaxSAT instances. However, for a uniform presentation, it makes sense to consider a straightforward lifting of BCE.

- **LCNF-lifting of BCE (LBCE):** A labelled clause C^L is blocked in Φ if C is blocked in $Cl(\Phi)$. LBCE allows for removing blocked clauses.

Example 3 Consider the LCNF-MaxSAT instance $\Phi_{F_{\text{ex}}}$ from Example 2. Applying LSE to remove $(z \vee t \vee x)^{\{l_4\}}$ and LVE to eliminate x and t results in the formula

$$\{(y)^{\{l_1\}}, (\neg z \vee y)^\emptyset, (\neg y \vee z)^\emptyset, ()^{\{l_1, l_2\}}, (y \vee \neg z)^{\{l_3\}}\}.$$

Removing $(y \vee \neg z)^{\{l_3\}}$ by LSE and eliminating z by LVE results in the preprocessed formula $\Phi_{F_{\text{ex}}}^{\text{pre}} = \{(y)^{\{l_1\}}, ()^{\{l_1, l_2\}}\}$.

LVE, LSSR, LSE, and LBCE are correct due to the following.

Proposition 4 ([7]) Let Φ be a LCNF-MaxSAT instance and Φ^{pre} the LCNF-MaxSAT instance resulting from an application of LVE, LSSR, LSE, and LBCE on Φ . Then $\text{LMUS}(\Phi) = \text{LMUS}(\Phi^{\text{pre}})$ and, by Theorem 1, $\text{LMCS}(\Phi) = \text{LMCS}(\Phi^{\text{pre}})$.

3 SUBSUMED LABEL ELIMINATION

We propose and analyze *subsumed label elimination* (SLE), a label-based preprocessing technique for MaxSAT. The primary goal of SLE is to provide further simplifications when applied in conjunction with SAT-based preprocessing; SLE focuses on removing labels from non-singleton label-sets (produced starting from non-labelled MaxSAT instances mainly by LVE). Before a formal definition of SLE, we begin with an example to illustrate some of the shortcomings of SAT-based preprocessing for MaxSAT that SLE seeks to address.

Example 5 Consider the MaxSAT instance $F = (F_h, F_s, w)$ with $w(C) = 1$ for all $C \in F_s$ and

$$F_h = \{(x \vee y)\} \text{ and } F_s = \{(\neg x), (\neg y)\}.$$

Converting F to LCNF gives the instance $\Phi_F = \{(x \vee y)^\emptyset, (\neg x)^{\{l_1\}}, (\neg y)^{\{l_2\}}\}$. Applying LVE to eliminate both x and y results in the LCNF-MaxSAT instance $\text{pre}(\Phi_F) = \{()^{\{l_1, l_2\}}\}$. Finally, converting $\text{pre}(\Phi_F)$ back to MaxSAT gives the MaxSAT instance $F' = (F'_h, F'_s, w)$ with

$$F'_h = \{(a_1 \vee a_2)\} \text{ and } F'_s = \{(\neg a_1), (\neg a_2)\},$$

i.e., the exact same instance as F modulo variable naming. In other words, LVE (or LSSR, LSE, and LBCE) is unable to simplify F . Furthermore, notice that F contains exactly one MUS: $\{(\neg x), (\neg y)\}$. As the clauses $(\neg x)$ and $(\neg y)$ occur in exactly the same MUSes, no optimal solution to F falsifies both of them. As an alternative view, no MCS of F contains both $(\neg x)$ and $(\neg y)$, which means that either clause could be hardened, i.e., changed to a hard clause, without removing all of the optimal solutions of the instance. As we will see, SLE captures this simplification on the LCNF-level.

More concretely, consider a LCNF-MaxSAT instance Φ . SLE is based on the following observation. Consider two labels $l_1, l_2 \in Lbls(\Phi)$ such that $w(l_1) \leq w(l_2)$, and l_1 appears in at least the same LMUSes of Φ as l_2 . Then l_2 is redundant in that l_2 can be replaced by l_1 in any LMCS R of Φ without increasing the cost of R . Hence l_2 can be removed from Φ while maintaining at least one minimum-cost LMCS. This is more formally stated as Theorem 6.

Theorem 6 Let $l_1, l_2 \in Lbls(\Phi)$ and $\Phi^{\text{pre}} = \text{REMOVE}(\Phi, \{l_2\})$. Assume that, for all $L \in \text{LMUS}(\Phi)$, $l_2 \in L$ implies $l_1 \in L$. Then $\emptyset \neq \text{LMCS}(\Phi^{\text{pre}}) \subseteq \text{LMCS}(\Phi)$.

Proof. $\Phi^{pre} \upharpoonright_{Lbbs(\Phi^{pre}) \setminus R} = \Phi \upharpoonright_{Lbbs(\Phi) \setminus R}$ for any label-set $R \subseteq Lbbs(\Phi^{pre})$. Hence it suffices to show that there is an $R \in \text{LMCS}(\Phi)$ s.t. $R \subseteq Lbbs(\Phi^{pre})$. This can be verified by viewing R as an irreducible hitting set of $\text{LMUS}(\Phi)$. If $R \not\subseteq Lbbs(\Phi^{pre})$, then $l_2 \in R$. By assumption, $R' = (R \setminus \{l_2\}) \cup \{l_1\}$, a subset of $Lbbs(\Phi^{pre})$, is also an irreducible hitting set of $\text{LMUS}(\Phi)$ and hence a LMCS of Φ . \square

While the assumption in Theorem 6 is likely not checkable in polynomial time, a stricter, easier-to-check version of the assumption, formalized in Proposition 7, gives the basis for SLE. In words, let L be any label-set and $C^{L'}$ any labelled clause of Φ . If L' contains labels l_1 and l_2 such that $l_2 \in L$ but $l_1 \notin L$, then $C^{L'}$ is not a member of the formula $\Phi|_L$. This is specifically true for any LMUS of Φ .

Proposition 7 *Let $l_1, l_2 \in Lbbs(\Phi)$ and $LCl(\Phi, l_2) \subseteq LCl(\Phi, l_1)$. Then, for all $L \in \text{LMUS}(\Phi)$, $l_2 \in L$ implies $l_1 \in L$.*

Proof. Let L be a label-set such that $l_2 \in L$ and $l_1 \notin L$. We show that L is not a LMUS of Φ . From the assumption $LCl(\Phi, l_2) \subseteq LCl(\Phi, l_1)$ it follows that, if $C^{L'}$ is a labelled clause for which $l_2 \in L'$, then $l_1 \in L'$. Thus $C^{L'} \notin \Phi|_L$, and hence $\Phi|_L = \Phi|_{L \setminus \{l_2\}}$. As such $L \notin \text{LMUS}(\Phi)$ as either $\Phi|_L$ is satisfiable or $\Phi|_{L_1}$ is unsatisfiable for $L_1 = L \setminus \{l_2\} \subset L$. \square

The final part in the formalization of SLE ensures that the removal of l_2 preserves at least one *minimum-cost* LMCS of the instance. This follows by adding an assumption on the weights of l_1 and l_2 .

Proposition 8 *Let $l_1, l_2 \in Lbbs(\Phi)$ and $\Phi^{pre} = \text{REMOVE}(\Phi, \{l_2\})$. Assume that, for all $L \in \text{LMUS}(\Phi)$, $l_2 \in L$ implies $l_1 \in L$, and $w(l_1) \leq w(l_2)$. Then all minimum-cost LMCSes of Φ^{pre} are also minimum-cost LMCSes of Φ .*

Proof. Following the proof of Theorem 6 let $R' = (R \setminus \{l_2\}) \cup \{l_1\}$ be the LMCS of Φ constructed in order to replace the LMCS $R \not\subseteq Lbbs(\Phi^{pre})$. The extra assumption on the weights guarantees that the cost of R' is not higher than the cost of R . \square

Putting these results together gives SLE. Informally, SLE removes subsumed labels l_2 , or, more formally, converts Φ into $\text{REMOVE}(\Phi, \{l_2\})$.

Definition 9 (Subsumed Label Elimination (SLE)) *Let Φ be a LCNF-MaxSAT instance and $l_1, l_2 \in Lbbs(\Phi)$. We say that l_1 subsumes l_2 if (i) $LCl(\Phi, l_2) \subseteq LCl(\Phi, l_1)$, and (ii) $w(l_1) \leq w(l_2)$. SLE allows for removing subsumed labels from LCNF-MaxSAT instances.*

Example 10 *Consider the LCNF-MaxSAT instance*

$$\begin{aligned} \Phi = & \{(x_i \vee y_j)^\emptyset \mid i, j = 1..4\} \cup \\ & \{(\neg x_i \vee \neg x_3)^\emptyset, (\neg x_i \vee \neg x_4)^\emptyset \mid i = 1, 2\} \cup \\ & \{(\neg y_i)^{\{l_i, t_i\}}, (\neg y_i)^{\{l_i, t_i\}} \mid i = 1..4\} \end{aligned}$$

with $w(l) = 1$ and $w(l_i) = w(t_i) = 2$ for all i . First note that LVE, LSSR, LSE, and LBCE cannot simplify Φ . Specifically, as every variable appears both negatively and positively at least twice and no produced resolvents are tautologies, LVE cannot eliminate any variables. However, l subsumes all of the other labels, and hence applying SLE gives

$$\begin{aligned} & \{(x_i \vee y_j)^\emptyset \mid i, j = 1..4\} \cup \\ & \{(\neg x_i \vee \neg x_3)^\emptyset, (\neg x_i \vee \neg x_4)^\emptyset \mid i = 1, 2\} \cup \\ & \{(\neg y_i)^{\{l_i\}} \mid i = 1..4\}. \end{aligned}$$

Each y_i appears negatively only in a single clause and can hence be eliminated by LVE, resulting in

$$\{(x_i)^{\{l_i\}} \mid i = 1..4\} \cup \{(\neg x_i \vee \neg x_3)^\emptyset, (\neg x_i \vee \neg x_4)^\emptyset \mid i = 1, 2\}.$$

Now each x_i only appears positively in a single clause. LVE then gives $\Phi^{pre} = \{()\}^{\{l_i\}}$.

Remark 1 *While the main focus of this work is on understanding the effect of SLE on the core structure of MaxSAT instances and the potential of SLE to speed up state-of-the-art MaxSAT solvers, we note that SLE (for MaxSAT) can be viewed as the counterpart of the so-called dominance rule proposed in the early 90s in conjunction with branch-and-bound approaches for the so-called binate covering problem [17, 16] with applications in logic synthesis. More details on this connection are provided in Appendix A. To the best of our knowledge, however, SLE has not been previously proposed, analyzed, or empirically evaluated in the context of MaxSAT.*

4 EFFECTS OF SLE

We continue by analyzing SLE in terms of how it simplifies LCNFs. We show that SLE is orthogonal to the LCNF-lifted SAT-based preprocessing techniques in terms of the LMUSes and LMCSes—and hence MaxSAT solutions—preserved under simplification.

We start with relatively simple corollaries of the definition. First, we observe that subsumed labels remain subsumed after applications of SAT-based preprocessing.

Proposition 11 *Let $l \in Lbbs(\Phi)$ and assume that SLE can eliminate l from Φ . Let Φ^{pre} be Φ after applying LVE, LSSR, LSE, or LBCE. Then SLE can eliminate l from Φ^{pre} .*

Proof. Let l_1 be a label that subsumes l in Φ . It suffices to show that the preconditions of SLE are satisfied in Φ^{pre} . First, the precondition $w(l_1) \leq w(l)$ is trivially satisfied as none of the techniques alter the weights of labels. For the second precondition, $LCl(\Phi^{pre}, l) \subseteq LCl(\Phi^{pre}, l_1)$, the non-trivial case is $LCl(\Phi^{pre}, l) \neq \emptyset$. As $LCl(\Phi, l) \subseteq LCl(\Phi, l_1)$, it is enough to verify that none of the SAT-based preprocessing techniques introduce a labelled clause $C^L \in \Phi^{pre}$ with $l \in L$ and $l_1 \notin L$. This is trivially true for LSE and LBCE as they only remove clauses. This is also true for LSSR as it only removes literals, not labels. Finally, LVE cannot produce resolvents which contain l but not l_1 , since there are no labelled clauses $C^{L'}$ in Φ with $l \in L'$ and $l_1 \notin L'$. Thus the label-set of any resolvent produced by LVE, which is a union of label-sets in Φ , contains either both or neither of l_1 and l . \square

Thus it makes sense to incorporate SLE into the preprocessing loop together with LVE, LSSR, LSE, and LBCE.

In analogy with Proposition 11, subsumed labels remain subsumed also under SLE steps quite generally. An exception comes from cases in which two labels l_1 and l_2 subsume each other, i.e., when l_1 and l_2 occur in exactly the same label-sets and $w(l_2) = w(l_1)$. Note also that, generally, if l_1 subsumes l_2 , and l_2 subsumes l_3 , then l_1 subsumes l_3 .

Turning to comparing SLE and SAT-based preprocessing, Propositions 4 and 12 together illustrate fundamental differences between SLE and LVE, LSSR, LSE, and LBCE. By Proposition 4, LVE, LSSR, LSE, and LBCE preserve the LMUSes of LCNF-MaxSAT instances. This is not true for SLE. Instead, SLE guarantees (only) that at least one minimum-cost (optimal) LMCS and, as such, that at least one optimal solution of the instance is preserved.

Proposition 12 *SLE does not in general preserve LMUSes (or LMCSes) of LCNF-MaxSAT instances.*

Proof. Consider the instances Φ and Φ^{pre} from Example 10. The sets $\{l, l_i\}$ and $\{l, t_i\}$ are LMUSes of Φ for all i but not of Φ^{pre} . \square

An alternative way of stating Proposition 12 is that applying SLE does not in general preserve all optimal solutions to LCNF-MaxSAT instances. For a simple example, consider the LCNF-MaxSAT instance $\Phi = \{(x)^{\{l_1\}}, (\neg x)^{\{l_2\}}\}$ with unit-weighted labels. There are two optimal solutions to Φ : $\tau_1(x) = 1$ satisfying $\Phi|_{Lbbs(\Phi) \setminus \{l_2\}}$, and $\tau_2(x) = 0$ satisfying $\Phi|_{Lbbs(\Phi) \setminus \{l_1\}}$. However, by LVE we can simplify Φ to $\{()\}^{\{l_1, l_2\}}$ and by SLE further to $\{()\}^{\{l_1\}}$. The only LMCS of the simplified instance is $\{l_1\}$, corresponding to the solution τ_2 .

Instead of preserving LMUSes, SLE could be seen as a form of LMUS minimization in the sense that all LMUSes remaining after SLE are projections of LMUSes of the original LCNF onto the remaining set of labels.

Theorem 13 *Let Φ be a LCNF-MaxSAT instance and $l \in Lbbs(\Phi)$ a subsumed label. Let $\Phi^{pre} = \text{REMOVE}(\Phi, \{l\})$, i.e., the formula after eliminating l by SLE from Φ . Then all LMUSes L^p of Φ^{pre} are of the form $L^p = L \cap Lbbs(\Phi^{pre})$ for some LMUS L of Φ .*

Proof. First notice that $\Phi|_{L^p} \subseteq \Phi^{pre}|_{L^p}$ as the restriction operator only removes labels from label-sets, not clauses. If $\Phi|_{L^p} = \Phi^{pre}|_{L^p}$, then the same will be true for any $L_s^p \subseteq L^p$, so L^p itself is an LMUS of Φ . Otherwise, the reason for a labelled clause C^L to be in $\Phi^{pre}|_{L^p}$ but not in $\Phi|_{L^p}$ is that the eliminated label l was in L , i.e., $C^L \notin \Phi$ but $C^{L \cup \{l\}} \in \Phi$. Hence $\Phi|_{L^p \cup \{l\}} = \Phi^{pre}|_{L^p}$, and $L^p \cup \{l\}$ is a LMUS of Φ . \square

For further differences between SLE and LVE, LSSR, LSE, and LBCE, consider a MaxSAT instance F and a soft clause $C \in F_s$. Let Φ_F be the LCNF-MaxSAT instance corresponding to F and l_C the label for which $C^{\{l_C\}} \in \Phi_F$. A simple application of Theorem 4 gives that if l_C is removed from Φ_F by LVE, LSSR, LSE, or LBCE, then any optimal solution to Φ_F , which is also an optimal solution to F , will satisfy C .

Proposition 14 *Let Φ_F^{pre} be the instance resulting after an application of LVE, LSSR, LSE, or LBCE on Φ_F . If $l_C \notin Lbbs(\Phi_F^{pre})$, then any optimal solution τ to Φ_F , which is also an optimal solution to F , will satisfy C .*

Proof. Since τ is optimal, it satisfies $\Phi_F|_{Lbbs(\Phi_F) \setminus R}$ for some minimum-cost LMCS R of Φ_F . By Theorem 4, $l_C \notin R$, and thus $C \in Cl(\Phi_F|_{Lbbs(\Phi_F) \setminus R})$. \square

Informally, it could be said that SAT-based preprocessing can only remove labels that are “uninteresting” in terms of LMCS computation. In contrast, elimination of l_C by SLE means that some (but not necessarily all) optimal solutions of F satisfy C , as shown next.

Proposition 15 *Let Φ_F^{pre} be the instance resulting from an application of SLE on Φ_F . If $l_C \notin Lbbs(\Phi_F^{pre})$, then there is an optimal solution τ to Φ_F and F that satisfies C . Furthermore, there may exist optimal solutions to Φ_F that do not satisfy C .*

Proof. By the assumption that l_C is subsumed, it follows from Theorem 6 and Proposition 8 that there is a minimum-cost LMCS R of Φ_F for which $l_C \notin R$. The first part of the claim follows by observing that $\Phi_F|_{Lbbs(\Phi_F) \setminus R}$ is satisfiable and $C \in Cl(\Phi_F|_{Lbbs(\Phi_F) \setminus R})$. For the second part of the claim, consider the discussion following Proposition 12. \square

5 SLE AND CORE-GUIDED SOLVERS

We now show that SLE has the potential to considerably lower the number of iterations made by so-called *core-guided MaxSAT solvers*, one of the most successful current MaxSAT solving approaches. The core-guided approach has several variants, e.g. [2, 38, 22, 25, 40, 36, 37, 18, 19]. In this work, we study the effect of SLE on two different types of core-guided solvers through generic abstractions. The first one, *CG-MaxSAT* (Algorithm 1), iteratively employs a SAT solver to extract unsatisfiable cores and rules out each of the found cores from the formula by a *clause replication and relaxation* step. Several algorithms that fit the CG-MaxSAT abstraction have been proposed [22, 25, 40, 36, 37]. The second one, MaxHS (Algorithm 2), is an abstraction of the implicit hitting set approach to MaxSAT [18, 19], iteratively using a SAT solver to extract unsatisfiable cores, and an exact minimum-cost hitting set algorithm to compute hitting sets over the found cores.

In more detail, at each iteration i , CG-MaxSAT invokes a SAT solver on the clauses of a working formula F_w^i (initialized as all clauses of the MaxSAT instance viewed as hard). If the working formula is satisfiable, CG-MaxSAT terminates and returns the satisfying assignment returned by the SAT solver. Otherwise, the SAT solver returns an unsatisfiable core κ of F_w^i . CG-MaxSAT then duplicates the clauses in κ to create two sets κ^r and $\kappa^{\bar{r}}$. Both sets contain exactly the same clauses as κ ; each clause $C \in \kappa$ is duplicated into two: $C^r \in \kappa^r$ and $C^{\bar{r}} \in \kappa^{\bar{r}}$. The weight of C^r is set to w_m , the minimum weight over the clauses in the core, and the weight of $C^{\bar{r}}$ to $w(C) - w_m$. The clauses of $\kappa^{\bar{r}}$ are added to the working formula unaltered. Finally, the working formula is updated by *relaxing* the clauses in κ^r . The method of relaxation varies between core-guided solvers. For our analysis, the important consequences of relaxation are that the (possibly altered) clauses of κ^r do not appear as a core in future iterations, and that the optimal cost of F_w^{i+1} (when viewed as a MaxSAT instance) is exactly w_m lower than the optimal cost of F_w^i . Termination of CG-MaxSAT is guaranteed by the fact that $w_m > 0$ on all iterations and that a MaxSAT instance of cost 0 is satisfiable as a SAT instance. For a concrete example of a relaxation step, consider the classical Fu-Malik algorithm [22] and its extensions to the weighted case [33, 1]. These algorithms augments each $C_i \in \kappa^r$ with a fresh relaxation variable r_i , creating the clause $C_i \vee r_i$, and additionally adds a hard *exactly-one* constraint $\sum r_i = 1$ over the relaxation variables. The intuition behind this step is that assigning a relaxation variable to 1 effectively removes the corresponding clause from the formula, hence removing the core κ^r . Additionally,

Input: MaxSAT instance $F = (F_h, F_s, w)$.

Output: An optimal solution τ for F .

$F_w^0 \leftarrow F_h \cup F_s$

for $i=0, \dots$ **do**

$(result, \kappa, \tau) \leftarrow \text{SAT SOLVE}(F_w^i)$

if $result = \text{“satisfiable”}$ **then**

 | return τ

 // optimal solution

else

$F_w^i = (F_w^i \setminus \kappa)$ // SAT solver returned unsat core

$w_m \leftarrow \min\{w(C) \mid C \in \kappa\}$

$(\kappa^r, \kappa^{\bar{r}}) \leftarrow \text{CLAU SEREPLICATE}(\kappa, w_m)$

$F_w^i \leftarrow F_w^i \cup \kappa^{\bar{r}}$

$F_w^{i+1} \leftarrow \text{RELAX}(F_w^i, \kappa^r)$

end

end

Algorithm 1: CG-MaxSAT

Input: MaxSAT instance $F = (F_h, F_s, w)$.

Output: An optimal solution τ for F .

```

 $\mathcal{K} \leftarrow \emptyset$  // set of found unsat cores of  $F$ 
 $F_w \leftarrow (F_h \cup F_s)$ 
while true do
   $H \leftarrow \text{MINCOSTHITTINGSET}(\mathcal{K})$ 
   $F_w \leftarrow F_h \cup (F_s \setminus H)$ 
   $(\text{result}, \kappa, \tau) \leftarrow \text{SATSOLVE}(F_w)$ 
  if result = "satisfiable" then
    | return  $\tau$  // optimal solution
  else
    |  $\mathcal{K} \leftarrow \mathcal{K} \cup \{\kappa\}$  // SAT solver returned unsat core
  end
end

```

Algorithm 2: MaxHS

the exactly-one constraint ensures that the cost is lowered exactly by w_m .

MaxHS is a hybrid algorithm that uses a SAT solver for core extraction over a working formula F_w (initialized as all clauses of the input instance viewed as hard). Given a collection \mathcal{K} of extracted cores, MaxHS uses an exact algorithm (integer programming solver in practice) to find a minimum-cost hitting set hs over \mathcal{K} . The working formula is then updated to contain all clauses of F except for the soft clauses in hs , and the SAT solver is invoked again. If the working formula is satisfiable, the satisfying assignment obtained is an optimal solution to F . Otherwise another core is obtained and the search continues again with hitting set computation.

The main result of this section is that there are families of LCNF-MaxSAT instances on which SLE can significantly decrease the number of SAT solver calls and clause replication when subsequently solving the instances with CG-MaxSAT or MaxHS.

Proposition 16 For $\mathcal{A} \in \{\text{CG-MaxSAT}, \text{MaxHS}\}$, there is a family of LCNF-MaxSAT instances Φ_N , with $\Theta(N)$ different labels, on which

- (i) \mathcal{A} requires $\Theta(N)$ calls to its SAT solver, and, for $\mathcal{A} = \text{CG-MaxSAT}$, \mathcal{A} requires $\Theta(N)$ clause replication steps, on $\Theta(N!)$ different executions; while
- (ii) \mathcal{A} is guaranteed to require only two (one unsatisfiable and one satisfiable) SAT solver calls if SLE is applied on Φ_N before \mathcal{A}

under the assumption that the internal SAT solver is guaranteed to return minimal unsatisfiable cores.

Proof. The family of LCNF-MaxSAT instances witnessing the claim is the same for CG-MaxSAT and MaxHS. Let N be sufficiently large and define

$$\Phi_N := \bigcup_{i=1}^{2N-2} \mathbf{P}_i \cup \bigcup_{i=1}^{N-1} \mathbf{H}_i, \text{ where}$$

$$\mathbf{P}_i = \bigcup_{j=1}^N \{(\neg p_i^j \vee \neg p_k^j)^\emptyset \mid k = (i+1)..(2N-1)\} \text{ and}$$

$$\mathbf{H}_i = \left\{ \left(\bigvee_{j=1}^N p_k^j \right)^{\{l, l_i\}} \mid k = i..(N+i) \right\},$$

with $w(l) = w(l_{N-1}) = N$ and $w(l_i) = 1$ for all other labels l_i . Notice that Φ_N contains $N-1$ LMUSes of the form $\{l, l_i\}$ for all $1 \leq i \leq N-1$. Hence, the only minimum-cost LMCS of Φ_N is

$\{l\}$. Furthermore, refuting any of the LMUSes requires proving the unsatisfiability of the formula $\Phi_N|_{\{l, l_i\}}$, which corresponds to an instance of the pigeonhole principle; meaning that the extraction any of the LMUSes of Φ_N requires an exponentially long SAT solver call [24]. Next we sketch the executions of both CG-MaxSAT and MaxHS that require $\Theta(N)$ SAT-solver calls when solving Φ_N .

Conversion of Φ_N to MaxSAT results in the formula $F = (F_h, F_s, w)$, where

$$F_h = \bigcup_{i=1}^{2N-2} \bigcup_{j=1}^N \{(\neg p_i^j \vee \neg p_k^j) \mid k = i..(2N-1)\} \\ \cup \bigcup_{i=1}^{N-1} \left\{ \left(a_l \vee a_i \vee \bigvee_{j=1}^N p_k^j \right) \mid k = i..(N+i) \right\}$$

and $F_s = \{(\neg a_l), (\neg a_1), \dots, (\neg a_{N-1})\}$

with $w((\neg a_l)) = w((\neg a_{N-1})) = N$ and $w(C) = 1$ for all other $C \in F_s$. The MUSes of F correspond exactly to the LMUSes of Φ_N and are of the form $\{(\neg a_l), (\neg a_i)\}$ for all $i = 1..N-1$. For an intuition on the executions requiring a linear number of SAT solver calls of both algorithms, notice that both can terminate immediately and only after encountering and processing the MUS $\{(\neg a_l), (\neg a_{N-1})\}$ corresponding to the the LMUS $\{l, l_{N-1}\}$.

For $\mathcal{A} = \text{MaxHS}$, assume that the internal SAT solver returns the MUSes of in any order with $\{(\neg a_l), (\neg a_{N-1})\}$ last. Then the hitting set hs computed by MaxHS will not contain the clause $(\neg a_l)$ before the $(N-1)$ th iteration and as such MaxHS can not terminate as $F \setminus hs$ will always contain the MUS $\{(\neg a_l), (\neg a_{N-1})\}$. There are a total of $(N-2)!$ executions in which the MUS $\{(\neg a_l), (\neg a_{N-1})\}$ is returned last.

For $\mathcal{A} = \text{CG-MaxSAT}$, the long executions are similar. Assume that the first MUS returned by the SAT-solver in CG-MaxSAT is $\{(\neg a_l), (\neg a_1)\}$. The smallest weight w_m of the clauses in the core is 1, so CG-MaxSAT proceeds by replicating the clause $(\neg a_l)$ into two clauses $C^r = (\neg a_l)$ and $C_2 = (\neg a_l)$, setting $w(C^r) = 1$ and $w(C_2) = N-1$, adding C_2 back into the working formula, relaxing the core $\{C^r, (\neg a_1)\}$, and reiterating. Assume that CG-MaxSAT proceeds similarly by processing the cores $\{(\neg a_l)^i, (\neg a_i)\}$ for $i = 1..N-2$ during the first $N-2$ iterations where $(\neg a_l)^i$ is the copy of the clause $(\neg a_l)$ produced in the previous iteration. Finally on the $(N-1)$ th iteration CG-MaxSAT encounters the core $\{(\neg a_l)^{N-2}, (\neg a_{N-1})\}$. At this point $w((\neg a_l)^{N-2}) = 2$ and $w((\neg a_{N-1})) = N$, so CG-MaxSAT replicates $(\neg a_{N-1})$ and relaxes the core before invoking its SAT solver one final time in order to find the current working formula satisfiable. In total, CG-MaxSAT performs N SAT solver calls and $N-1$ clause replications. A similar argument can be made for any ordering of the MUSes with $\{(\neg a_l), (\neg a_{N-1})\}$ last.

Part (ii) of the proposition follows by noting that SLE can remove l_{N-1} due to l , resulting in the formula

$$\text{pre}(\Phi_N) := \bigcup_{i=1}^{2N-2} \mathbf{P}_i \cup \bigcup_{i=1}^{N-2} \mathbf{H}_i \cup \left\{ \left(\bigvee_{j=1}^N p_k^j \right)^{\{l\}} \mid k = (N-1)..(2N-1) \right\}.$$

The only LMUS of the preprocessed formula is $\{l\}$, which is why both algorithms are guaranteed to need only a single unsatisfiable and a single satisfiable SAT-solver call, and furthermore, why CG-MaxSAT needs no clause replication steps, during solving. \square

6 EXPERIMENTS

Complementing the theoretical analysis, we evaluate the practical effects of SLE on the 2015 MaxSAT Evaluation benchmarks (<http://www.maxsat.udl.cat/15/>). We observe that SLE is beneficial especially on *industrial* weighted partial benchmark instances. When applying SLE in conjunction with the LCNF-lifted SAT-based preprocessing techniques (LVE, LSSR, LSE, LBCE), noticeably more labels can be removed than without applying SLE. Furthermore, SLE improves the overall performance of various state-of-the-art MaxSAT solvers on industrial weighted partial benchmarks.

All reported solving times include the time spent in preprocessing as well as in the actual MaxSAT solving. The experiments were run on 2.53-GHz Intel Xeon quad-core machines with 32-GB RAM under Linux. A per-instance timeout of 1800 seconds and a memory limit of 30 GB were enforced.

We implemented SLE by extending the Coprocessor 2.0 SAT preprocessor [34] in the following way. Given a MaxSAT instance as input, we convert the instance to LCNF, apply Coprocessor to preprocess the LCNF, and then convert the preprocessed LCNF back to a MaxSAT instance. LVE, LSSR, LSE, LSSR, and LBCE are realized by representing a labelled clause C^L as $C \vee \bigvee_{l_i \in L} a_i$ in Coprocessor, applying the existing implementations of VE, SSR, SE and BCE, while forbidding the elimination of any of the a_i variables corresponding to the labels.

A simple way of implementing SLE consists of explicitly checking for each label l whether or not l is subsumed. A potentially more efficient way of implementing SLE would be to track the resolvents produced by LVE and only check labels that have appeared in resolvents produced. However, as shown in Figure 3, even the simple implementation appears to be sufficient; we did not observe any significant increase in total preprocessing time (w/pre+SLE) compared to not using SLE (w/pre). We also note that SLE does not increase overall memory consumption wrt SAT-based preprocessing.

The fraction of labels (i.e. soft clauses) remaining after preprocessing with and without SLE (applying in both cases LVE, LSSR, LSE, and LBCE) is shown in Figure 2 for both unweighted and weighted partial industrial and crafted instances. SLE is effective in removing additional labels in particular on the industrial weighted partial instances. For example, for one third of the instances ($x = 0.3$), with SLE close to 80% of the labels are eliminated ($y \approx 0.2$, i.e., some 20% of the labels remain afterwards); in comparison, without SLE only $\approx 45\%$ are eliminated. As a side-note, when examining the instance families in more detail, we found that out of the 172 industrial benchmarks in which no labels were removable by prepro-

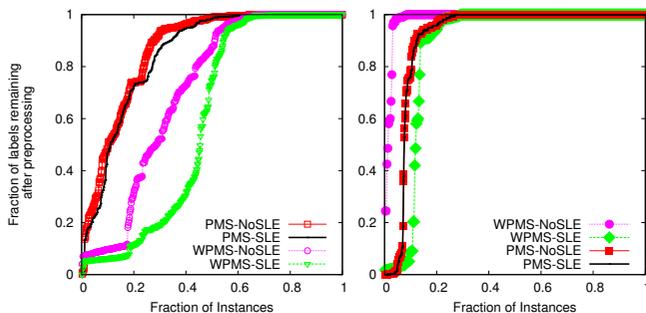


Figure 2: Fraction of labels remaining in industrial (left) and crafted (right) unweighted (PMS) and weighted (WPMS) benchmarks after preprocessing with and without SLE.

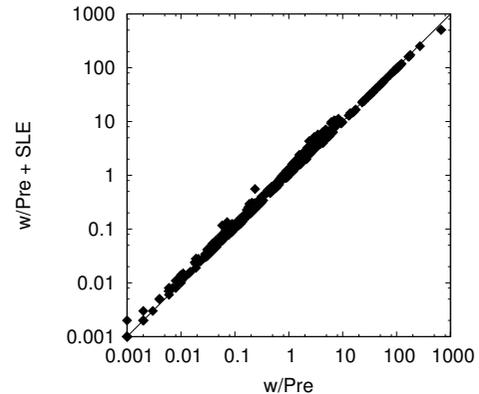


Figure 3: Influence of SLE on preprocessing time

cessing, 151 were new instances in the 2015 evaluation. In fact, when preprocessing the 2014 evaluation instances—which are a subset of the 2015 evaluation instances—using SLE, at least 80% of the labels are eliminated from over 50% of the instances. This suggests that, in terms of SLE, the instances added for 2015 are structurally different from the ones from 2014.

Table 1: Number of solved industrial weighted partial benchmarks and total time spent on solved instances without preprocessing (default), with SAT-based preprocessing (w/pre), and with both SAT-based preprocessing and SLE (w/pre+SLE).

config.	Solved instances (total running time over solved in seconds)			
	Eva	LMHS	Open-WBO	Primal-Dual
default	379 (22,543)	354 (50,981)	331 (15,762)	390 (18,423)
w/pre	384 (20,613)	368 (46,525)	369 (12,345)	391 (15,267)
w/pre+SLE	386 (19,138)	389 (48,277)	369 (11,739)	392 (13,925)

The additional simplifications obtained via SLE are also reflected in the total number of solved instances and solver runtimes on industrial weighted partial instances. Results are shown in Table 1 for the state-of-the-art MaxSAT solvers Eva [40], core-guided, best industrial weighted partial solver in 2014; LMHS [11], one of the best crafted and industrial weighted partial solvers in 2015, a labelled lifting of the SAT-IP hybrid MaxSAT solver MaxHS [18]; Open-WBO [36], one of the best industrial unweighted solvers in 2015; and Primal-Dual [12], a new core-guided solver from 2015. SAT-based preprocessing together with SLE results in the highest number of solved instances for each of the solvers. The increase in the number of solved instances is especially noticeable for LMHS. SLE also decreases the total runtime over all solved instances for each of the solvers. For example, for both Eva and Primal-Dual, using SLE improves further on applying only SAT-based preprocessing by decreasing the total runtime by approximately 10%, at the same time enabling Primal-Dual and Eva to solve one and two more instances, respectively. Finally, Figure 4 shows a comparison the running times of the individual instances with the solvers are presented in the order LMHS (first column), Eva (second), Open-WBO (third), and Primal-Dual (fourth column). For each solver, we compare runtimes on logscale when applying SLE together with LVE, LSSR, LSE, and LBCE ('w/pre+SLE') to (i) without preprocessing (left), and (ii) preprocessing only with LVE, LSSR, LSE, and LBCE ('w/pre', right). For a majority of the instances, SLE improves the total solving time of each of the solvers both compared to using no preprocessing, and only using LVE, LSSR, LSE and LBCE.

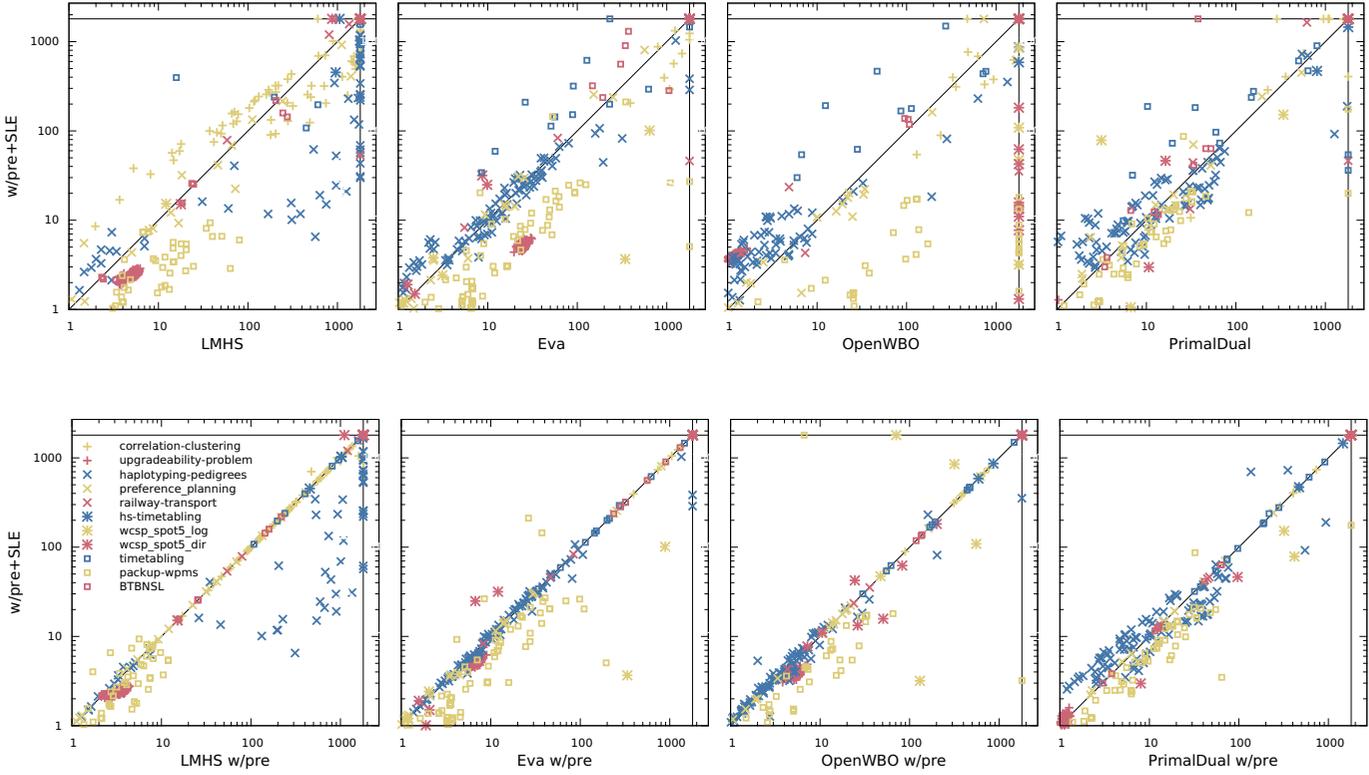


Figure 4: Effect of SLE on runtimes without (top) and with (bottom) other preprocessing on industrial weighted partial instances.

7 CONCLUSIONS

We proposed *subsumed label elimination* (SLE) as a MaxSAT preprocessing technique that is beneficial to apply in conjunction with SAT-based preprocessing techniques before MaxSAT solving. SLE is orthogonal to SAT-based preprocessing in that SLE can eliminate redundant auxiliary variables (labels) from clauses irrespective of the original variables occurring in clauses. On the level of labelled CNFs, this accounts to removing redundant labels from LMUSES, thereby resulting in cases in a decrease in the number and sizes of MUSes of MaxSAT instances. Furthermore, SLE has the potential to drastically reduce the number of iterations performed by core-guided MaxSAT solvers, currently one of the important classes of MaxSAT solvers. Applying SLE further improves the running times of various state-of-the-art MaxSAT solvers on standard industrial weighted partial benchmarks. For future work, we aim to study more general notions of redundancies over labels in LCNFs to obtain further label-based preprocessing techniques for MaxSAT, as well as to study potential applications in MUS extraction.

ACKNOWLEDGEMENTS

This work has been funded by Academy of Finland, grants 251170 COIN, 276412, and 284591; and Doctoral School in Computer Science DoCS and Research Funds of the University of Helsinki.

A SLE and Dominance in Binate Covering

SLE (for MaxSAT) can be viewed as the counterpart of the so-called *dominance rule* proposed in the early 90s in conjunction with branch-and-bound approaches for the so-called *binate covering problem* [17, 16] with applications in logic synthesis. In short, in the binate covering problem, we are given a Boolean function $f: \{0, 1\}^n \rightarrow \{0, 1\}$ over the variables x_1, \dots, x_n , and a function $cost: \{1..n\} \rightarrow \mathbb{N}$ assigning a non-negative cost $cost(i)$ to each variable x_i . The task is to find a truth assignment τ over x_1, \dots, x_n that minimizes $\sum_{i=1}^n \tau(x_i) \cdot cost(i)$ subject to $f(\tau(x_1), \dots, \tau(x_n)) = 1$. The dominance rule for binate covering is described in [17] for the so-called modified covering matrix representation of binate covering for Boolean functions in CNF. We interpret the rule directly on the definition as follows: variable x_i dominates x_j if (i) the literal x_i occurs in a clause C whenever the literal x_j occurs in C ; (ii) $\neg x_j$ occurs in a clause C whenever $\neg x_i$ occurs in C ; and (iii) $cost(x_i) \leq cost(x_j)$. A dominated variable can be assigned to 0.

A LCNF-MaxSAT instance (Φ, w) can be viewed as an instance of binate covering by viewing each labelled clause $C^L \in \Phi$ as the clause $C \vee L$, and letting $cost(l) = w(l)$ for each $l \in Lbls(\Phi)$ and $cost(x) = 0$ for each variable in $\bigcup Cl(\Phi)$. After this reduction, one can observe that, for any label $l \in Lbls(\Phi)$, it holds that l is dominated in the resulting binate covering instance if and only if SLE can eliminate l from (Φ, w) .

REFERENCES

- [1] Carlos Ansótegui, Maria Luisa Bonet, and Jordi Levy, ‘Solving (weighted) partial MaxSAT through satisfiability testing’, in *Proc. SAT*, volume 5584 of *Lecture Notes in Computer Science*, pp. 427–440. Springer, (2009).
- [2] Carlos Ansótegui, Maria Luisa Bonet, and Jordi Levy, ‘SAT-based MaxSAT algorithms’, *Artificial Intelligence*, **196**, 77–105, (2013).
- [3] Carlos Ansótegui, Idelfonso Izquierdo, Felip Manyà, and José Torres-Jiménez, ‘A Max-SAT-based approach to constructing optimal covering arrays’, in *Proc. CCIA*, volume 256 of *Frontiers in Artificial Intelligence and Applications*, pp. 51–59. IOS Press, (2013).
- [4] Josep Argelich, Daniel Le Berre, Inês Lynce, João P. Marques-Silva, and Pascal Rapicault, ‘Solving linux upgradeability problems using boolean optimization’, in *Proc. LoCoCo*, volume 29 of *EPTCS*, pp. 11–22, (2010).
- [5] Josep Argelich, Chu Min Li, and Felip Manyà, ‘A preprocessor for Max-SAT solvers’, in *Proc. SAT*, volume 4996 of *Lecture Notes in Computer Science*, pp. 15–20. Springer, (2008).
- [6] Anton Belov and Joao Marques-Silva, ‘Generalizing redundancy in propositional logic: Foundations and hitting sets duality’, *CoRR*, **abs/1207.1257**, (2012).
- [7] Anton Belov, Antonio Morgado, and Joao Marques-Silva, ‘SAT-based preprocessing for MaxSAT’, in *Proc. LPAR-19*, volume 8312 of *Lecture Notes in Computer Science*, pp. 96–111. Springer, (2013).
- [8] Jeremias Berg and Matti Järvisalo, ‘SAT-based approaches to treewidth computation: An evaluation’, in *Proc. ICTAI*, pp. 328–335. IEEE Computer Society, (2014).
- [9] Jeremias Berg and Matti Järvisalo, ‘Cost-optimal constrained correlation clustering via weighted partial maximum satisfiability’, *Artificial Intelligence*, (2015). in press.
- [10] Jeremias Berg, Matti Järvisalo, and Brandon Malone, ‘Learning optimal bounded treewidth Bayesian networks via maximum satisfiability’, in *Proc. AISTATS*, volume 33, pp. 86–95. JMLR, (2014).
- [11] Jeremias Berg, Paul Saikko, and Matti Järvisalo, ‘Improving the effectiveness of SAT-based preprocessing for MaxSAT’, in *Proc. IJCAI*, pp. 239–245. AAAI Press, (2015).
- [12] Nikolaj Bjørner and Nina Narodytska, ‘Maximum satisfiability using cores and correction sets’, in *Proc. IJCAI*, pp. 246–252. AAAI Press, (2015).
- [13] Maria Luisa Bonet, Jordi Levy, and Felip Manyà, ‘Resolution for Max-SAT’, *Artificial Intelligence*, **171**(8-9), 606–618, (2007).
- [14] Yibin Chen, Sean Safarpour, João Marques-Silva, and Andreas G. Veneris, ‘Automated design debugging with maximum satisfiability’, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **29**(11), 1804–1817, (2010).
- [15] Yibin Chen, Sean Safarpour, Andreas G. Veneris, and João P. Marques-Silva, ‘Spatial and temporal design debug using partial MaxSAT’, in *Proc. 19th ACM Great Lakes Symposium on VLSI*, pp. 345–350. ACM, (2009).
- [16] Olivier Coudert, ‘On solving covering problems’, in *Proc. DAC*, pp. 197–202. ACM Press, (1996).
- [17] Olivier Coudert and Jean Christophe Madre, ‘New ideas for solving covering problems’, in *Proc. DAC*, pp. 641–646. ACM Press, (1995).
- [18] Jessica Davies and Fahiem Bacchus, ‘Exploiting the power of MIP solvers in MaxSAT’, in *Proc. SAT*, volume 7962 of *Lecture Notes in Computer Science*, pp. 166–181. Springer, (2013).
- [19] Jessica Davies and Fahiem Bacchus, ‘Postponing optimization to speed up MAXSAT solving’, in *Proc. CP*, volume 8124 of *Lecture Notes in Computer Science*, pp. 247–262. Springer, (2013).
- [20] Niklas Eén and Armin Biere, ‘Effective preprocessing in SAT through variable and clause elimination’, in *Proc. SAT*, volume 3569 of *Lecture Notes in Computer Science*, pp. 61–75. Springer, (2005).
- [21] Zhiwen Fang, Chu-Min Li, Kan Qiao, Xu Feng, and Ke Xu, ‘Solving maximum weight clique using maximum satisfiability reasoning’, in *Proc. ECAI*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 303–308. IOS Press, (2014).
- [22] Zhaohui Fu and Sharad Malik, ‘On solving the partial MaxSAT problem’, in *Proc. SAT*, volume 4121 of *Lecture Notes in Computer Science*, pp. 252–265. Springer, (2006).
- [23] Joao Guerra and Ines Lynce, ‘Reasoning over biological networks using maximum satisfiability’, in *Proc. CP*, volume 7514 of *Lecture Notes in Computer Science*, pp. 941–956. Springer, (2012).
- [24] Armin Haken, ‘The intractability of resolution’, *Theoretical Computer Science*, **39**, 297–308, (1985).
- [25] Federico Heras, Antonio Morgado, and Joao Marques-Silva, ‘Core-guided binary search algorithms for maximum satisfiability’, in *Proc. AAAI*. AAAI Press, (2011).
- [26] Marijn Heule, Matti Järvisalo, Florian Lonsing, Martina Seidl, and Armin Biere, ‘Clause elimination for SAT and QSAT’, *Journal of Artificial Intelligence Research*, **53**, 127–168, (2015).
- [27] Alexey Ignatiev, Mikolás Janota, and João Marques-Silva, ‘Towards efficient optimization in package management systems’, in *Proc. ICSE*, pp. 745–755. ACM, (2014).
- [28] Matti Järvisalo, Armin Biere, and Marijn Heule, ‘Blocked clause elimination’, in *Proc. TACAS*, volume 6015 of *Lecture Notes in Computer Science*, pp. 129–144. Springer, (2010).
- [29] Matti Järvisalo, Marijn Heule, and Armin Biere, ‘Inprocessing rules’, in *Proc. IJCAR*, volume 7364 of *Lecture Notes in Computer Science*, pp. 355–370. Springer, (2012).
- [30] Manu Jose and Rupak Majumdar, ‘Cause clue clauses: error localization using maximum satisfiability’, in *Proc. PLDI*, pp. 437–446. ACM, (2011).
- [31] Chu Min Li, Felip Manyà, Nouredine Ould Mohamedou, and Jordi Planes, ‘Exploiting cycle structures in Max-SAT’, in *Proc. SAT*, volume 5584 of *Lecture Notes in Computer Science*, pp. 467–480. Springer, (2009).
- [32] Inês Lynce and João Marques-Silva, ‘Restoring CSP satisfiability with MaxSAT’, *Fundam. Inform.*, **107**(2-3), 249–266, (2011).
- [33] Vasco M. Manquinho, João P. Marques-Silva, and Jordi Planes, ‘Algorithms for weighted boolean optimization’, in *Proc. SAT*, volume 5584 of *Lecture Notes in Computer Science*, pp. 495–508. Springer, (2009).
- [34] Norbert Manthey, ‘Coprocessor 2.0 - A flexible CNF simplifier’, in *Proc. SAT*, volume 7317 of *Lecture Notes in Computer Science*, pp. 436–441. Springer, (2012).
- [35] Joao Marques-Silva, Mikolas Janota, Alexey Ignatiev, and Antonio Morgado, ‘Efficient model based diagnosis with maximum satisfiability’, in *Proc. IJCAI*, pp. 1966–1972. AAAI Press, (2015).
- [36] Ruben Martins, Saurabh Joshi, Vasco M. Manquinho, and Ines Lynce, ‘Incremental cardinality constraints for MaxSAT’, in *Proc. CP*, volume 8656 of *Lecture Notes in Computer Science*, pp. 531–548. Springer, (2014).
- [37] Antonio Morgado, Carmine Dodaro, and Joao Marques-Silva, ‘Core-guided MaxSAT with soft cardinality constraints’, in *Proc. CP*, volume 8656 of *Lecture Notes in Computer Science*, pp. 564–573. Springer, (2014).
- [38] Antonio Morgado, Federico Heras, Mark H. Liffiton, Jordi Planes, and Joao Marques-Silva, ‘Iterative and core-guided MaxSAT solving: A survey and assessment’, *Constraints*, **18**(4), 478–534, (2013).
- [39] António Morgado, Mark H. Liffiton, and João Marques-Silva, ‘MaxSAT-based MCS enumeration’, in *Revised Selected Papers of HVC 2012*, volume 7857 of *Lecture Notes in Computer Science*, pp. 86–101. Springer, (2013).
- [40] Nina Narodytska and Fahiem Bacchus, ‘Maximum satisfiability using core-guided MaxSAT resolution’, in *Proc. AAAI*, pp. 2717–2723. AAAI Press, (2014).
- [41] James D. Park, ‘Using weighted MAX-SAT engines to solve MPE’, in *Proc. AAAI*, pp. 682–687. AAAI Press / The MIT Press, (2002).
- [42] Johannes Peter Wallner, Andreas Niskanen, and Matti Järvisalo, ‘Complexity results and algorithms for extension enforcement in abstract argumentation’, in *Proc. AAAI*, pp. 1088–1094. AAAI Press, (2016).
- [43] Lei Zhang and Fahiem Bacchus, ‘MAXSAT heuristics for cost optimal planning’, in *Proc. AAAI*. AAAI Press, (2012).
- [44] Charlie S. Zhu, Georg Weissenbacher, and Sharad Malik, ‘Post-silicon fault localisation using maximum satisfiability and backbones’, in *Proc. FMCAD*, pp. 63–66. FMCAD Inc., (2011).

Vertical Optimization of Resource Dependent Flight Paths

Anders N. Knudsen and Marco Chiarandini and Kim S. Larsen¹

Abstract. Flight routes are paths calculated on a network of waypoints representing 3D-coordinates. A common approach is first to calculate a path in a 2D-network, taking into account feasibility constraints, and then to optimize the altitude of the flight.

We focus on the problem of determining the vertical trajectory defined by an altitude at each waypoint of a 2D-route. The cost of an airway depends, directly, on fuel consumption and on flying time, and, indirectly, on weight and on weather conditions. These dependencies invalidate the FIFO property, which is commonly assumed for time-dependent networks. Moreover, the amount of fuel at departure has an impact on the weight and depends on the length of the route. This, therefore, needs to be decided upon for our problem. We aim at minimizing the total cost.

We study path-finding algorithms, both exact and heuristic, that we iterate in a line-search procedure to decide the fuel amount at departure. We use real-life data for the experimental analysis and conclude that on those data the assumption of the FIFO property remains a good heuristic choice.

1 Introduction

Flight route optimization aims at finding 3D-paths for aircraft in airway networks that minimize the total cost determined by fuel consumption and flying time. It has evident financial and environmental impacts. Airway networks can be huge, due to the added dimension compared with road networks, and side constraints complicate further the problem. Moreover, most of the constraints are determined by a central control institution, e.g., Eurocontrol in Europe and FAA in USA, and change rapidly with time in order to take traffic conditions into account. Therefore, the common practice is to determine the precise flight route a few hours before take-off. For this to be feasible and bring any advantage, the route determination must be very fast, say on the order of a few seconds. If necessary, the route can then be adjusted during the flight by real-time optimization, considering more up-to-date information.

In this work, we focus on the off-line flight route optimization. The common approach in the industry has been to de-

compose the problem: first a shortest, feasible 2D-route is found and, afterwards, the vertical trajectory is optimized. The problem consists in finding a shortest path in a network, made of nodes that are the combination of waypoints, decided by the 2D-route returned by the first stage, and the allowed altitudes for the aircraft. Arcs connect nodes belonging to consecutive waypoints in the 2D-route if the distance between them is enough for the aircraft to make the corresponding altitude transition.

The costs of flying through these arcs depend on two resources: time and fuel. There is a direct dependency because airlines calculate the total cost as a weighted sum of time and fuel consumed and there is an indirect dependency, because the amount of time and fuel consumed depends on the performance of the aircraft, which is influenced by the weather conditions, which in turn depend on time, and by the weight, which in turn depends on the fuel consumed. A consequence of these dependencies is that the cost and the feasibility of each arc are not statically given but become known only when the path to a node is determined. The situation is further complicated by the fact that the amount of fuel at departure is not given, since it is also a decision that can be optimized. The main issue for solution algorithms is that due to the indirect dependencies and the impossibility to wait at the nodes (even at the departure airport), the total cost is a non-monotone function with respect to time and fuel. That is, it may be disadvantageous to arrive cheaply at a node (and hence earlier or lighter than other alternatives) as it may preclude the chance to obtain high savings in the remaining path due to favorable weather conditions developing somewhere at a later time. For labeling algorithms, this means that a total ordering of the labels at the nodes is not available.

In the latest years, a considerable amount of research has focused on engineering aspects of algorithms for the shortest path problem (SPP) on large road networks [1]. Huge improvements have been achieved using several advanced techniques, most of which we deem inapplicable in our setting due to arc costs being unknown before starting the search. In time-dependent SPP arc costs can change with the time of traversal. In one of the first solution attempts to solve this variant, Bellman's iterative algorithm [3] was extended by performing finitely many iterations for any possible starting time [5]. A property of the problem that demarcates solution approaches is the FIFO property. If a network is FIFO, then a vehicle leaving a node cannot be overtaken by another vehicle leaving the same node at a later stage. The SPP in time-dependent FIFO networks is polynomially solvable [9], while

¹ Department of Mathematics and Computer Science, University of Southern Denmark, Denmark

email: {andersnk,marco,kslarsen}@imada.sdu.dk

The first author acknowledges financial support by the Innovation Fund Denmark. The third author was partially supported by the Danish Council for Independent Research, Natural Sciences, grant DFF-1323-00247.

the problem becomes NP-hard for non-FIFO networks [12]. A variation of the FIFO property, which we will continue to refer to as FIFO, can be formulated also for the vertical trajectory problem that we study here. However, in general this property does not hold for this problem. Most of the literature has focused on FIFO networks. In [4], the authors present an extension for FIFO networks that results in significant speed-up over earlier Dijkstra approaches. In [6], a lower bound on the cost from each node to the goal, for use as the heuristic in A*, is calculated in a preprocessing step by backwards breadth-first search assuming the fastest possible speed on each arc. In [2], travel time profile queries are discussed, where the departure time at the source can be shifted at convenience, in a similar fashion as we can decide the amount of fuel to embark at departure. The shortest path with time-dependencies in non-FIFO networks has so far been disregarded in the literature. Articles discussing a problem similar to ours appeared only in specialized literature (see e.g., [13]).

We study the classical shortest path methods, breadth-first[11][10], Dijkstra[7] and A*[8] search, modified to take resource dependencies into account. We compare their performance in terms of quality and running time. In particular, we set out to assess empirically whether it is important in terms of final cost to work without the FIFO assumption and whether it is computationally feasible. We design a lower bound for A*. Under the FIFO assumption we show that this A* search runs faster than Dijkstra's algorithm while still producing optimal solutions under the assumption. Then, we show that the A* approach in [6] but without the FIFO assumption leaves the search computationally infeasible. To improve running time we introduce an upper bound to prune path extensions. We are able to conclude that for the real life data, on which we based our study, the increased scrutiny obtained by relaxing the FIFO assumption does not pay off.

Our work is in collaboration with the industrial partner Aviation Cloud (AC) A/S, a Danish company, whose core business is in flight route planning. The company is interested in an algorithm that can solve the problem very fast, within a few seconds, since it must be used as an aiding tool for flight route planning in an interactive setting. Differently from other sources, e.g., [13], that used the Base of Aircraft Data, an open source database provided by Eurocontrol, our experimental analysis is conducted on data from the specific business case of Aviation Cloud. As common in the sector, aircraft manufacturers supply airlines with look-up tables to calculate the resource consumption of operating the aircraft. Airlines then provide these data to Aviation Cloud. We include a comparison of the quality of our vertical trajectories with the solution currently in use at Aviation Cloud.

2 Problem formulation

Airspaces in different areas of the world are represented by 2D networks, i.e., directed graphs $D_{2D} = (V_{2D}, A_{2D})$. The nodes in V_{2D} represent *waypoints* defined by latitude and longitude coordinates. Altitude is not part of a waypoint description. The arcs in A_{2D} represent feasible connections between waypoints. A 2D (*flying*) route is an (s, g) -path in D_{2D} represented by n waypoints plus a departure node (source) s and an arrival node (goal) g , that is, $R = (s, r_1, \dots, r_n, g)$, $s, r_i, g \in V_{2D}$ for $i = 1..n$, $(r_i, r_{i+1}) \in A_{2D}$ for $i = 1..n - 1$

and $(s, r_1), (r_n, g) \in A_{2D}$.

In this work we assume that a 2D route R is given and we want to optimize the vertical trajectory. Aircraft may only cruise at specific (standard) *flight levels* that may differ from area to area. Hence, at each waypoint r_i , $i = 1..n$, of R we associate a set of flight levels F_1, \dots, F_n . Thus, for a 2D route R a vertical trajectory is an (s, g) -path in a *layered*, directed graph $D_R = (V_R, A_R)$ where $V_R = \{s\} \cup F_1 \cup \dots \cup F_n \cup \{g\}$ and $A_R = A_0 \cup A_1 \cup \dots \cup A_{n-1} \cup A_n$ with $A_0 = \{(s, f) \mid f \in F_1\}$, $A_n = \{(f, g) \mid f \in F_n\}$, $A_i = \{(f, h) \mid f \in F_i, h \in F_{i+1}\}$ for $i \in \{1..n-1\}$. If we denote by m the size of all F_i for $i = 1..n$, then $|V_R| = nm + 2 = \Theta(nm)$ and $|A_R| = (n-1)m^2 + 2m$.

Connections in A_R exist if allowed by regulations and by the operational properties of the aircraft. Aircrafts can only begin a climb or a descent at waypoints but they cannot climb or descend more than at a given rate, which depends on the current weight and weather conditions.

For these reasons the existence of connections in D_R can be determined only during the search for routes when the state of the aircraft at a node becomes known.

We consider fuel and time as resources whose amounts are nonnegative, i.e., they belong to \mathbb{R}_+ . The resource consumption accumulated at a node u in V_R along an (s, u) -path P in D_R is a pair $\vec{\tau}_P = (\tau_P^x, \tau_P^t) \in \mathbb{R}_+^2$, where the superscripts x and t are used to denote the fuel and time components of the resource cost. We assume known the departure time from the starting airport, τ_s^t , and unknown the amount of fuel at departure, τ_s^x .

The changes in the resource consumption associated with an arc $(u, v) \in A_R$ are given by *resource extension function* (REF) values and can be represented by the vector $\vec{f}_{uv} = (f_{uv}^x, f_{uv}^t) \in \mathbb{R}_+^2$. The components of a REF vector, f_{uv}^x and f_{uv}^t , depend on the consumption of resources at the tail node u of arc (u, v) , which can be calculated from the resource consumption accumulated along a path P from s to u . Specifically, the fuel at u is $\tau_s^x - \tau_P^x$ and the time is $\tau_s^t + \tau_P^t$. The resource consumption accumulated along a path $P' = (s, \dots, u, v)$ that extends P by an arc (u, v) is thus $\vec{\tau}_P + \vec{f}_{uv}(\vec{\tau}_P)$.

For a given aircraft, REF values are looked up in a set of tables of *aircraft performance data* that output the time and fuel consumption for an arc using as inputs: (i) flight level of starting and arrival point, (ii) weight, (iii) temperature deviation, (iv) wind component, and (v) cost index. The cost index, ρ , is an efficiency ratio between the time-related cost and the fuel cost that airlines use to specify how to operate the aircraft. It determines the speed of the aircraft and we assume it given. It is decided at strategic level and it cannot be changed during the planning phase. Note that for a path $P = (s, \dots, u)$ and an arc $(u, v) \in A_R$, \vec{f}_{uv} depends on $\vec{\tau}_P^x$ because of (ii) and depends on $\vec{\tau}_P^t$ because of (iii) and (iv).

A path $P = (s, v_1, \dots, v_k)$ in D_R is *resource feasible* if $\tau_{P'}^x > 0$ for any prefix path of P' (there are no restrictions for $\tau_{P'}^t$). Both resources contribute to define the cost function $c: A_R \times \mathbb{R}_+^2 \rightarrow \mathbb{R}_+$ for the network specified by D_R . This function represents the monetary cost of flying through an arc. Airlines use the following formula (see e.g. [13]):

$$c((u, v), \vec{\tau}_P) = f_{uv}^t(f_{uv}^x/f_{uv}^t + \rho).$$

For an arbitrary path $P = (v_1, \dots, v_k)$ in D_R with prefixes $P_i = (v_1, \dots, v_i)$, $i = 2..k-1$, its resource-dependent cost, c_P ,

is defined recursively as follows:

$$\begin{aligned} c_{P_1} &= 0 \\ c_{P_i} &= c_{P_{i-1}} + c((v_{i-1}, v_i), \vec{\tau}_{P_{i-1}}) \end{aligned}$$

We can now, more formally, define our vertical flight trajectory optimization problem.

Definition 1 (Vertical flight trajectory problem). *Given a 2D flight route R from a departure node s to an arrival node g , the layered digraph $D_R = (V_R, A_R)$ constructed from R and the available flight levels, a resource extension function $\vec{f} : A_R \rightarrow \mathbb{R}^2$, a cost function $c : A_R \times \mathbb{R}_+^2 \rightarrow \mathbb{R}_+$, and a departure time τ_s^t at s ; find an amount of fuel at departure, τ_s^x , and a resource feasible (s, g) -path P in D_R such that c_P is minimum, i.e.,*

$$c_P = \min\{c_Q \mid Q \text{ is an } (s, g)\text{-path in } D_R \text{ and } \tau_Q^x > 0 \\ \text{for any prefix } Q' \text{ of } Q\}.$$

Further we define two functions $\lambda, \mu : A_R \rightarrow \mathbb{R}_+$ with the following properties:

$$\lambda(u, v) \leq c((u, v), \vec{\tau}_P) \quad \forall (u, v) \in A_R, \\ \vec{\tau}_P \in \mathbb{R}_+^2, P = (s, \dots, u) \subseteq D_R$$

$$\mu(u, v) \geq c((u, v), \vec{\tau}_P) \quad \forall (u, v) \in A_R, \\ \vec{\tau}_P \in \mathbb{R}_+^2, P = (s, \dots, u) \subseteq D_R$$

In other terms, λ is a lower bound and μ an upper bound on the cost of arcs independent on resources. In our application, the minimal cost can be given by any condition from having the strongest tailwind and carrying as little fuel as possible to having the strongest headwind and carrying the maximum amount of fuel.² The values of λ and μ for every arc $uv \in A_R$ are calculated in a global preprocessing phase and saved in a table. More precisely, for each arc, we look up the corresponding distance and flight level change in the aircraft performance data and try all conditions recording the smallest and highest cost. Distances are discretized and the value to look up is rounded down for calculating λ and up for μ .

Using these functions, it is possible to calculate a lower and an upper bound on the cost of a path from any node $u \in V_R$ to the final node g , i.e.,

$$LB(u, g) = \min \left\{ \sum_{i=1}^n \lambda(v_i, v_{i+1}) \mid (v_1, \dots, v_{n+1}) \right. \\ \left. \text{path in } D_R \text{ with } v_1 = u, v_{n+1} = g \right\}$$

$$UB(u, g) = \max \left\{ \sum_{i=1}^n \mu(v_i, v_{i+1}) \mid (v_1, \dots, v_{n+1}) \right. \\ \left. \text{path in } D_R \text{ with } v_1 = u, v_{n+1} = g \right\}$$

² Indeed, although counterintuitive, high weight may imply low cost, because it may allow descents at a speed otherwise impossible. Similarly, strong headwind may allow an airplane to climb or descend larger differences in altitude for a given ground distance than under normal conditions.

These values can be computed in a preprocessing stage by a backwards breadth-first search.

An analogous of the FIFO property (or non-overtaking property) for time-dependent networks can be defined for our case as follows.

Definition 2 (FIFO property). *For any pair of nodes, $u, v \in V_R$, and any pair of paths, $P = (s, \dots, u)$ and $Q = (s, \dots, u)$, arriving in u with $\vec{\tau}_P$ and $\vec{\tau}_Q$, respectively, if $c_P < c_Q$ then $c_P + c((u, v), \vec{\tau}_P) \leq c_Q + c((u, v), \vec{\tau}_Q)$.*

However, this property may be violated in our case. Indeed, if at node u an (s, u) path Q is more expensive than an (s, u) -path P ($c_P < c_Q$) then, because of the way the cost is calculated, it must be that Q has consumed more time or more fuel or both. But in all these cases it is still possible that there is a cheaper (s, \dots, u, v) path extending Q rather than P if, for example, weather conditions at time $\tau_s^t + \tau_Q^t$ are more beneficial than at time $\tau_s^t + \tau_P^t$ (and waiting is not possible). Although the FIFO property seems not to hold in theory it is unclear how much it is violated in the real data and how much worse we do by assuming it as a heuristic. We study therefore both the FIFO and the non-FIFO cases.

3 Algorithm design

The overall solution approach is sketched in Alg. 1. The main function, `FINDFUELANDPATH()`, performs a line search on the amount of fuel at departure. We start with an initial guess $\tau_{s,0}^x$. Then at each iteration i , we call a path finding procedure, `FINDPATH`, with the current guess $\tau_{s,i}^x$. If the amount of fuel is sufficient to find a path in D_R from s to g such that $\tau_{g,i}^x \geq 0$, then we update our guess with $\tau_{s,i+1}^x = \tau_{s,i}^x - \tau_{g,i}^x$. Otherwise, the path finding procedure continues with empty tank until the goal is reached accumulating a negative $\tau_{g,i}^x$, which is used to update $\tau_{s,i+1}^x$ yielding an increase of value. We continue in this way until the tentative values converge, that is, when the amount of fuel carried, but not spent, $\tau_{g,i}^x$, is less than 2% of the fuel at departure, $\tau_{s,i}^x$. Depending on the discretization of the aircraft performance data, the algorithm might never reach this state. This can be counteracted by increasing the threshold after several iterations. However, this was not necessary in any of our tests.

The core of the `FINDFUELANDPATH` function is the path finding function `FINDPATH`, which will be executed a few times. We focus, therefore, on algorithms to make this function effective and efficient. We consider breadth-first search and variants of best-first search. In Alg. 1 we give a general template that remains valid for all the algorithms described below. A *label* is a piece of information associated with a node of D_R and created when the algorithm first considers or “opens” a node. The expansion of a label is the operation of generating a new label for any node in D_R reachable by an arc going out from the node of the label under expansion.

The algorithm takes as input information about how to construct the network, the REF tables, the amount of fuel at departure and the departure time. It then works with a list of open labels \mathcal{Q} and in turn expands labels from this list. The specific algorithms differ by how a label is selected for expansion (line 13) and by how it is inserted in the list (line 20).

```

1 Function FINDFUELANDPATH( $D_R, \vec{f}, c, \tau_s^t$ )
2   initialize  $\tau_{s,0}^x$ 
3    $P, \tau_{g,0}^x, \tau_{g,0}^t \leftarrow$  FINDPATH( $D_R, \vec{f}, c, \tau_{s,0}^x, \tau_{s,0}^t$ )
4    $i \leftarrow 0$ 
5   while  $\tau_{g,i}^x \geq 0.02 \cdot \tau_{s,i}^x$  do
6      $P, \tau_{g,i}^x, \tau_{g,i}^t \leftarrow$  FINDPATH( $D_R, \vec{f}, c, \tau_{s,i}^x, \tau_{s,i}^t$ )
7      $\tau_{s,i+1}^x = \tau_{s,i}^x - \tau_{g,i}^x$ 
8      $i = i + 1$ 
9   return  $P, \tau_s^x$ 

10 Function FINDPATH( $D_R, \vec{f}, c, \tau_s^x, \tau_s^t$ )
11   initialize the open list  $\mathcal{Q}$  by inserting a label for the
    departure node
12   initialize  $\ell$  with an empty path of cost infinite
13   while true do
14      $\ell' \leftarrow$  select a label from  $\mathcal{Q}$   $\triangleright$  Selection criterion
    depends on algorithm
15     if (cost of  $\ell'$  greater than cost of  $\ell$ ) then break
16     if ( $\ell'$  is at destination) and (cost of  $\ell'$  smaller
    than cost of  $\ell$ ) then
17        $\ell \leftarrow \ell'$ 
18       break
19     foreach reachable and allowed node at next
    waypoint do
20       calculate cost of reaching node
21       add new label to  $\mathcal{Q}$   $\triangleright$  Insertion depends on
    algorithm
22   return  $P, \tau_g^x, \tau_g^t$  of  $\ell$ 

```

Algorithm 1: A general template for solving the flight trajectory problem

When expanding a label on line 18 the possible reachable nodes in the next layer are calculated. The distance between consecutive waypoints is given by the 2D-route. It is then possible to calculate which flight levels in the next layer can be reached from a node by iteratively trying higher or lower altitudes until the distance available is not enough for the climb or descent. Using the aircraft performance data one can retrieve the REF vector for each of the reachable altitudes and for each node create a label. Since aircraft are allowed to reach waypoints outside of standard flight levels, we might have to add labels at nodes that are not associated with a standard flight level. These labels are *flagged*, meaning that they can be expanded only by a climbing or descending arc and not by a cruising arc. Only two of these labels need to be created: one at the maximum climb rate and one at the maximum descent rate. Due to the discretization of the performance data, we allow only one flagged label of each kind between each pair of adjacent standard flight levels. The influence of these on the graph is discussed in the analysis section.

Initial fuel amount The initial fuel, $\tau_{s,0}^x$, which is used as the first value in the line search, is calculated as follows. We start with the maximum load and adjust its value using the information gathered by a linear time greedy algorithm. This algorithm expands only one label at each waypoint. The label expanded belongs to the next waypoint and has flight level equal to the minimum between a given value h and the highest flight level reachable given the current condition of

the aircraft. Whenever from a waypoint the destination node cannot be reached at the steepest descent, the construction backtracks to the previous waypoint and descends directly to the destination from there. After such construction the amount of fuel at departure is updated. The whole procedure is repeated decreasing h through all standard flight levels and halting when either 5 iterations are done or the fuel amount left after the path is less than 2% of the value at departure. Finally, $\tau_{s,0}^x$ is set equal to the value that generated the cheapest route during this phase.

Breadth-first search All labels at one layer of D_R (corresponding to all the different flight levels at one waypoint of the 2D route) are expanded before moving on to the next waypoint. The order of expansion is not relevant. Using the FIFO property, label domination occurs at each node: for multiple labels reaching a node only the cheapest one is kept. We denote this baseline algorithm by **F-BFS**.

Dijkstra The open list is sorted according to the current cost of the labels and the cheapest label is selected for expansion. The FIFO property is used for label domination and an additional list of closed nodes prevents expanding nodes more than once. We denote this algorithm by **F-Dijk**.

A* The cost of labels is given by the sum of the cost of the path to the corresponding node plus a heuristic cost of the path from the node to the goal. The cheapest label is selected for expansion and, under the FIFO assumption, at any given node, the cheapest label dominates the others, which are then removed from (or never added to) the open list. To guarantee optimality the heuristic must be admissible, i.e., it never overestimates the cost of reaching the goal, and consistent, i.e., for every node u , the estimated cost of reaching the goal from there is no greater than the cost of getting to a successor v plus the estimated cost from v to the goal. It is easy to show that consistency is a stronger property as it implies admissibility. We use the lower bounds LB defined in Sec. 2 that is thus both consistent and admissible. We denote this algorithm **F-A***.

Non-FIFO A* Lifting the FIFO assumption in **F-A*** means that labels cannot be dominated at nodes based purely on the cost of reaching that node. This algorithm, denoted by **nF-A***, does not use the FIFO assumption but it tries to reintroduce some form of domination at nodes. It does so using the upper bound UB (see Sec. 2). Let P and Q be two paths in D_R from s to u and let ℓ and ℓ' be the corresponding labels in u , respectively. The label ℓ' is dominated by ℓ , if $c_Q > c_P$ and $c_Q - c_P > UB(u, e) - LB(u, e)$. That is, it is impossible for the most expensive label ℓ' to gain the difference in cost over the cheapest label ℓ in the remaining path to destination even if in the case that ℓ' continues in the best possible way and ℓ in the worst possible way. When a new label is created it may either be dominated by the currently cheapest label on that node, or it may dominate any number of the more expensive current labels at the node. Selection is still based on the cost of the path plus the heuristic cost to the goal.

A* with an inconsistent and inadmissible heuristic Experimental analysis shows that **nF-A*** lacks efficiency. The heuristic via lower bound LB is quite weak and this is bad when domination can occur only seldom. Further experimental observations suggested that using always maximum tail

wind and zero fuel in the calculations of the λ values associated to each arc led to a stronger LB bound and therefore to faster searches. However, as explained earlier in this way we cannot guarantee the consistency and admissibility of the heuristic. However in order to be able to solve instances of a considerable sizes, we introduce a variant of $\mathbf{nF-A}^*$, which uses the stronger bound. We denote it $\mathbf{nF-iA}^*$. In Sec. 5 we analyze the trade-off between computation speed and quality of the solutions in the two variants.

We also tested the impact of the stronger lower bound in a FIFO setting. $\mathbf{F-iA}^*$ is a variant of $\mathbf{F-A}^*$ that uses the stronger bound, but also allows a node to be revisited, should a cheaper label be discovered later. The point of allowing revisits is that it causes inconsistency to no longer lead to suboptimal results. This limits the amount of instances where the algorithm does not find optimal solutions. However, it does not fix problems caused by inadmissibility, so optimality is still not guaranteed.

4 Analysis

Data structures The open list is implemented as a red-black tree. We maintain a pointer to the minimum cost label, so we are able to retrieve it for expansion in amortized constant time. Access is worst case constant time and rebalancing after deletion is amortized constant time. To check domination, when a new label is created, the label must be compared with the minimum cost label at the same node. For $\mathbf{F-Dijk}$ and $\mathbf{F-A}^*$, there can only be a single label for each node and so they can be stored in a simple list. For $\mathbf{nF-A}^*$, we need to keep track of the labels per node and be able to retrieve the minimum and the maximum cost labels. This is achieved by keeping a red-black tree of labels at each node with a pointer to the minimum cost label and one to the maximum. The entries in these trees have pointers to the corresponding entries in the open list and vice versa. Then we are able to perform all extractions and deletions in the two structures in amortized constant time. Finally, the closed list is implemented as a bitwise array with one bit per node.

FIFO algorithms Let n be the number of waypoints and m the number of standard flight levels. Due to the domination in $\mathbf{F-BFS}$, $\mathbf{F-Dijk}$, and $\mathbf{F-A}^*$, we can expand exactly one label for each node at a standard flight level. Hence, the size of the open list and the number of labels we can expand is $O(nm)$. Each expanded label can generate a maximum of m additional labels. This means that we can create $O(nm^2)$ labels overall. This is also true when taking into account the climb- and descent-only labels. Whenever expanding a label leads to the creation of either, it means that some number of standard labels were not created. The expansion of the climb- or descent-only label can then generate exactly that number of additional labels, leading to a maximum of $m+2$ labels created per expansion if both kinds are created. Due to allowing revisits the labels created by $\mathbf{F-iA}^*$ may increase exponentially to $O(m^n)$.

Upon creation, a label must be inserted into the open list and into the node list. $\mathbf{F-BFS}$ simply inserts it into a simple list in $O(1)$. $\mathbf{F-Dijk}$, $\mathbf{F-A}^*$ and $\mathbf{F-iA}^*$ insert them into the red-black tree that implements the open list in $O(\log(nm))$. With our datasets this cost is, however, dominated by the constant time needed to expand a label, due to several look-ups to calculate the cost of an arc.

Non-FIFO Since domination is restricted in both $\mathbf{nF-A}^*$ and $\mathbf{nF-iA}^*$, the number of labels in the open list grows exponentially with respect to the depth of the search, which we know to be n . Thus, the open list can end up containing $O(m^n)$ labels. The insertion of each label costs $O(n \log m)$.

Preprocessing The \mathbf{A}^* algorithms need the values $LB(u, g)$ for all $u \in V_R$. Since these values are resource independent we calculate them once for every (s, g) -path query, that is, before running $\mathbf{FINDPATH}$ and hence it is unaffected by the repetition of $\mathbf{FINDPATH}$. We do this by a backwards breadth-first search in $O((n \cdot m) \cdot (n + m))$. For $\mathbf{nF-A}^*$ we also calculate $UB(u, e)$ for all $u \in V$. This latter needs additional $O((n \cdot m) \cdot (n + m))$ for a backwards breadth-first search.

5 Experimental Assessment

All algorithms described above except $\mathbf{nF-A}^*$ are heuristic algorithms, that is, because of the FIFO assumption or the use of inconsistent and inadmissible heuristics they do not guarantee to terminate with optimal solutions. We set out then to assess the deterioration in quality of the heuristic algorithms and their running time with the goal of suggesting the best trade off. We also compare the quality of our solutions with those found by \mathbf{AC} , a fast greedy algorithm previously in use at Aviation Cloud.

We use real life data provided by Aviation Cloud. This data consists of aircraft performance data, weather data forecast in standardized GRIB2 format and 2D-routes provided by Aviation Cloud's route planner. The aircraft performance data refers to one single aircraft and the standard flight levels are from 0 to 48000 feet in intervals of 200 feet. The weather data forecast are from three different days with intervals of three hours and with varying top wind speeds (the strongest wind speed recorded was 270 km/h). A specific test instance (or query) is determined by the time of departure and the 2D-route. The same aircraft performance data and weather conditions are shared by several instances. All tests were run with a default of 5% contingency fuel. To account for fluctuations in CPU time measurement, each instance was tested 15 times and only the fastest was recorded.

The instances have been grouped by the number of waypoints in the 2D-routes, varying from 11 to 50 with increments of 1. This is the most important factor for the complexity of the problem. For each different number of waypoints 30 routes are available, arbitrarily chosen among the routes serviced by Aviation Cloud. Overall we tested on 3600 instances, which should be enough to guarantee that the visual differences between algorithms in the following analysis are statistically significant.

All algorithms were implemented in C# and the tests were carried out on a virtual machine in a cloud environment with an Intel Xeon E5-2673 processor at 2.40Ghz and with 7GB RAM. We introduced time limits for the non-FIFO algorithms that were 30 seconds for $\mathbf{nF-iA}^*$ and 10 minutes for $\mathbf{nF-A}^*$. $\mathbf{nF-A}^*$ was able to solve instances only with less than 14 waypoints. FIFO \mathbf{A}^* algorithms used in the worst case 100 MB of memory of which 90 MB was used to store the preprocessed aircraft performance data. $\mathbf{nF-iA}^*$ reached 500 MB when the time limit was reached and $\mathbf{nF-A}^*$ reached more than 5 GB in several instances.

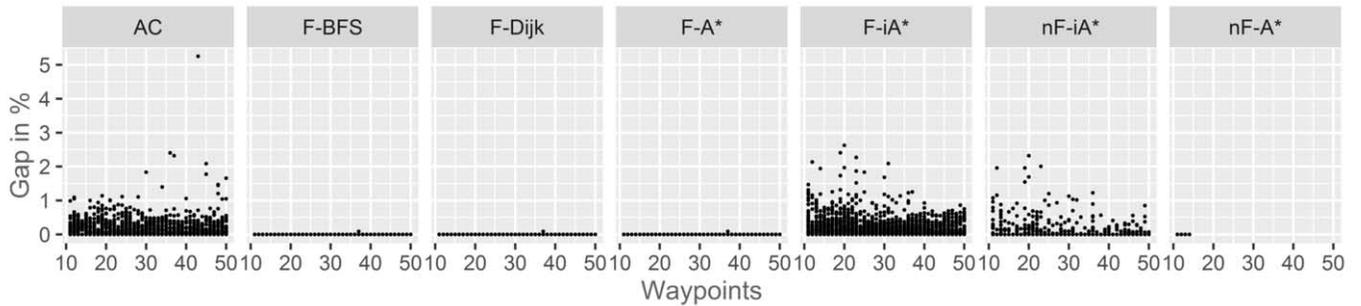


Figure 1. Scatter plot of percentage gap to the best solutions on each instance.

Cost assessment We ran each algorithm i on each instance j and recorded the cost returned by the algorithm, x_{ij} , and the cheapest cost found by any algorithm in our study on an instance, x_j^* . We then computed the percentage gap $y_{ij} = 100 \cdot (x_{ij} - x_j^*)/x_j^*$. A scatter plot of this metric is reported in Fig. 1. We observe that F-BFS, F-Dijk, and F-A* find trajectories of exactly same cost. nF-A* also finds the same solutions in the instances it is able to solve. These solutions are also always the best except for one single instance where the two non-FIFO algorithms find a better solution. This is the only case out of 3600 where we observed that not assuming the FIFO property is necessary to achieve better solutions. For instances below 14 waypoints, where nF-A* terminated, a gap equal to zero indicates that the solution is provably optimal. For more waypoints the optimal solutions could be better than those found in our experiments and hence the impact of assuming FIFO might be more pronounced. Another explanation for the low impact of the FIFO assumption can be the low frequency of changes in our weather data. Given more frequent changes, the impact might be more relevant. It should also be noted that fluctuating weather conditions could make the optimal route bumpy, and, therefore, unpleasant for the passengers. Our algorithm could be adjusted to prevent bumps however this turned out to be unnecessary in our tests that resulted always in stable routes.

As expected, inconsistency and inadmissibility issues in both F-iA* and nF-iA* seem to have a considerable impact on the deterioration of solution quality. Finally, the solutions we found with any algorithm in this study are better than those of AC.

Computation cost assessment The running time of the whole trajectory optimization algorithm comprises the time to carry out preprocessing computations, where needed, and the time to execute FINDPATH until convergence. The number of calls to FINDPATH was in most of the cases two or three with peaks of four or five with many waypoints. There is no significant influence on this number by the algorithm used to implement FINDPATH, hence in the analysis that follows we consider results only on preprocessing and single runs of FINDPATH. A scatter plot of the number of opened and expanded nodes and of the computation time in milliseconds is reported in Fig. 2.

It is evident from the figure that the number of labels opened and expanded is much higher in non-FIFO algorithms. As a consequence several runs do not terminate within the time limit. For this reason we stopped prematurely running

nF-A* on instances with more than 14 waypoints. The use of the inconsistent and inadmissible heuristic in nF-iA* was instead effective. It reduces by a good margin the amount of labels opened and expanded, and thus nF-iA* can solve also the instances with high number of waypoints, although still not all. This comes at the cost of suboptimal routes though.

We observe that F-Dijk opens fewer nodes than F-BFS but expands about the same number of nodes and hence has almost the same running times. Optimal paths typically climb to a high altitude and then cruise at that altitude for most of their lengths. However, F-Dijk gets trapped expanding labels of descending nodes, as these are almost always cheaper. Whenever F-BFS opens a label, it only needs to insert it into a list, whereas F-Dijk needs to add it to the red-black tree. This fact has however no evident consequence on the running time, which is anyway dominated by the calculation of the cost of an arc and, hence, by the number of labels expanded.

F-A* opens and expands less nodes than F-Dijk and is therefore also slightly faster, even including the preprocessing time required to do the backwards breadth-first search to find the heuristic values. However, as mentioned earlier the bound is not very strong and thus the decrease in search space is not impressive. F-iA* reduces the search space by a much larger margin, and is also considerably faster. Once again though, this comes at the cost of suboptimal routes.

6 Conclusions

We found impractical from a computational point of view finding optimal trajectories without assuming the FIFO property. Then, our best suboptimal algorithm without assuming FIFO is nF-A*. However, we found only a single instance out of 3600 tested, where this algorithm finds better solutions than those found by much faster algorithms that instead assume the FIFO property. To solve the problem in a FIFO setting, either F-A* or F-iA* is the best choice, depending on whether solution cost or running time is deemed most important. F-A* finds always the solution of best cost, which is optimal in the most common case that the FIFO property holds in the data. F-A* finds less good solutions but is faster.

The design of a heuristic for A* that makes the algorithm optimal while maintaining its efficiency is a possible focus for further research. However, important savings in costs could be found by an integrated solution approach for 3D route optimization, where the 2D route and the trajectory are optimized together.

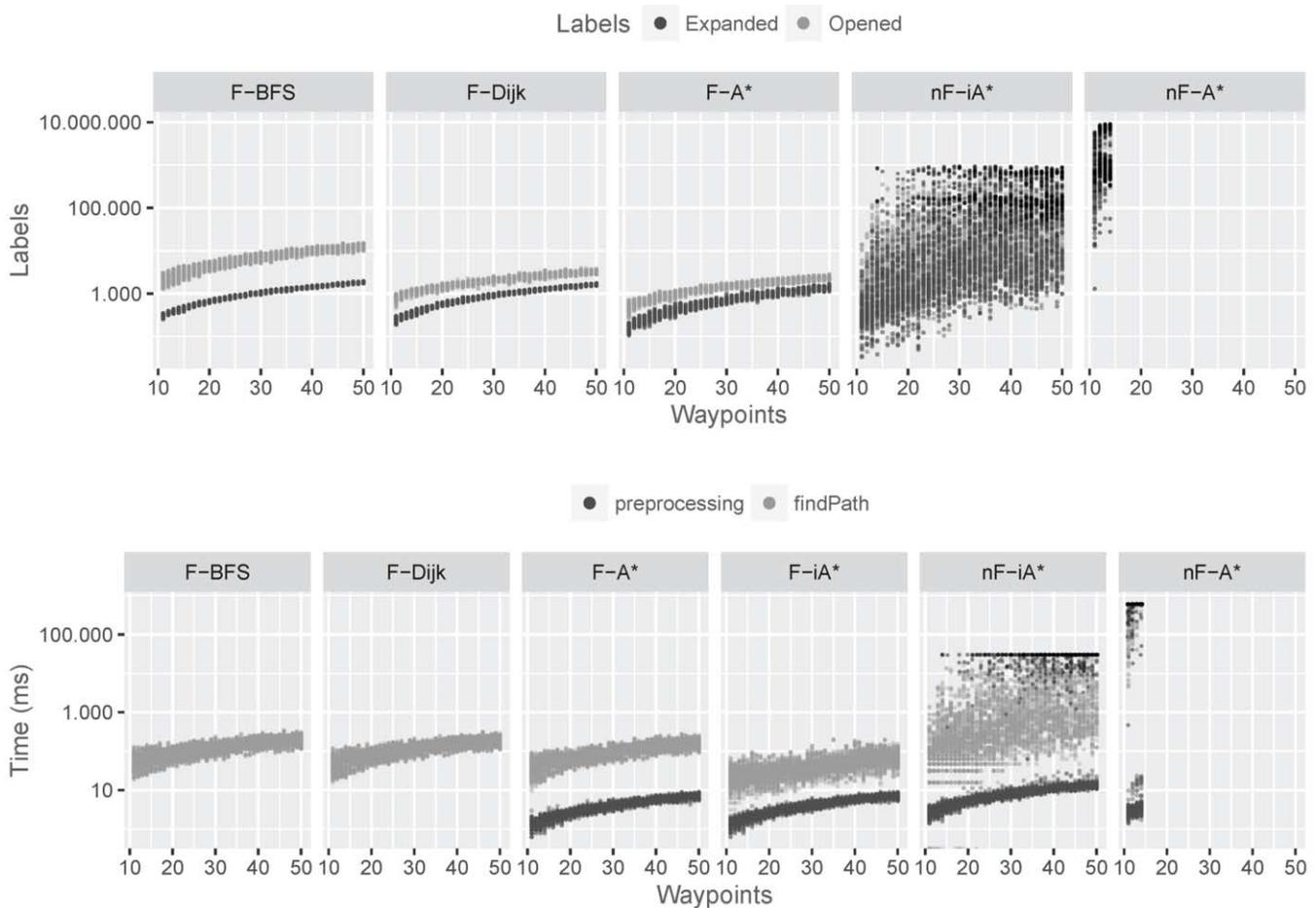


Figure 2. Above, semi-logarithmic plot of the number of opened and expanded nodes; below, semi-logarithmic plot of the computation time of preprocessing and FINDPATH. There is no preprocessing in F-BFS and F-Dijk. Data points related to runs that exceeded the time limit are marked in black.

References

- [1] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R.F. Werneck, 'Route planning in transportation networks', Technical Report arXiv:1504.05140 [cs.DS], arXiv, (2015).
- [2] G.V. Batz, R. Geisberger, P. Sanders, and C. Vetter, 'Minimum time-dependent travel times with contraction hierarchies', *ACM Journal of Experimental Algorithmics*, **18**(1), Article No. 1.4, (2013).
- [3] R. Bellman, 'On a routing problem', Technical Report P-1000, Defense Technical Information Center, (1956).
- [4] I. Chabini and S. Lan, 'Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks', *IEEE Transactions on Intelligent Transportation Systems*, **3**(1), 60–74, (2002).
- [5] K.L. Cooke and E. Halsey, 'The shortest route through a network with time-dependent internodal transit times', *Journal of mathematical analysis and applications*, **14**(3), 493–498, (1966).
- [6] D. Delling and G. Nannicini, 'Bidirectional core-based routing in dynamic time-dependent road networks', in *International Symposium on Algorithms and Computation (ISAAC)*, volume 5369 of *LNCS*, pp. 812–823. Springer, (2008).
- [7] E. W. Dijkstra, 'A note on two problems in connexion with graphs', *Numerische Mathematik*, **1**(1), 269–271, (1959).
- [8] P. E. Hart, N. J. Nilsson, and B. Raphael, 'A formal basis for the heuristic determination of minimum cost paths', *IEEE Transactions on Systems Science and Cybernetics*, **4**(2), 100–107, (July 1968).
- [9] D.E. Kaufman and R.L. Smith, 'Fastest paths in time-dependent networks for intelligent vehicle-highway systems application', *Journal of Intelligent Transportation Systems*, **1**(1), 1–11, (1993).
- [10] C. Y. Lee, 'An algorithm for path connection and its applications, ec-10(3)', *IRE Transactions on Electronic Computers*, 346–365, (1961).
- [11] Edward F. Moore, 'The shortest path through a maze', in *The International Symposium on the Theory of Switching*, pp. 285–285. Harvard University Press, (1959).
- [12] A. Orda and R. Rom, 'Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length', *Journal of the ACM*, **37**(3), 607–625, (1990).
- [13] R.S.F. Patrón, Y. Berrou, and R. Botez, 'Climb, cruise and descent 3d trajectory optimization algorithm for a flight management system', in *AIAA/3AF Aircraft Noise and Emissions Reduction Symposium*. American Institute of Aeronautics and Astronautics (AIAA), (June 2014).

Exploiting Bayesian Network Sensitivity Functions for Inference in Credal Networks

Janneke H. Bolt¹ and Jasper De Bock² and Silja Renooij³

Abstract. A Bayesian network is a concise representation of a joint probability distribution, which can be used to compute any probability of interest for the represented distribution. Credal networks were introduced to cope with the inevitable inaccuracies in the parametrisation of such a network. Where a Bayesian network is parametrised by defining unique local distributions, in a credal network sets of local distributions are given. From a credal network, lower and upper probabilities can be inferred. Such inference, however, is often problematic since it may require a number of Bayesian network computations exponential in the number of credal sets. In this paper we propose a preprocessing step that is able to reduce this complexity. We use sensitivity functions to show that for some classes of parameter in Bayesian networks the qualitative effect of a parameter change on an outcome probability of interest is independent of the exact numerical specification. We then argue that credal sets associated with such parameters can be replaced by a single distribution.

1 INTRODUCTION

Ever since the introduction of Bayesian networks [19], we have seen a growing interest for probabilistic graphical models in AI. A probabilistic graphical model concisely represents joint probability distributions over a set of stochastic variables, by combining the use of a graph to represent the independence relation among the variables with probability distributions over subsets of those variables. A *Bayesian network* defines a unique joint probability distribution by combining an acyclic directed graph with local discrete distributions, one for each node in the graph conditioned on its parents. A Bayesian network can be used to infer any probability of interest from the represented distribution.

Since in practice the specified probabilities, also called parameters, may be inaccurate, methods were developed to cope with such inaccuracies. A sensitivity analysis, for example, can be used to study the impact of inaccuracies on outcome probabilities of interest [16]. Alternatively, *credal networks* adopt the framework of imprecise probabilities [2, 23] by allowing the use of closed convex sets of distributions (also called credal sets), rather than defining unique local distributions per variable. Credal networks as such represent a set of Bayesian networks [1, 11, 12].

One of the main computational problems in a credal network is the computation of tight lower and upper bounds on the outcome probabilities that correspond to the represented Bayesian networks. These computations in essence require a number of Bayesian network inferences that is exponential in the number of imprecise local credal

sets, which are the local credal sets that consist of more than one distribution.

Although this combinatoric explosion can, in general, not be avoided, in many cases a reduction is possible. For example, similar to what can be done for Bayesian networks, it is possible to apply a preprocessing step in which the variables that are irrelevant for a specific problem, such as d-separated or barren nodes, are pruned in advance [11]. In this paper, we introduce a second type of preprocessing step to reduce the combinatoric complexity of credal network inference. Unlike the use of d-separation or the removal of barren nodes, our preprocessing step is tailored specifically to credal networks, and does not apply to Bayesian networks, as it involves replacing some of the local imprecise credal sets by a single distribution.

In order to prove the validity of our preprocessing step for credal network inference, we exploit results that we derive from Bayesian network sensitivity functions. More specifically, we use sensitivity functions to prove that for certain categories of distributions we can predict the qualitative effect of changing their parameters on an outcome probability, irrespective of the numerical specification of the network. In a credal network, this result allows us to identify, prior to the inference, local credal sets that can be replaced by a single distribution without changing the result of the inference. By applying this preprocessing step, the number of imprecise local credal sets and the corresponding combinatoric complexity of inference is reduced. Moreover, we demonstrate that for some widely used special classes of networks even all credal sets can be replaced by a single distribution, thereby essentially reducing the problem of computing a lower or upper probability in a credal network to an inference problem in a single Bayesian network.

This paper is organised as follows. In Section 2, we introduce the notation used throughout the paper. In Section 3, we start with a brief review of Bayesian networks and subsequently derive the properties of sensitivity functions for Bayesian networks that we want to exploit in the context of credal networks. Section 4 introduces the concept of a credal network, and then uses the results in Section 3 to establish the aforementioned preprocessing technique. We end the paper with conclusions and suggestions for future research.

2 NOTATION

This paper is concerned with graphical models for (sets of) joint probability distributions $\Pr(\mathbf{V})$ over a set of finite-valued random variables \mathbf{V} .

We use upper case V to denote a single random variable, writing $v \in V$ to indicate a value (lowercase v) of V . For binary-valued V , we use v and \bar{v} to denote its two possible value assignments. Boldfaced letters are used to indicate sets (capitals), both of variables

¹ Utrecht University, The Netherlands, email: j.h.bolt@uu.nl

² Ghent University, Belgium, email: jasper.debock@ugent.be

³ Utrecht University, The Netherlands, email: s.renooij@uu.nl

and of value assignments, or a joint value assignment (lowercase) to a set of variables; the distinction will be clear from the context. For example $V \in \mathbf{W}$ indicates that V is a variable in a set of variables \mathbf{W} , whereas $\mathbf{w} \in \mathbf{W}$ indicates that \mathbf{w} is a joint value assignment of that same set of variables \mathbf{W} .

Two value assignments are said to be compatible, denoted by \sim , if they agree on the values of the shared variables; otherwise they are said to be incompatible, denoted by \approx . For example, for three distinct variables U, V and W in \mathbf{V} , and value assignments $u, u' \in U, v \in V$ and $w \in W$ such that $u \neq u'$, we have that $uv \sim uvw$ and $uv \approx u'vw$. Assignments to disjoint sets are always considered to be compatible.

The probabilistic graphical models that we consider are Bayesian networks and credal networks. The graphical part of these models consists of a directed graph, the nodes of which have a one-to-one correspondence with the random variables V in \mathbf{V} . For a given node V , π indicates its set of parents in the directed graph, and we will use $\boldsymbol{\pi}$ to indicate a joint instantiation of these parents. Without loss of generality, we assume that the set \mathbf{V} of all random variables consists of the disjoint subsets \mathbf{H} (hypothesis variables), \mathbf{E} (evidence variables) and \mathbf{R} , with $\mathbf{V} = \mathbf{H} \cup \mathbf{E} \cup \mathbf{R}$. The sets of hypothesis variables and evidence variables that do not have parents in \mathbf{R} are denoted by $\mathbf{H}_{|\mathbf{R}}$ and $\mathbf{E}_{|\mathbf{R}}$, respectively. Similarly, the set $\mathbf{H}_{|\mathbf{R}}$ consists of those hypothesis variables that have no parents in \mathbf{H} .

Finally, when referring to monotonic functions, we will use the term *increasing* to indicate both strictly increasing and non-decreasing functions; likewise *decreasing* refers to both strictly decreasing and non-increasing functions.

3 BAYESIAN NETWORKS' SENSITIVITY PROPERTIES

In this section, we first briefly review Bayesian networks and sensitivity analysis. Subsequently, we establish guarantees on the behaviour of sensitivity functions for some categories of parameters. Next, we use these results to define types of local distributions, which will then allow us in Section 4 to exploit our results in the context of credal networks.

3.1 Bayesian network preliminaries

A Bayesian network \mathcal{B} is a concise representation of a joint probability distribution. It uses a directed acyclic graph to capture the (in)dependences among its variables \mathbf{V} using the well-known d-separation criterion [19].⁴ Furthermore, for each variable $V \in \mathbf{V}$, it specifies exactly one local distribution $\Pr(V|\boldsymbol{\pi})$ over the values of V for each value assignment $\boldsymbol{\pi}$ to $\boldsymbol{\pi}$; the separate probabilities in these distributions are termed the network's *parameters*. The joint probability distribution \Pr factorises over the local distributions as follows:

$$\Pr(\mathbf{v}) = \prod_{V \in \mathbf{V}} \Pr(v|\boldsymbol{\pi})|_{v\boldsymbol{\pi} \sim v}$$

where the notation $|\text{prop}$ is used to indicate the properties the arguments in the preceding formula adhere to. Outcome probabilities of the form $\Pr(\mathbf{h}|\mathbf{e})$ can be computed with various algorithms [15].

⁴ For any three disjoint sets of nodes X, Y, Z , the set Z is said to d-separate X and Y if there do not exist any active chains between X and Y given evidence for Z . A chain between two nodes is active if each of its head-to-head nodes is either instantiated or has an instantiated descendant, and none of its other nodes are instantiated. The variables captured by d-separated nodes are considered probabilistically independent.

An example Bayesian network is shown in Figure 1. The figure shows a graph with hypothesis variables $\mathbf{H} = \{G, H, K\}$ (double circles), evidence variables $\mathbf{E} = \{E, F\}$ (shaded), and remaining variable $\mathbf{R} = \{R\}$. The figure also shows conditional probability tables (CPTs) that specify the local distributions. Actual numbers are not provided, but the top row in the table for $\Pr(H|G)$ could for example specify the parameters $\Pr(h|g) = 0.4$ and $\Pr(\bar{h}|g) = 0.6$.

The probabilities computed from a Bayesian network depend on the parameters specified in the network, which are inevitably inaccurate. To study the effects of such inaccuracies on an output probability of interest, a sensitivity analysis can be performed. One approach to performing a sensitivity analysis is to compute so-called sensitivity functions [16]. An n -way sensitivity function describes an output probability of interest $\Pr(\mathbf{h}|\mathbf{e})$ as a function of n network parameters $\vec{x} = \{x_1, \dots, x_n\}$:

$$\Pr(\mathbf{h}|\mathbf{e})(\vec{x}) = \frac{\Pr(\mathbf{h}\mathbf{e})(\vec{x})}{\Pr(\mathbf{e})(\vec{x})}$$

where both the numerator and the denominator are known to be multi-linear expressions in \vec{x} . For $n = 1$, the one-way sensitivity function takes the general form

$$\Pr(\mathbf{h}|\mathbf{e})(x) = \frac{\tau_1 \cdot x + \tau_2}{\kappa_1 \cdot x + \kappa_2}$$

where the constants τ_1, τ_2, κ_1 and κ_2 are composed of non-varied network parameters. Throughout this paper we assume $\Pr(\mathbf{e})$ to be, and to remain, strictly positive.

Upon varying a parameter x of a distribution, the other parameters of the same distribution have to be co-varied to let the distribution sum to 1. If the distribution is associated with a binary variable, the co-varying parameter equals $1 - x$. If a variable is multi-valued, however, different co-variation schemes are possible [21].

Based upon graphical considerations alone, in computing a probability $\Pr(\mathbf{h}|\mathbf{e})$, and thus also for performing sensitivity analyses with respect to this output probability, a Bayesian network can be safely pruned by removing variables that are either d-separated from \mathbf{H} given \mathbf{E} , or that are barren given these sets.⁵ The remaining variables all have evidence or parameters that may be required for the computation of $\Pr(\mathbf{h}|\mathbf{e})$ [22]. This set of requisite variables coincides with the union of \mathbf{E} and the so-called sensitivity set [18], which is the set of variables for which a change in one of its parameters may result in a change in $\Pr(\mathbf{h}|\mathbf{e})$ [10]. In our example network in Figure 1 all variables are in the sensitivity set of $\Pr(ghk|ef)$.

3.2 Guaranteed effects of parameter changes

In this section, we first consider one-way sensitivity functions $\Pr(\mathbf{h}|\mathbf{e})(x)$ and identify categories of parameters that are guaranteed to give monotonically increasing and parameters that are guaranteed to give monotonically decreasing one-way functions. We also identify parameters inside the sensitivity set of $\Pr(\mathbf{h}|\mathbf{e})$ that are not used in computing the outcome probability $\Pr(\mathbf{h}|\mathbf{e})$. Next, we use these results to provide guarantees on n -way functions $\Pr(\mathbf{h}|\mathbf{e})(\vec{x})$.

We begin by categorising the different parameters $x = \Pr(v|\boldsymbol{\pi})$ of a Bayesian network with respect to a specific outcome probability $\Pr(\mathbf{h}|\mathbf{e})$. We only categorise the subset of all parameters for which we can provide guarantees on their qualitative effect on the outcome probability. The categorisation is given in Table 1. For our example network in Figure 1, the categories are indicated with respect to the output probability $\Pr(ghk|ef)$.

⁵ Barren variables are in the set \mathbf{R} and have just barren descendants.

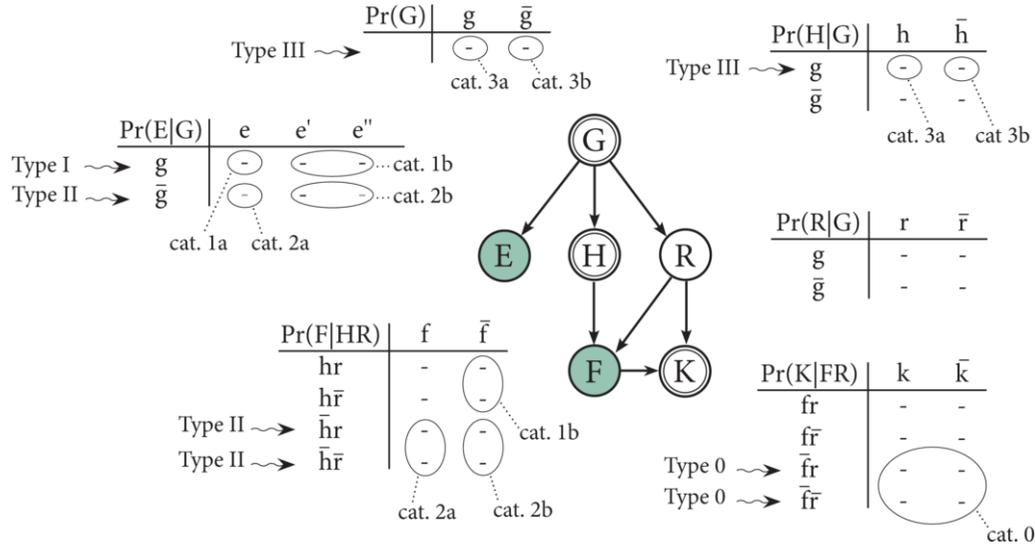


Figure 1: An example Bayesian—and credal—network. The CPTs are labelled with parameter categories (see Table 1) and distribution types (see Section 3.3) with respect to output probability $\Pr(ghk|ef)$.

Table 1: Categorisation of parameters $x = \Pr(v|\pi)$ with respect to the outcome probability $\Pr(\mathbf{h}|\mathbf{e})$.

	$\pi \sim \mathbf{h}$	$\pi \approx \mathbf{h}$
$\pi \sim \mathbf{e}$	cat. 0	cat. 0
$V \in \mathbf{E}, v \approx \mathbf{e}$ and $\pi \sim \mathbf{e}$	cat. 1b	cat. 2b
$V \in \mathbf{E}_{\mathcal{R}}$ and $v \pi \sim \mathbf{e}$	cat. 1a	cat. 2a
$V \in \mathbf{E} \setminus \mathbf{E}_{\mathcal{R}}$ and $v \pi \sim \mathbf{e}$	-	cat. 2a
binary $V \in \mathbf{H}_{\mathcal{R}}$ and $\pi \sim \mathbf{e}$	cat. 3a ($v \sim \mathbf{h}$) cat. 3b ($v \approx \mathbf{h}$)	-

The following proposition states that parameters from category 1a result in sensitivity functions that are monotonically increasing, whereas parameters from category 2a give monotonically decreasing sensitivity functions, *irrespective of the values of the non-varied network parameters and irrespective of the co-variation scheme that is used.*

Proposition 1 Consider a Bayesian network \mathcal{B} and a probability of interest $\Pr(\mathbf{h}|\mathbf{e})$. Let $x = \Pr(v|\pi)$ be a parameter of $V \in \mathbf{E}$ with $v \pi \sim \mathbf{e}$. If $V \in \mathbf{E}_{\mathcal{R}}$ and $\pi \sim \mathbf{h}$, then the sensitivity function $\Pr(\mathbf{h}|\mathbf{e})(x)$ is monotonically increasing. If $\pi \approx \mathbf{h}$, then the sensitivity function $\Pr(\mathbf{h}|\mathbf{e})(x)$ is monotonically decreasing.

Proof. We first consider the numerator of the sensitivity function:

$$\Pr(\mathbf{h}\mathbf{e}) = \sum_{\mathbf{r} \in \mathbf{R}} \Pr(\mathbf{r}\mathbf{h}\mathbf{e}) = \sum_{\mathbf{r} \in \mathbf{R}} \prod_{V^* \in \mathbf{V}} \Pr(v^*|\pi^*)|_{v^*\pi^* \sim \mathbf{r}\mathbf{h}\mathbf{e}} \quad (1)$$

Note that $x = \Pr(v|\pi)$ is a factor in this expression only if $\pi \sim \mathbf{h}$. For $\pi \approx \mathbf{h}$, therefore, $\Pr(\mathbf{h}\mathbf{e})(x) = \tau_1$ for some constant $\tau_1 \geq 0$. Returning to the case where $\pi \sim \mathbf{h}$, we observe that if $V \in \mathbf{E}_{\mathcal{R}}$, we have that $\pi \sim \mathbf{r}\mathbf{h}\mathbf{e}$ for all $\mathbf{r} \in \mathbf{R}$ and we thus find that

$$\Pr(\mathbf{h}\mathbf{e})(x) = x \cdot \left(\sum_{\mathbf{r} \in \mathbf{R}} \prod_{V^* \in \mathbf{V} \setminus \{V\}} \Pr(v^*|\pi^*)|_{v^*\pi^* \sim \mathbf{r}\mathbf{h}\mathbf{e}} \right)$$

Given $\pi \sim \mathbf{h}$ and $V \in \mathbf{E}_{\mathcal{R}}$, therefore, $\Pr(\mathbf{h}\mathbf{e})(x) = x \cdot \tau_1$ for some constant $\tau_1 \geq 0$.

Next, we consider the denominator of the sensitivity function:

$$\begin{aligned} \Pr(\mathbf{e}) &= \sum_{\mathbf{r} \in \mathbf{R}, \mathbf{h}^* \in \mathbf{H}} \Pr(\mathbf{r}\mathbf{h}^*\mathbf{e}) \\ &= \sum_{\mathbf{r} \in \mathbf{R}, \mathbf{h}^* \in \mathbf{H}} \prod_{V^* \in \mathbf{V}} \Pr(v^*|\pi^*)|_{v^*\pi^* \sim \mathbf{r}\mathbf{h}^*\mathbf{e}} \quad (2) \end{aligned}$$

We observe that $x = \Pr(v|\pi)$ is a factor in each term of the summation for which $\pi \sim \mathbf{r}\mathbf{h}^*$, and otherwise is absent. That is, no term includes x as a co-varying parameter. Therefore, $\Pr(\mathbf{e})(x) = \kappa_1 \cdot x + \kappa_2$, for constants $\kappa_1, \kappa_2 \geq 0$.

We are now able to determine the general form of the sensitivity function $\Pr(\mathbf{h}|\mathbf{e})(x)$ under the given conditions. Moreover, we can use the first derivative of this sensitivity function to determine whether the function is increasing or decreasing. If $\pi \sim \mathbf{h}$ and $V \in \mathbf{E}_{\mathcal{R}}$, we find that

$$\Pr(\mathbf{h}|\mathbf{e})'(x) = \frac{\tau_1 \cdot \kappa_2}{(\kappa_1 \cdot x + \kappa_2)^2}.$$

This first derivative is always non-negative which implies that $\Pr(\mathbf{h}|\mathbf{e})(x)$ is a monotone increasing function, irrespective of the values of the non-varied network parameters and irrespective of the co-variation scheme. If $\pi \approx \mathbf{h}$, we find that

$$\Pr(\mathbf{h}|\mathbf{e})'(x) = \frac{-\tau_1 \cdot \kappa_1}{(\kappa_1 \cdot x + \kappa_2)^2}.$$

This first derivative is always non-positive, which implies that $\Pr(\mathbf{h}|\mathbf{e})(x)$ is a monotone decreasing function, irrespective of the values of the non-varied network parameters and irrespective of the co-variation scheme. \square

Our next proposition states that parameters from category 3a result in monotonically increasing sensitivity functions, whereas parameters from category 3b give monotonically decreasing sensitivity functions, again *irrespective of the values of the non-varied network*

parameters. The proposition concerns parameters for binary hypothesis variables, which implies that the co-variation scheme is fixed.

Proposition 2 Consider a Bayesian network \mathcal{B} and a probability of interest $\Pr(\mathbf{h}|\mathbf{e})$. Let $x = \Pr(v|\boldsymbol{\pi})$ be a parameter of a binary variable $V \in \mathbf{H}_{\mathcal{R}}$, with $\boldsymbol{\pi} \sim \mathbf{e}$ and $\boldsymbol{\pi} \sim \mathbf{h}$. If $v \sim \mathbf{h}$ then the sensitivity function $\Pr(\mathbf{h}|\mathbf{e})(x)$ is monotonically increasing. If $v \sim \mathbf{e}$ then $\Pr(\mathbf{h}|\mathbf{e})(x)$ is monotonically decreasing.

Proof. We first consider the parameters x with $v \sim \mathbf{h}$. As in the proof of Proposition 1, the numerator of the sensitivity function can be rewritten as

$$\Pr(\mathbf{h}\mathbf{e})(x) = x \cdot \left(\sum_{\mathbf{r} \in \mathbf{R}} \prod_{V^* \in \mathbf{V} \setminus \{H\}} \Pr(v^*|\boldsymbol{\pi}^*) \Big|_{v^* \boldsymbol{\pi}^* \sim \mathbf{r}\mathbf{h}\mathbf{e}} \right)$$

because, under the given conditions, x is a factor in all terms of the summation. Therefore, we have that $\Pr(\mathbf{h}\mathbf{e})(x) = x \cdot \tau_1$, for some constant $\tau_1 \geq 0$.

The assumption that V is binary enforces the following co-variation scheme: $\Pr(\bar{v}|\boldsymbol{\pi}) = 1 - \Pr(v|\boldsymbol{\pi})$. Therefore, with respect to the denominator of the sensitivity function—see Equation (2)—we observe the following: 1) x itself is a factor in any term of the summation for which $\boldsymbol{\pi} \sim \mathbf{h}^*$ and $v \sim \mathbf{h}^*$, 2) x appears as $(1 - x)$ for $\boldsymbol{\pi} \sim \mathbf{e}^*$ and $v \sim \mathbf{e}^*$, and 3) x is absent whenever $\boldsymbol{\pi} \sim \mathbf{h}^*$. Hence, $\Pr(\mathbf{e})(x) = x \cdot \kappa_1 + (1 - x) \cdot \kappa_2 + \kappa_3$, with constants $\kappa_i \geq 0$, $i = 1, 2, 3$.

We now find that the first derivative of the sensitivity function equals the following non-negative function:

$$\Pr(\mathbf{h}|\mathbf{e})'(x) = \frac{(\kappa_2 + \kappa_3) \cdot \tau_1}{(\kappa_2 + \kappa_3 + \kappa_1 \cdot x - \kappa_2 \cdot x)^2}$$

This implies that $\Pr(\mathbf{h}|\mathbf{e})(x)$ is a monotonically increasing function irrespective of the values of the non-varied network parameters.

Next, we consider the parameters $x = \Pr(v|\boldsymbol{\pi})$ with $v \sim \mathbf{e}$. Since we know that $\Pr(v|\boldsymbol{\pi}) = 1 - \Pr(\bar{v}|\boldsymbol{\pi})$ with $\bar{v} \sim \mathbf{h}$, an increase (decrease) of x is equivalent to a decrease (increase) of the parameter consistent with \mathbf{h} . Hence, using the first part of this proof, it follows that $\Pr(\mathbf{h}|\mathbf{e})(x)$ is a monotonically decreasing function irrespective of the values of the non-varied network parameters. \square

Although the above propositions are concerned with one-way sensitivity functions, we have shown that their monotone increasing or decreasing behaviour does not depend on the actual values of the non-varied parameters which determine the constants of the function. As a result, our observations extend to specific n -way sensitivity functions.

Theorem 1 Consider a Bayesian network \mathcal{B} and a probability of interest $\Pr(\mathbf{h}|\mathbf{e})$. Let $\vec{x}_+ = \{x_1^+, \dots, x_k^+\}$ be a set of parameters from categories 1a and/or 3a, and let $\vec{x}_- = \{x_1^-, \dots, x_l^-\}$ be a set of parameters from categories 2a and/or 3b. Then the $(k + l)$ -way sensitivity function $\Pr(\mathbf{h}|\mathbf{e})(\vec{x}_+, \vec{x}_-)$ is monotonically increasing for increasing parameters x_i^+ and decreasing parameters x_i^- ; the sensitivity function is monotonically decreasing for decreasing x_i^+ and increasing x_i^- .

Proof. From Propositions 1 and 2 we have that a one-way sensitivity function $\Pr(\mathbf{h}|\mathbf{e})(x_i^+)$ increases with increasing x_i^+ and a one-way sensitivity function $\Pr(\mathbf{h}|\mathbf{e})(x_i^-)$ increases with decreasing x_i^- , irrespective of the values of the non-varied parameters, and irrespective of the co-variation scheme. Given a simultaneous increase of

parameters in \vec{x}_+ and decrease of parameters in \vec{x}_- , therefore, the probability $\Pr(\mathbf{h}|\mathbf{e})(\vec{x}_+, \vec{x}_-)$ will increase. A similar observation holds for decreasing x_i^+ and increasing x_i^- . \square

The above results describe an output probability of interest as a function of changing input parameters of categories 1a, 2a, 3a and 3b. The example network in Figure 1 also includes parameters of category 0, 1b and 2b. Parameters of these categories are not used in computing the output probability. For category 0, the parameters make up an entire local distribution (see e.g. the CPT for $\Pr(K|FR)$); for parameters of categories 1b and 2b, parameters are in a local distribution with a parameter of category 1a or 2a (see e.g. the CPTs for $\Pr(E|G)$ and $\Pr(F|HR)$).

Proposition 3 Consider a Bayesian network \mathcal{B} and a probability of interest $\Pr(\mathbf{h}|\mathbf{e})$. Let $\Pr(v|\boldsymbol{\pi})$ be a parameter. If $v \sim \mathbf{e}$ and/or $\boldsymbol{\pi} \sim \mathbf{e}$, then $\Pr(v|\boldsymbol{\pi})$ is not used in the computation of $\Pr(\mathbf{h}|\mathbf{e})$.

Proof. As before, we have that

$$\Pr(\mathbf{h}|\mathbf{e}) = \frac{\Pr(\mathbf{h}\mathbf{e})}{\Pr(\mathbf{e})}$$

with numerator

$$\Pr(\mathbf{h}\mathbf{e}) = \sum_{\mathbf{r} \in \mathbf{R}} \Pr(\mathbf{r}\mathbf{h}\mathbf{e}) = \sum_{\mathbf{r} \in \mathbf{R}} \prod_{V^* \in \mathbf{V}} \Pr(v^*|\boldsymbol{\pi}^*) \Big|_{v^* \boldsymbol{\pi}^* \sim \mathbf{r}\mathbf{h}\mathbf{e}}$$

The parameter $\Pr(v|\boldsymbol{\pi})$ can only be a factor in this numerator if $v \sim \mathbf{e}$ and $\boldsymbol{\pi} \sim \mathbf{e}$. A completely analogous observation holds for the denominator as well. \square

Note that we cannot conclude from the above proposition that the sensitivity function $\Pr(\mathbf{h}|\mathbf{e})(x)$ for parameters $x = \Pr(v|\boldsymbol{\pi})$ of categories 1b and 2b is constant. A change of such parameters may, by co-variation, induce a change in the parameter $\Pr(v'|\boldsymbol{\pi})$ of the same local distribution with $v' \sim \mathbf{e}$ which is used in the computation of $\Pr(\mathbf{h}|\mathbf{e})$. The actual value of a parameter with $v \sim \mathbf{e}$, however, is irrelevant for $\Pr(\mathbf{h}|\mathbf{e})$. We can conclude that the sensitivity function $\Pr(\mathbf{h}|\mathbf{e})(x)$ for parameters of category 0 is a constant, however, since given a parameter with $\boldsymbol{\pi} \sim \mathbf{e}$ this condition is fulfilled for all parameters in the same local distribution.

3.3 Defining local distribution types

In the previous subsection we have focused on properties of different categories of parameters in a Bayesian network. In order to apply our results in the context of credal networks, it will be useful to lift these properties to the level of distributions. To this end, we define four types of local distributions $\Pr(V|\boldsymbol{\pi})$, again with respect to a specific outcome probability of interest $\Pr(\mathbf{h}|\mathbf{e})$.

Definition 1 (Local distribution of Type 0) Consider a Bayesian network \mathcal{B} and a probability of interest $\Pr(\mathbf{h}|\mathbf{e})$. A local distribution $\Pr(V|\boldsymbol{\pi})$ is said to be of Type 0 if $\boldsymbol{\pi} \sim \mathbf{e}$.

All parameters of a local distribution of Type 0 are in category 0. From Proposition 3, we know that parameters from a local distribution of Type 0 are irrelevant for the computation of the outcome probability $\Pr(\mathbf{h}|\mathbf{e})$.

Definition 2 (Local distribution of Type I) Consider a Bayesian network \mathcal{B} and a probability of interest $\Pr(\mathbf{h}|\mathbf{e})$. A local distribution $\Pr(V|\boldsymbol{\pi})$ is said to be of Type I if $\boldsymbol{\pi} \sim \mathbf{e}$, $V \in \mathbf{E}_{\mathcal{R}}$ and $\boldsymbol{\pi} \sim \mathbf{h}$.

The unique parameter of a distribution of Type I with $v \sim e$ is in category 1a. Any other parameter $\Pr(v' | e)$ of the distribution is in category 1b and is therefore irrelevant for computing $\Pr(\mathbf{h} | e)$, since $v' \approx e$. Given changes in a local distribution of Type I, therefore, knowing the qualitative change of the parameter with $v \sim e$ suffices to know the qualitative effect of the changes in the entire distribution on $\Pr(\mathbf{h} | e)$. In particular, an increase (decrease) of this parameter results in an increase (decrease) of $\Pr(\mathbf{h} | e)$.

Definition 3 (Local distribution of Type II) Consider a Bayesian network \mathcal{B} and a probability of interest $\Pr(\mathbf{h} | e)$. A local distribution $\Pr(V | \pi)$ is said to be of Type II if $\pi \sim e$, $V \in \mathbf{E}$ and $\pi \approx \mathbf{h}$.

Analogous to distributions of Type I, knowing the qualitative change of the parameter with $v \sim e$ is sufficient to know the qualitative effect of changes in the entire local distribution on $\Pr(\mathbf{h} | e)$. Now the unique parameter with $v \sim e$ is in category 2a, which implies that an increase (decrease) of this parameter results in an decrease (increase) of $\Pr(\mathbf{h} | e)$.

Definition 4 (Local distribution of Type III) Consider a Bayesian network \mathcal{B} and a probability of interest $\Pr(\mathbf{h} | e)$. A local distribution $\Pr(V | \pi)$ is said to be of Type III if $\pi \sim e$, $\pi \sim \mathbf{h}$, $V \in \mathbf{H}_{\mathcal{R}}$ and $|V| = 2$.

Distributions of Type III consist of exactly two parameters, one for v and one for \bar{v} , which are related by the fixed co-variation scheme: $\Pr(v | \pi) = 1 - \Pr(\bar{v} | \pi)$. The parameter that is compatible with \mathbf{h} is in category 3a, which implies that an increase (decrease) of this parameter results in an increase (decrease) of $\Pr(\mathbf{h} | e)$. The parameter that is not compatible with \mathbf{h} is in category 3b, which implies that the concurrent decrease (increase) of this parameter results in an increase (decrease) of $\Pr(\mathbf{h} | e)$ as well.

By Theorem 1 and Proposition 3 we have that, given simultaneous changes in multiple local distributions, $\Pr(\mathbf{h} | e)$ will increase (decrease) given an increase (decrease) of the parameters with $v \sim e$ in the distributions of Type I and III and a decrease (increase) of the parameters compatible with $v \sim e$ in the distributions of Type II. The outcome probability $\Pr(\mathbf{h} | e)$ is not affected by changes in local distributions of Type 0.

3.4 Important special cases

In Section 3.3 we identified types of distributions for which we can guarantee the direction of change in $\Pr(\mathbf{h} | e)$ upon changes in certain parameters from those distributions. In general, as illustrated in Figure 1, a network will include distributions that do not belong to any of those types. We can, however, identify classes of networks for which the majority of distributions—or even all of them—can be classified as Type 0, I, II or III.

Consider for example a network obeying the following constraint: each observed variable only has hypothesis variables or other observed variables as parents, that is, $\mathbf{E} = \mathbf{E}_{\mathcal{R}}$. It then follows from Definitions 1–3 that all local distributions of the observed variables are of type 0, I or II. Bayesian network classifiers with full evidence obey this constraint. Bayesian classifiers, a special type of Bayesian network, are widely used in practice (see [14] for an overview).

Now consider a network obeying the constraint that all hypothesis variables are binary valued and can only have observed variables as parents, that is, $\mathbf{H} = \mathbf{H}_{\mathcal{R}} \cap \mathbf{H}_{\mathcal{R}}$ and $|V| = 2$ for each V in \mathbf{H} . It then follows from Definitions 1 and 4 that all local distributions of

the hypothesis variables are of Type 0 or III.⁶ This second constraint is for example obeyed by Bayesian network classifiers with a single binary class variable, or by Bayesian network classifiers with multiple binary class variables, provided that these class variables have no class parents.⁷ Given full evidence, these two types of Bayesian network classifiers also satisfy the first constraint, it follows therefore that, given full evidence, for those networks all the distributions are of Type 0, I, II or III.

4 AN APPLICATION TO CREDAL NETWORKS

Since a sensitivity function is an object that is associated with a given Bayesian network, it might at first sight seem as if it has little to do with credal networks, because these are essentially sets of Bayesian networks. However, for the particular sensitivity properties that we developed in the previous section, this is not the case. As we will explain in this section, these sensitivity properties allow us to replace some of the local credal sets of a credal network by a single probability mass function, and in this way, the combinatoric nature of credal network inference can be reduced.

4.1 Credal network preliminaries

An important limitation of Bayesian networks is that they require the exact specification of a large number of probabilities. Since these probabilities have to be elicited from experts or estimated from data, this requirement is often unrealistic. By enforcing it anyway, as is conventionally done, we run the risk of ending up with a model whose precision is unwarranted, thereby making the resulting conclusions unreliable. In order to avoid these issues, credal networks explicitly drop the assumption that probabilities should be specified exactly, and instead adopt the framework of imprecise probabilities [2, 23, 25], thereby trading off precision for reliability.

Basically, a *credal network* [1, 11, 12] is just a Bayesian network of which the local models are partially specified, in the sense that they are only known to belong to some set of candidate distributions. Specifically, for every random variable $V \in \mathbf{V}$ and every instantiation π of its parents π , the local distribution $\Pr(V | \pi)$ is replaced by a non-empty set $\mathcal{M}(V | \pi)$ of candidate distributions. This set is usually taken to be closed and convex, and is then called a *credal set*. The local credal sets of a credal network can be obtained in various ways [23], the most typical of which are to elicit them from experts [20], to learn them from data [4, 24] or to create a neighbourhood model around some distribution [3].

For example, in Figure 1, if an expert judges that the probability of g is at most $2/3$ but not less than $1/3$, then in particular, we are lead to consider the local credal set $\mathcal{M}(G)$ that is defined by

$$\Pr(G) \in \mathcal{M}(G) \Leftrightarrow \frac{1}{3} \leq \Pr(g) \leq \frac{2}{3}. \quad (3)$$

The same credal set can also be obtained by ϵ -contaminating the uniform probability distribution on $\{g, \bar{g}\}$ with $\epsilon = 1/3$ [3] or by learning it from a dataset in which each of the two possible outcomes— g and \bar{g} —is counted twice, using the Imprecise Dirichlet Model (IDM) with $s = 2$ [4, 24]. As another example, consider a situation in which, conditional on $G = g$, an expert judges that e is at least as

⁶ $V \in \mathbf{H}_{\mathcal{R}}$ implies that $\pi \sim \mathbf{h}$

⁷ Note that, by definition, in Bayesian classifiers class variables do not have feature parents, as a consequence, $\mathbf{H} = \mathbf{H}_{\mathcal{R}}$, even in case of incomplete evidence.

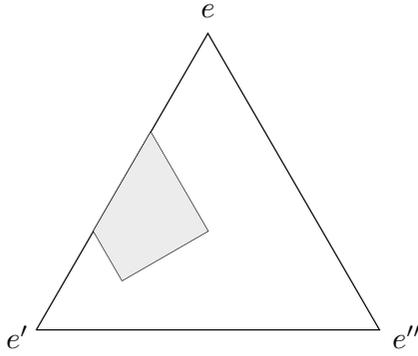


Figure 2: Every point in the above equilateral triangle of height one can be identified with a distribution $\Pr(E|g)$ on $\{e, e', e''\}$, by letting $\Pr(e|g)$ be equal to the perpendicular distance from that point to the edge that opposes the corner that corresponds to e , and similarly for $\Pr(e'|g)$ and $\Pr(e''|g)$. In this way, the grey area depicts the distributions in the credal set $\mathcal{M}(E|g)$ of Equation (4).

probable as e'' and that the probability of e' is at most $2/3$ and at least $1/3$. In that case, we are lead to consider the credal set $\mathcal{M}(E|g)$ that is defined by

$$\Pr(E|g) \in \mathcal{M}(E|g) \Leftrightarrow \Pr(e|g) \geq \Pr(e''|g) \text{ and } \frac{1}{3} \leq \Pr(e'|g) \leq \frac{2}{3}, \quad (4)$$

as depicted in Figure 2. The other local credal sets can be obtained similarly. Depending on the amount of data and/or expert assessments that is available, some of these local credal sets—such as the two examples we have just provided—will be *imprecise*, meaning that they consist of multiple distributions, while others may be *precise*, meaning that they consist of a single distribution.

If all the local credal sets are precise, a credal network is exactly the same as a Bayesian network. However, if some of the credal sets $\mathcal{M}(V|\pi)$ are imprecise, then since the local distributions $\Pr(V|\pi)$ are only known to belong to these credal sets, they do not determine a unique Bayesian network. Therefore, in general, a credal network corresponds to a set of Bayesian networks \mathbb{B} , defined by⁸

$$\mathcal{B} \in \mathbb{B} \Leftrightarrow (\forall V \in \mathbf{V}) (\forall \pi \in \Pi) \Pr(V|\pi) \in \mathcal{M}(V|\pi).$$

If all the local credal sets are precise, the set \mathbb{B} consists of a single Bayesian network.

An important computational problem in credal networks consists in computing tight lower and upper bounds on the probabilities that correspond to the Bayesian networks \mathcal{B} in \mathbb{B} , which are called lower and upper probabilities. In particular, we consider the problem of computing tight lower and upper bounds on probabilities of the form $\Pr(\mathbf{h}|\mathbf{e})$, defined by

$$\underline{\Pr}(\mathbf{h}|\mathbf{e}) := \min_{\mathcal{B} \in \mathbb{B}} \Pr(\mathbf{h}|\mathbf{e}) \text{ and } \overline{\Pr}(\mathbf{h}|\mathbf{e}) := \max_{\mathcal{B} \in \mathbb{B}} \Pr(\mathbf{h}|\mathbf{e}). \quad (5)$$

⁸ There are also other ways in which the local credal sets of a credal network can be combined to form a global model, depending on the type of credal network that is considered [11]; our definition corresponds to what is called a credal network under complete independence. Our results also apply to credal networks under strong independence, which replace \mathbb{B} by its convex hull, but not to credal networks under epistemic irrelevance [5, 7, 9] or epistemic independence [8].

In order for these definitions to make sense, we require that $\Pr(\mathbf{e})$ is strictly positive for all $\mathcal{B} \in \mathbb{B}$.⁹

Computing the lower and upper probabilities in Equation (5) consists in optimising $\Pr(\mathbf{h}|\mathbf{e})$, which is a fraction of two multilinear functions of the network’s parameters, under the constraint that each of the local distributions $\Pr(V|\pi)$ belongs to its credal set $\mathcal{M}(V|\pi)$. Obtaining the exact solution is only possible in special cases [13, 17] or for networks that are sufficiently small. Therefore, applied work on credal networks often resorts to approximate algorithms; see Reference [1] for a recent overview.

Most of the existing algorithms exploit the fact that for the purposes of computing $\underline{\Pr}(\mathbf{h}|\mathbf{e})$ or $\overline{\Pr}(\mathbf{h}|\mathbf{e})$, the—closed and convex—credal sets $\mathcal{M}(V|\pi)$ can be replaced by their set of extreme points¹⁰ $\text{ext}(\mathcal{M}(V|\pi))$ [13]:

$$\underline{\Pr}(\mathbf{h}|\mathbf{e}) = \min_{\mathcal{B} \in \mathbb{B}_{\text{ext}}} \Pr(\mathbf{h}|\mathbf{e}) \text{ and } \overline{\Pr}(\mathbf{h}|\mathbf{e}) = \max_{\mathcal{B} \in \mathbb{B}_{\text{ext}}} \Pr(\mathbf{h}|\mathbf{e}),$$

where \mathbb{B}_{ext} is the set of Bayesian networks in \mathbb{B} whose local distributions $\Pr(V|\pi)$ take values in $\text{ext}(\mathcal{M}(V|\pi))$, defined by

$$\mathcal{B} \in \mathbb{B}_{\text{ext}} \Leftrightarrow (\forall V \in \mathbf{V}) (\forall \pi \in \Pi) \Pr(V|\pi) \in \text{ext}(\mathcal{M}(V|\pi)).$$

The reason why this is useful is because in practice, credal sets are usually *finitely generated*, meaning that they are defined by means of a finite number of linear constraints, or equivalently, that they have a finite number of extreme points. For example, the credal set $\mathcal{M}(G)$ of Equation (3) has two extreme points, which are characterised by $\Pr(g) = 1/3$ and $\Pr(g) = 2/3$, respectively. Similarly, the credal set $\mathcal{M}(E|g)$ in Figure 2 has four extreme points $\Pr(E|g) = (\Pr(e|g), \Pr(e'|g), \Pr(e''|g))$:

$$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), \left(\frac{2}{3}, \frac{1}{3}, 0\right), \left(\frac{1}{3}, \frac{2}{3}, 0\right) \text{ and } \left(\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\right). \quad (6)$$

If all the local credal sets are finitely generated, \mathbb{B}_{ext} is finite, and computing lower and upper probabilities is then a combinatoric optimisation problem, the size of which is exponential in the number of local credal sets that are imprecise.

4.2 Reducing imprecise credal sets to precise ones

As we will now show, in many cases, we can use the sensitivity properties of Section 3.2 to reduce the computational complexity of inference in credal networks.

Basically, the idea is that, for the purposes of computing $\underline{\Pr}(\mathbf{h}|\mathbf{e})$ or $\overline{\Pr}(\mathbf{h}|\mathbf{e})$, some of the local credal sets of a credal network can be replaced by a singleton, that is, some imprecise local credal sets can be replaced by precise ones, without changing the result of the inference. In this way, the size of the combinatoric optimisation problem that needs to be solved can be reduced in advance, as a preprocessing step, before applying an inference algorithm of choice.

We introduce four types of local credal sets for which this is the case. Their definitions are completely analogous to the local distribution types that were defined in Section 3.3. For example, a local credal set $\mathcal{M}(V|\pi)$ is of Type 0 if and only if its distributions are of Type 0, and similarly for credal sets that are of type I, II or III.

⁹ For a detailed discussion on how to define $\underline{\Pr}(\mathbf{h}|\mathbf{e})$ and $\overline{\Pr}(\mathbf{h}|\mathbf{e})$ if $\Pr(\mathbf{e})$ is zero for some—but not all— \mathcal{B} in \mathbb{B} , see Reference [6].

¹⁰ An extreme point of a set is a point that cannot be written as a proper convex combination of two other points of that set. If a set is a closed and convex subset of a finite-dimensional space, then by Minkowski’s finite-dimensional version of the Krein-Milman theorem, it is the convex hull of its extreme points.

Definition 5 (Local credal set types) Consider a credal network \mathbb{B} and a lower or upper probability of interest $\underline{\Pr}(\mathbf{h}|\mathbf{e})$ or $\overline{\Pr}(\mathbf{h}|\mathbf{e})$. A local credal set $\mathcal{M}(V|\boldsymbol{\pi})$ is then said to be of

- Type 0** if $\boldsymbol{\pi} \approx \mathbf{e}$;
- Type I** if $\boldsymbol{\pi} \sim \mathbf{e}$, $V \in \mathbf{E}_{\setminus \mathcal{R}}$ and $\boldsymbol{\pi} \sim \mathbf{h}$;
- Type II** if $\boldsymbol{\pi} \sim \mathbf{e}$, $V \in \mathbf{E}$ and $\boldsymbol{\pi} \approx \mathbf{h}$;
- Type III** if $\boldsymbol{\pi} \sim \mathbf{e}$, $\boldsymbol{\pi} \sim \mathbf{h}$, $V \in \mathbf{H}_{\setminus \mathcal{R}}$ and $|V| = 2$.

Since the elements of a credal set $\mathcal{M}(V|\boldsymbol{\pi})$ of Type 0 are distributions of Type 0, it follows from Proposition 3 that this credal set is of no influence to the inference at hand. The following result is an immediate consequence of this fact.

Proposition 4 Consider a credal network \mathbb{B} and a lower or upper probability of interest $\underline{\Pr}(\mathbf{h}|\mathbf{e})$ or $\overline{\Pr}(\mathbf{h}|\mathbf{e})$. Then replacing a credal set $\mathcal{M}(V|\boldsymbol{\pi})$ of Type 0 with a singleton $\mathcal{M}'(V|\boldsymbol{\pi}) := \{\Pr(V|\boldsymbol{\pi})\}$ that consists of an arbitrary probability mass function $\Pr(V|\boldsymbol{\pi})$ on V will not change the result of the inference.

Note that $\Pr(V|\boldsymbol{\pi})$ is not required to be an element of $\mathcal{M}(V|\boldsymbol{\pi})$, and that it can therefore be chosen as simple as possible.

As we are about to show, similar results can also be obtained for credal sets of Type I, II and III. However, in those cases, the distribution $\Pr(V|\boldsymbol{\pi})$ can no longer be chosen arbitrarily. We start with credal sets of Type I or II.

Since a distribution in a credal set of Type I or II is itself of Type I or II, we infer from the discussion in Section 3.3 that, for the purposes of computing $\Pr(\mathbf{h}|\mathbf{e})$, the only part of this distribution that is relevant is the parameter $\Pr(v|\boldsymbol{\pi})$, with v the unique instantiation of V that is compatible with \mathbf{e} . Furthermore, because of the monotone effect of changing this parameter—see Theorem 1—we know that the minimum and maximum value of $\Pr(\mathbf{h}|\mathbf{e})$ will be obtained by a Bayesian network $\mathcal{B} \in \mathbb{B}_{\text{ext}}$ for which the parameter $\Pr(v|\boldsymbol{\pi})$ is minimal or maximal, in the sense that it is equal to the local lower probability

$$\underline{\Pr}(v|\boldsymbol{\pi}) := \min_{\Pr(V|\boldsymbol{\pi}) \in \mathcal{M}(V|\boldsymbol{\pi})} \Pr(v|\boldsymbol{\pi})$$

or the local upper probability

$$\overline{\Pr}(v|\boldsymbol{\pi}) := \max_{\Pr(V|\boldsymbol{\pi}) \in \mathcal{M}(V|\boldsymbol{\pi})} \Pr(v|\boldsymbol{\pi}),$$

and we can predict in advance which of these two extreme values it will be. Combined, these two observations allow us to replace the (possibly imprecise) credal set $\mathcal{M}(V|\boldsymbol{\pi})$ with a precise one. The following proposition makes this explicit.

Proposition 5 Consider a credal network \mathbb{B} and a lower or upper probability of interest $\underline{\Pr}(\mathbf{h}|\mathbf{e})$ or $\overline{\Pr}(\mathbf{h}|\mathbf{e})$. A credal set $\mathcal{M}(V|\boldsymbol{\pi})$ of Type I or II can then be replaced with a singleton $\mathcal{M}'(V|\boldsymbol{\pi}) := \{\Pr(V|\boldsymbol{\pi})\}$, without changing the result of the inference. If we let v be the unique value of V that is compatible with \mathbf{e} , then for $\underline{\Pr}(\mathbf{h}|\mathbf{e})$, $\Pr(V|\boldsymbol{\pi})$ can be any probability mass function on V such that

$$\Pr(v|\boldsymbol{\pi}) = \begin{cases} \underline{\Pr}(v|\boldsymbol{\pi}) & \text{if } \mathcal{M}(V|\boldsymbol{\pi}) \text{ is of Type I;} \\ \overline{\Pr}(v|\boldsymbol{\pi}) & \text{if } \mathcal{M}(V|\boldsymbol{\pi}) \text{ is of Type II,} \end{cases}$$

and for $\overline{\Pr}(\mathbf{h}|\mathbf{e})$, $\Pr(V|\boldsymbol{\pi})$ can be any probability mass function on V such that

$$\Pr(v|\boldsymbol{\pi}) = \begin{cases} \overline{\Pr}(v|\boldsymbol{\pi}) & \text{if } \mathcal{M}(V|\boldsymbol{\pi}) \text{ is of Type I;} \\ \underline{\Pr}(v|\boldsymbol{\pi}) & \text{if } \mathcal{M}(V|\boldsymbol{\pi}) \text{ is of Type II.} \end{cases}$$

Proof. We only prove the result for $\underline{\Pr}(\mathbf{h}|\mathbf{e})$; the proof for $\overline{\Pr}(\mathbf{h}|\mathbf{e})$ is completely analogous.

If the local credal set $\mathcal{M}(V|\boldsymbol{\pi})$ is of Type I, then because of Theorem 1, the minimum value of $\Pr(\mathbf{h}|\mathbf{e})$ is obtained for a local distribution $\Pr(V|\boldsymbol{\pi})$ of which the parameter $\Pr(v|\boldsymbol{\pi})$ is minimal—equal to $\underline{\Pr}(v|\boldsymbol{\pi})$. Furthermore, due to Proposition 3, the other parameters of this local distribution are irrelevant. Hence, for the purposes of computing $\underline{\Pr}(\mathbf{h}|\mathbf{e})$, $\mathcal{M}(V|\boldsymbol{\pi})$ can be replaced by a precise credal set $\mathcal{M}'(V|\boldsymbol{\pi}) := \{\Pr(V|\boldsymbol{\pi})\}$ that consists of a single distribution $\Pr(V|\boldsymbol{\pi})$ on V , with $\Pr(v|\boldsymbol{\pi}) = \underline{\Pr}(v|\boldsymbol{\pi})$. If the local credal set $\mathcal{M}(V|\boldsymbol{\pi})$ is of Type II, it follows from Theorem 1 that the minimum value of $\Pr(\mathbf{h}|\mathbf{e})$ is obtained for a local distribution $\Pr(V|\boldsymbol{\pi})$ of which the parameter $\Pr(v|\boldsymbol{\pi})$ is maximal—equal to $\overline{\Pr}(v|\boldsymbol{\pi})$ —and therefore, $\mathcal{M}(V|\boldsymbol{\pi})$ can again be replaced by a singleton that consists of a single distribution $\Pr(V|\boldsymbol{\pi})$, now with $\Pr(v|\boldsymbol{\pi}) = \overline{\Pr}(v|\boldsymbol{\pi})$. \square

Note that $\Pr(V|\boldsymbol{\pi})$ is again not required to be an element of $\mathcal{M}(V|\boldsymbol{\pi})$, which allows us to choose it as simple as possible.

Consider for example the network in Figure 1, and assume that we want to compute $\underline{\Pr}(ghk|ef)$. The local credal set $\mathcal{M}(E|g)$ —defined in Equation (4) and depicted in Figure 2—is then of Type I and can therefore be replaced by a singleton $\mathcal{M}'(E|g) = \{\Pr(E|g)\}$, where $\Pr(E|g)$ can be any probability mass function on $\{e, e', e''\}$ such that

$$\Pr(e|g) := \underline{\Pr}(e|g) = \frac{1}{6}.$$

The values of $\Pr(e'|g)$ and $\Pr(e''|g)$ are of no importance. If we want $\Pr(E|g)$ to belong to $\mathcal{M}(E|g)$, then the only possible choice is $\Pr(e'|g) = 2/3$ and $\Pr(e''|g) = 1/6$, which corresponds to one of the extreme points in Equation (6). However, this is not necessary. For example, choosing $\Pr(e'|g) = 0$ and $\Pr(e''|g) = 5/6$ works equally well.

Finally, for local credal sets that are of Type III, we obtain a result that is very similar to that of Proposition 5.

Proposition 6 Consider a credal network \mathbb{B} and a lower or upper probability of interest $\underline{\Pr}(\mathbf{h}|\mathbf{e})$ or $\overline{\Pr}(\mathbf{h}|\mathbf{e})$. A credal set $\mathcal{M}(V|\boldsymbol{\pi})$ of Type III can then be replaced with a singleton $\mathcal{M}'(V|\boldsymbol{\pi}) := \{\Pr(V|\boldsymbol{\pi})\}$, without changing the result of the inference. For $\underline{\Pr}(\mathbf{h}|\mathbf{e})$, $\Pr(V|\boldsymbol{\pi})$ is defined by

$$\Pr(v|\boldsymbol{\pi}) := \begin{cases} \underline{\Pr}(v|\boldsymbol{\pi}) & \text{if } v \sim \mathbf{h}; \\ \overline{\Pr}(v|\boldsymbol{\pi}) & \text{if } v \approx \mathbf{h}, \end{cases} \quad (7)$$

and for $\overline{\Pr}(\mathbf{h}|\mathbf{e})$, $\Pr(V|\boldsymbol{\pi})$ is defined by

$$\Pr(v|\boldsymbol{\pi}) := \begin{cases} \overline{\Pr}(v|\boldsymbol{\pi}) & \text{if } v \sim \mathbf{h}; \\ \underline{\Pr}(v|\boldsymbol{\pi}) & \text{if } v \approx \mathbf{h}. \end{cases}$$

Proof. We only prove the result for $\underline{\Pr}(\mathbf{h}|\mathbf{e})$; the proof for $\overline{\Pr}(\mathbf{h}|\mathbf{e})$ is completely analogous.

Let v be the unique value of V that is compatible with \mathbf{h} . If the local credal set $\mathcal{M}(V|\boldsymbol{\pi})$ is of Type III, then because of Theorem 1, the minimum value of $\Pr(\mathbf{h}|\mathbf{e})$ is obtained for a local distribution $\Pr(V|\boldsymbol{\pi})$ of which the parameter $\Pr(v|\boldsymbol{\pi})$ is minimal—equal to $\underline{\Pr}(v|\boldsymbol{\pi})$ —and the parameter $\Pr(\bar{v}|\boldsymbol{\pi})$ is maximal. The unique distribution for which this is the case is given by Equation (7). Hence, for the purposes of computing $\underline{\Pr}(\mathbf{h}|\mathbf{e})$, $\mathcal{M}(V|\boldsymbol{\pi})$ can be replaced by the singleton $\mathcal{M}'(V|\boldsymbol{\pi}) := \{\Pr(V|\boldsymbol{\pi})\}$. \square

Note that in this case, $\Pr(V|\pi)$ is unique and belongs to $\mathcal{M}(V|\pi)$.

For example, if we again consider the network in Figure 1, and revisit the problem of computing $\underline{\Pr}(ghk|ef)$, then the local credal set $\mathcal{M}(G)$ —defined in Equation (3)—is of Type III and can therefore be replaced by a singleton $\mathcal{M}'(G) = \{\Pr(G)\}$, with

$$\Pr(g) := \underline{\Pr}(g) = 1/3 \text{ and } \Pr(g) := \overline{\Pr}(g) = 2/3.$$

In our example, in total, eight out of the fifteen credal sets of the credal network in Figure 1 can be replaced by a singleton, whereas other preprocessing techniques—such as the removal of barren or d-separated nodes—are not able to simplify the problem.

4.3 Special cases

Since a local credal set is of a given type if and only if its elements are, the special cases that were discussed in Section 3.4 carry over to credal networks. For example, for a credal network classifier with full evidence, the credal sets that correspond to the evidence variables are all of Type 0, I or II, and therefore, they can be replaced by singletons. It is essentially this feature that allows for tractable computations in basic credal classifiers such as the naive credal classifier [26] and the tree-augmented naive classifier [27]. In fact, many of the formulas in References [26] and [27] can be shown to follow from our results.

In the special case where we do not only have full evidence, but also know that each of the hypothesis variables $V \in \mathbf{H}$ is binary and has no parents in \mathbf{H} , then all the local credal sets of the credal network can be classified as one of the four types in Definition 5. This has far-reaching consequences: in this case, due to the results in the previous section, it follows that for the purpose of computing $\underline{\Pr}(\mathbf{h}|\mathbf{e})$ or $\overline{\Pr}(\mathbf{h}|\mathbf{e})$, all the local credal sets can be replaced by singletons, without changing the result of the inference. Hence, in this case, computing $\underline{\Pr}(\mathbf{h}|\mathbf{e})$ or $\overline{\Pr}(\mathbf{h}|\mathbf{e})$ reduces to the problem of computing $\Pr(\mathbf{h}|\mathbf{e})$ in a specific Bayesian network, the parameters of which can easily be determined by means of the methods in Section 4.2.

For example, in this special case, for the purpose of computing $\underline{\Pr}(\mathbf{h}|\mathbf{e})$, the local credal sets of a credal network can be replaced by precise distributions $\Pr_{\bullet}(V|\pi)$, the parameters of which are defined by

$$\Pr_{\bullet}(v|\pi) := \begin{cases} \underline{\Pr}(v|\pi) & \text{if } V \in \mathbf{H} \text{ and } v \sim \mathbf{h} \\ \overline{\Pr}(v|\pi) & \text{if } V \in \mathbf{H} \text{ and } v \approx \mathbf{h} \\ \underline{\Pr}(v|\pi) & \text{if } V \in \mathbf{E}, v \sim \mathbf{e} \text{ and } \pi \sim \mathbf{h} \\ \overline{\Pr}(v|\pi) & \text{if } V \in \mathbf{E}, v \sim \mathbf{e} \text{ and } \pi \approx \mathbf{h} \\ \dots & \text{if } V \in \mathbf{E} \text{ and } v \approx \mathbf{e} \end{cases} \quad (8)$$

This definition does not distinguish between the cases $\pi \sim \mathbf{e}$ and $\pi \approx \mathbf{e}$ because we know from Proposition 4 that a parameter for which $\pi \approx \mathbf{e}$ does not effect the result of the inference anyway. Similarly, the definition for the case with $V \in \mathbf{E}$ and $v \approx \mathbf{e}$ is not provided because we know from Proposition 5 that these parameters do not influence the result of the inference either.

As an immediate consequence, for this special case, we obtain the following expression:

$$\underline{\Pr}(\mathbf{h}|\mathbf{e}) = \frac{\prod_{V \in \mathbf{V}} \Pr_{\bullet}(v|\pi)|_{v\pi \sim \mathbf{h}\mathbf{e}}}{\sum_{\mathbf{h}^* \in \mathbf{H}} \prod_{V \in \mathbf{V}} \Pr_{\bullet}(v|\pi)|_{v\pi \sim \mathbf{h}^*\mathbf{e}}}$$

Furthermore, by removing the parameters $\Pr_{\bullet}(v|\pi)$ for which V and its parents π belong to \mathbf{E} —since these are common factors of

the numerator and denominator—and then explicitly applying Equation (8) to the numerator, this expression can be simplified even more:

$$\underline{\Pr}(\mathbf{h}|\mathbf{e}) = \frac{\prod_{V \in \mathbf{K}} \underline{\Pr}(v|\pi)|_{v\pi \sim \mathbf{h}\mathbf{e}}}{\sum_{\mathbf{h}^* \in \mathbf{H}} \prod_{V \in \mathbf{K}} \Pr_{\bullet}(v|\pi)|_{v\pi \sim \mathbf{h}^*\mathbf{e}}}$$

where \mathbf{K} is the set of variables that consist of the variables in \mathbf{H} and their children variables.

Similar conclusions can be reached for the problem of computing $\overline{\Pr}(\mathbf{h}|\mathbf{e})$. In the special case of Section 3.4, this problem can again be reduced to computing $\Pr(\mathbf{h}|\mathbf{e})$ in a single Bayesian network. However, of course, it might not be the same Bayesian network as the one that is used to compute $\underline{\Pr}(\mathbf{h}|\mathbf{e})$. In this case, the local credal sets can be replaced by distributions $\Pr_{\bullet}(V|\pi)$, the parameters of which are defined by

$$\Pr_{\bullet}(v|\pi) := \begin{cases} \overline{\Pr}(v|\pi) & \text{if } V \in \mathbf{H} \text{ and } v \sim \mathbf{h} \\ \underline{\Pr}(v|\pi) & \text{if } V \in \mathbf{H} \text{ and } v \approx \mathbf{h} \\ \overline{\Pr}(v|\pi) & \text{if } V \in \mathbf{E}, v \sim \mathbf{e} \text{ and } \pi \sim \mathbf{h} \\ \underline{\Pr}(v|\pi) & \text{if } V \in \mathbf{E}, v \sim \mathbf{e} \text{ and } \pi \approx \mathbf{h} \\ \dots & \text{if } V \in \mathbf{E} \text{ and } v \approx \mathbf{e} \end{cases}$$

5 CONCLUSIONS AND FUTURE WORK

A credal network represents a set of Bayesian networks thereby allowing for imprecisions in its parametrisation. One of the main computational problems in a credal network is the computation of lower and upper probabilities. Given a brute force approach, however, this requires a number of Bayesian network inferences that is exponential in the number of imprecisely specified local probability distributions.

The pruning of all variables irrelevant for a specific problem is a well-known approach to make inference more tractable. In this paper we proposed a different type of preprocessing step. Using Bayesian network sensitivity functions, we proved that for certain categories of parameters, we can predict the qualitative effect of their change on an outcome probability without needing any knowledge of the numerical specification of the network. This result allows for the identification of imprecisely specified local distributions that can be replaced by precisely specified ones, without affecting the outcome of the computation. Depending on the structure of the network and the specific problem at hand, our preprocessing step can be quite rewarding. We argued, for example, that for some classes of networks, even all imprecisely specified local distributions can be replaced by precise ones. In this case, credal network inference reduces to a single Bayesian network computation.

In future work, we would like to investigate the empirical impact of our preprocessing step on various existing algorithms. Moreover, we would like to use our results as a basis for the design of new algorithms. For example, we foresee that the algorithm in [13] could be extended to more general cases. Finally, we would like to extend our work to other types of inference, such as the computation of lower and upper expectations, choosing the most likely hypothesis, and maximising expected utility.

ACKNOWLEDGEMENTS

This research was supported by the Netherlands Organisation for Scientific Research (NWO) and the Fund for Scientific Research – Flanders (FWO).

REFERENCES

- [1] A. Antonucci, C. P. de Campos, M. Zaffalon. 2014. Probabilistic graphical models. In T. Augustin, F. P. A. Coolen, G. De Cooman, M. C. M. Troffaes (eds.), *Introduction to Imprecise Probabilities*, 207-229, John Wiley & Sons, Chichester.
- [2] T. Augustin, F. P. A. Coolen, G. De Cooman, M. C. M. Troffaes (eds.). 2014. *Introduction to Imprecise Probabilities*. John Wiley & Sons.
- [3] J. O. Berger. 1985. *Statistical decision theory and Bayesian analysis*. Springer series in statistics, Springer, New York.
- [4] J. Bernard. 2005. An introduction to the imprecise Dirichlet model for multinomial data. *International Journal of Approximate Reasoning* 39(2-3), 123-150.
- [5] J. De Bock, G. De Cooman. 2015. Credal networks under epistemic irrelevance: the sets of desirable gambles approach. *International Journal of Approximate Reasoning* 56, 178-207.
- [6] J. De Bock, G. De Cooman. 2015. Conditioning, updating and lower probability zero. *International Journal of Approximate Reasoning* 67, 1-36.
- [7] J. De Bock. 2015. *Credal networks under epistemic irrelevance: theory and algorithms*, PhD thesis, Ghent University, Faculty of Engineering and Architecture.
- [8] C. P. de Campos, F. G. Cozman. 2007. Computing lower and upper expectations under epistemic independence. *International Journal of Approximate Reasoning* 44(3), 244-260.
- [9] G. De Cooman, F. Hermans, A. Antonucci, M. Zaffalon. 2010. Epistemic irrelevance in credal nets: the case of imprecise Markov trees. *International Journal of Approximate Reasoning* 51(9), 1029-1052.
- [10] V. M. H. Coupé, L. C. van der Gaag. 2002. Properties of sensitivity analysis of Bayesian belief networks. *Annals of Mathematics and Artificial Intelligence*, 36(4), 323-356.
- [11] F. G. Cozman. 2000. Credal networks. *Artificial Intelligence* 120, 199-233.
- [12] F. G. Cozman. 2005. Graphical models for imprecise probabilities. *International Journal of Approximate Reasoning* 39(2), 167-184.
- [13] E. Fagiuoli and M. Zaffalon. 1998. 2U: an exact interval propagation algorithm for polytrees with binary variables. *Artificial Intelligence*, 106(1), 77-107.
- [14] M. J. Flores, J. A. Gámez, A. M Martínez. 2012. Supervised classification with Bayesian networks: A review on models and applications. *Intelligent Data Analysis for Real-Life Applications: Theory and Practice*, IGI Global, 72-102.
- [15] F. V. Jensen, T. D. Nielsen. 2007. *Bayesian Networks and Decision Graphs*, second edition, Springer Verlag.
- [16] U. Kjærulff, L. C. van der Gaag. 2000. Making sensitivity analysis computationally efficient. In C. Boutilier, M. Goldszmidt (eds.), *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 317-325, Morgan Kaufmann Publishers, San Francisco.
- [17] D. D. Mauá, C. P. de Campos, A. Benavoli, A. Antonucci. 2014. Probabilistic inference in credal networks: new complexity results. *Journal of Artificial Intelligence Research* 50, 603-637.
- [18] M. Meekes, S. Renooij, L. C. van der Gaag. 2015. *Relevance of evidence in Bayesian networks*. In S. Destercke, T. Denoeux (eds.), *Proceedings of the 13th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, LNAI 9161, Springer, 366-375.
- [19] J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Palo Alto.
- [20] A. Piatti, A. Antonucci, M. Zaffalon. 2010. Building knowledge-based systems by credal networks: a tutorial. In A. R. Baswell (ed.), *Advances in Mathematics Research* 11, 227-279, Nova Science Publishers.
- [21] S. Renooij. 2014. Co-variation for sensitivity analysis in Bayesian networks: Properties, consequences and alternatives. *International Journal of Approximate Reasoning*, 55, 1022-1042.
- [22] R. Shachter. 1998. Bayes-Ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In G. Cooper, S. Moral (eds.), *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 480-487, Morgan Kaufmann, San Francisco.
- [23] P. Walley. 1991. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, New York.
- [24] P. Walley. 1996. Inferences from multinomial data: learning about a bag of marbles. *Journal of the Royal Statistical Society, Series B* 58, 3-57.
- [25] P. Walley. 2000. Towards a unified theory of imprecise probability. *International Journal of Approximate Reasoning* 24(2-3), 125-148.
- [26] M. Zaffalon. 2002. The naive credal classifier. *Journal of Statistical Planning and Inference*, 105(1), 5-21.
- [27] M. Zaffalon and E. Fagiuoli. 2003. Tree-Based Credal Networks for Classification. *Reliable Computing*, 9, 487-509.

Interval-Based Relaxation for General Numeric Planning

Enrico Scala and Patrik Haslum and Sylvie Thiebaux and Miquel Ramirez
 The Australian National University and NICTA
 Canberra, ACT, Australia
 firstname.lastname@anu.edu.au

Abstract. We generalise the interval-based relaxation to sequential numeric planning problems with non-linear conditions and effects, and cyclic dependencies. This effectively removes all the limitations on the problem placed in previous work on numeric planning heuristics, and even allows us to extend the planning language with a wider set of mathematical functions. Heuristics obtained from the generalised relaxation are pruning-safe. We derive one such heuristic and use it to solve discrete-time control-like planning problems with autonomous processes. Few planners can solve such problems, and search with our new heuristic compares favourably with them.

1 Introduction

The ability to express quantitative information is crucial to realistically model many planning domains, in particular domains that involve interaction with physical systems. Examples include positions in time and space, quantities such as pressure, flow or volume, as well as resources.

Planning with unbounded numeric variables is significantly harder than classical planning with only finite-domain variables. Not only can the values of variables grow unboundedly, but even finite values may be reachable only asymptotically. For example, repeatedly applying the effect $x = (x + y)/2$ brings x arbitrarily close to y , whilst repeatedly applying the pair of effects $x = (x + y)/2$ and $y = x \times y$ can either diverge to $\pm\infty$ or converge to 0 or 1, depending on the starting values of x and y . In general, numeric effects can be state-dependent: The effect on x of the update $x = (x + y)/2$ depends on the current value of both x and y . Behaviours of this kind complicate reachability analysis, and therefore the construction of informative heuristics for numeric planning. Because of this, work on heuristic search for numeric planning has focused on restricted forms of numeric conditions and effects, such as linear expressions. Starting from the MetricFF planner [17], the majority of numeric planning heuristics have (implicitly or explicitly) relied on an interval relaxation, which approximates reachable values with upper and lower bounds. Only recently did Aldinger et al. [1] examine the interval-based relaxation theoretically, and show that asymptotic reachability is decidable for the full range of numeric effects expressible with arithmetic expressions (built using $+$, $-$, \times and \div), as allowed in PDDL 2.1 [11]. However, even their analysis is restricted to problems with only a very limited form of cyclic dependencies between variables. The example above, in which the effect on both x and y depend on the value of both variables, is outside their scope.

In this paper we show that interval-based relaxed reachability analysis is feasible, and produces informative heuristic guidance, for a much wider range of numeric planning problems, involving

cyclic dependencies as well as expressions using standard mathematical functions such as exponentiation, square root, and others. This is achieved by two innovations: an asymptotic relaxed reachability analysis that works also with cyclic dependencies among additive numeric effects, and a syntactic transformation of the problem which eliminates non-additive state-dependent effects, at the cost of inducing an additional relaxation.

Finally, we apply the resulting heuristic to solve time-discretised planning problems with autonomous processes, which feature complex numeric conditions and non-linear effects. Few planners have the expressive range to attempt these problems and heuristic search using our generalised interval-based relaxation heuristic is more efficient than comparable alternatives.

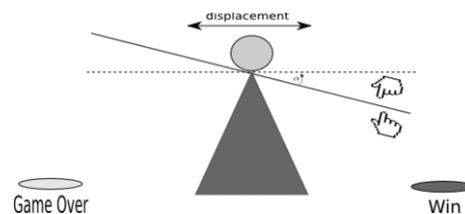


Figure 1: The ball moves along the plane, and is accelerated by tilting the plane. The goal is to make it fall into the winning hole on the right. If the ball falls off the left side, the game is lost.

2 Motivating Example

Let us consider the numeric domain represented in Figure 1: A ball, at position x on a plane, moves with velocity v , possibly falling from one of the sides. $x = 0$ at the centre, and position and velocity is positive to the right. The change in velocity is a function of the inclination α of the plane, accounting for gravity (g) and drag along the surface, proportional with a constant μ to the speed squared (like in the car domain by Bryce et al. [3]). The drag always acts in the opposite direction to the current velocity. An action models the *discretised* movement of the ball over δ_t units of time, changing variables x and v as follows:

$$x = x + v\delta_t$$

$$v = \begin{cases} v + (g \sin \alpha - \mu v^2)\delta_t & \text{when } v \geq 0 \\ v + (g \sin \alpha + \mu v^2)\delta_t & \text{when } v < 0 \end{cases}$$

The left hand side is the updated value of x and v . δ_t is a constant and determines the precision of the discretisation. The other two actions increase and decrease α by a constant amount, by tilting the plane.

The goal is to bring the ball into the hole on the right; this is achieved when x exceeds a given threshold.

This domain has both cyclic dependencies (i.e., a self-cycle in the second numeric effect) and non-linear effects (e.g., the velocity squared). Earlier work on defining heuristics for numeric planning has focused on linear effect expressions [17, 5, 13] or on *acyclic* planning tasks [1]. By removing these two assumptions, we address this class of numeric planning problems. Our key concern is the relaxation that underlies heuristics.

3 Notation and Background Material

We focus on sequential numeric planning with ground actions, corresponding to PDDL 2.1 level 2 [11] but extended with additional interpreted mathematical functions.

A state of the system is a partial assignment over propositions \mathcal{P} and real-valued numeric variables \mathcal{X} ; $\mathcal{V} = \mathcal{P} \cup \mathcal{X}$. A propositional condition is a positive literal, while a numeric condition is a tuple $\langle \xi, \triangleright, 0 \rangle$ where $\triangleright \in \{\geq, >, =\}$ and ξ is a numeric expression recursively defined as follows: (I) a rational constant is an expression; (II) a variable $x \in \mathcal{X}$ is an expression; (III) if ξ' is an expression, then so are $\xi \oplus \xi'$ where $\oplus \in \{+, -, \cdot, \div\}$, ξ^n , where $n \in \mathbb{N}$, b^ξ and $\log_b(\xi)$, where $b \in \mathbb{R}$ and $b > 0$, and $\sqrt[\xi]{\cdot}$. Other functions (e.g., trigonometric functions, n th root, etc) can also be supported, as long as they are functions in the mathematical sense, and computable; for the interval-based relaxation, functions must also have an interval extension (cf. next section). We write $\text{val}(x, s)$ and $\text{val}(\xi, s)$ for the value of variable x and expression ξ in state s , respectively. The value of a variable not assigned in s is *undefined*. Undefinedness propagates through expressions, in the sense that the value of a function is undefined whenever any of its arguments is. Some functions have restricted domains, and their result is undefined when any of their arguments is outside its domain; for example, \sqrt{x} is undefined whenever $x < 0$. A condition $\langle \xi, \triangleright, 0 \rangle$ is unsatisfied whenever the value of ξ is undefined [11]. With slight abuse of notation, we write $x \in \xi$ to mean that variable x appears in expression ξ .

Let \mathcal{C} be a set of propositional and numeric conditions. We write $s \models \mathcal{C}$ when state s satisfies all conditions in \mathcal{C} .

Definition 1 (Numeric Action). *A numeric action a is a pair $\langle \text{pre}(a), \text{eff}(a) \rangle$ where $\text{pre}(a)$ is a set of propositional and numeric conditions and $\text{eff}(a)$ is a set of effects. A classical effect is of the form $p = \top$ or $p = \perp$ ($p \in \mathcal{P}$). A numeric effect is $x \circ = \xi$, where $\circ = \{=, +, -, =\}$ and ξ is an expression over variables in \mathcal{X} . $\text{eff}(a)$ cannot contain multiple effects on the same variable.*

We use subscripts to distinguish propositional and numeric parts (e.g., $\text{eff}_{\text{num}}(a)$ is the set of numeric effects of a). For an effect e , $\text{lhs}(e)$ denotes the affected variable, $\text{op}(e)$ the effect operator ($=$, $+$ or $-$) and $\text{rhs}(e)$ denotes the right-hand side expression.

Definition 2 (Numeric Planning Problem). *A numeric planning problem is a tuple $\Pi = \langle s_0, \mathcal{A}, \mathcal{G}, \mathcal{V} \rangle$ where $\mathcal{V} = \mathcal{P} \cup \mathcal{X}$ is the set of variables, s_0 is an assignment to variables in \mathcal{V} that is complete for variables in \mathcal{P} , \mathcal{A} is a set of numeric actions, and \mathcal{G} is a set of propositional and numeric goal conditions.*

An effect is called *state-dependent* if the right-hand side is not a constant. Increase and decrease effects can be reformulated as assignments (e.g., $x += \xi$ as $x = x + \xi$). However, we will see that this distinction is, surprisingly, crucial, because increase and decrease effects are *additive* operations, which assignments are, in general, not.

Given an action a , we partition $\text{eff}_{\text{num}}(a)$ into sets $\text{incr}(a)$, $\text{decr}(a)$ and $\text{assn}(a)$ of increase, decrease and assign effects, respectively. We also write $\text{const_assn}(a)$ for the set of state-independent propositional and numeric assignment effects of a (e.g., $x = 5$).

Action a is applicable in state s iff $s \models \text{pre}(a)$, and its execution results in state $s' = \text{succ}(s, a)$ such that $\forall x \in \mathcal{V} : \text{val}(x, s') =$

$$\begin{cases} \text{rhs}(e) & \text{if } \exists e \in \text{const_assn}(a) : \text{lhs}(e) = x \\ \text{val}(\text{rhs}(e), s) & \text{if } \exists e \in \text{assn}(a) : \text{lhs}(e) = x \\ \text{val}(x, s) + \text{val}(\text{rhs}(e), s) & \text{if } \exists e \in \text{incr}(a) : \text{lhs}(e) = x \\ \text{val}(x, s) - \text{val}(\text{rhs}(e), s) & \text{if } \exists e \in \text{decr}(a) : \text{lhs}(e) = x \\ \text{val}(x, s) & \text{otherwise (Frame Axiom)} \end{cases}$$

Definition 3 (Plan). *A plan π for $\Pi = \langle s_0, \mathcal{A}, \mathcal{G}, \mathcal{V} \rangle$ is a sequence of actions a_0, \dots, a_{n-1} from \mathcal{A} such that each action in π is applicable in the state resulting from the application of its predecessors, i.e., $s_0 \models \text{pre}(a_0)$, $\text{succ}(s_0, a_0) \models \text{pre}(a_1)$, etc, and $\text{succ}(s_{n-1}, a_{n-1}) \models \mathcal{G}$. A plan π is said to be optimal if, among all valid plans, it has a minimal number of actions.*

3.1 The Interval-Based Relaxation

Numeric planning is harder than propositional planning, and undecidable in the general case [15]. To obtain useful heuristics we must look for relaxations of the model that yield a computationally effective representation of the task to solve. Moreover, it is important to look for relaxations generating heuristics that are *adequate* for the task they are relaxing [17]. We consider only relaxations (and heuristics) that are *pruning-safe*. A relaxation is pruning-safe if it has no solution only when the original (non-relaxed) problem is also unsolvable; that is, it overestimates the space of reachable states.

An idea pursued in previous work in both classical and numeric planning is that of abstracting away negative effects of the actions, considering only their positive contribution towards achieving a goal or precondition. In propositional planning this amounts to ignoring the delete effects of actions. In general, it implies a possibilistic relaxed interpretation, in which variables accumulate sets of possible values monotonically [14]. Applying this principle to numeric variables, which have unbounded domains, requires a compact representation of the set of values; this is provided by the interval-based representation [1, 23]. Pioneered by Hoffman [17]¹, the interval-based relaxation [1] is the underlying principle used in nearly all heuristics for numeric planning [17, 13, 5, 6]. An exception is the work of Eyerich et al. [9] on the Temporal Fast Downward system, which extends the context-enhanced additive heuristic [16] to temporal and numeric planning. However, this heuristic does not account for indirect effects: action a indirectly affects variable x when the effect e of another action b on x depends on a variable y (in $\text{rhs}(e)$) which is changed by a . Ignoring indirect effects in numeric planning makes any reachability analysis not pruning-safe (The context-enhanced additive heuristic is also not pruning-safe, but for a different reason.)

For ease of presentation, we will in the following describe only the relaxation of the numeric part of the problem. The propositional part is handled by standard delete-relaxation. The two parts interact only in the relaxed satisfaction of conjunctive conditions.

¹ Note that Hoffman does not exploit the interval representation explicitly, but uses a transformation to Linear Normal Form where variables' domains can only increase. This can be interpreted as constructing an enclosure, by lower and upper bounds, of the possible values attainable by a variable.

3.1.1 Definition of the Interval-Based Relaxation

In the interval-based relaxation (IBR), a state assigns each (defined) numeric variable to an interval of the real line, representing the set of values that the variable can possibly attain. We will refer to this as a *relaxed state*, s^+ . A closed interval $x = [\underline{x}, \bar{x}]$ denotes the lower bound \underline{x} and upper bound \bar{x} a variable x can attain. An open interval $x = (\underline{x}, \bar{x})$ is analogous but with \underline{x} and \bar{x} excluded; i.e., $(\underline{x}, \bar{x}) = \{z : \underline{x} < z < \bar{x}, z \in \mathbb{R}\}$. A mixed bounded interval mixes open and closed bounds. Closed interval binary operations between two intervals x and y are defined as follows [23]:

- $x + y = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$;
- $x - y = [\underline{x} - \bar{y}, \bar{x} - \underline{y}]$;
- $x \times y = [\min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}), \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y})]$;
- $x \div y = [\min(\underline{x} \div \underline{y}, \underline{x} \div \bar{y}, \bar{x} \div \underline{y}, \bar{x} \div \bar{y}), \max(\underline{x} \div \underline{y}, \underline{x} \div \bar{y}, \bar{x} \div \underline{y}, \bar{x} \div \bar{y})]$ (if $0 \notin y$ otherwise one of the bounds diverges [23]).

Binary operations between open or mixed bounded intervals follows the same rules; if an open and a closed bound contribute to the new interval bound, the result is open.

The interval extensions of other mathematical functions in the planning language (such as x^n , \sqrt{x} , b^x , $\log_b(x)$) can be similarly defined. The requirement is only that the result of the interval operation contains every value that could result from applying the function to any value(s) in the argument interval(s). To illustrate, we show the interval power, exponentiation and square root. Let x be an interval $[\underline{x}, \bar{x}]$, $n \in \mathbb{N}$ and $b \in \mathbb{R}$, $b > 0$. We have:

$$x^n = \begin{cases} [\underline{x}^n, \bar{x}^n], & \text{if } \underline{x} > 0 \text{ and } n \text{ is odd} \\ [\bar{x}^n, \underline{x}^n], & \text{if } \bar{x} < 0 \text{ and } n \text{ is even} \\ [0, \max(\underline{x}^n, \bar{x}^n)], & \text{if } 0 \in x \text{ and } n \text{ is even} \end{cases}$$

$$b^x = \begin{cases} [b^{\underline{x}}, b^{\bar{x}}] & \text{for } 0 < b < 1 \\ [b^{\bar{x}}, b^{\underline{x}}] & \text{for } b \geq 1 \end{cases}$$

$$\sqrt{x} = \begin{cases} [\sqrt{\underline{x}}, \sqrt{\bar{x}}], & \text{if } \underline{x} \geq 0 \\ [0, \sqrt{\bar{x}}], & \text{if } \bar{x} \geq 0 \\ \text{undefined}, & \text{otherwise} \end{cases}$$

The square root of a negative value is not a real number. Hence, the interval square root is the result of restricting the argument interval to the range over which the function is defined, and undefined only if there is no such value. All three extend to open or mixed bounded intervals: for integral power, inequalities in the two first cases are non-strict; for the square root, the second case applies when $\bar{x} > 0$.

Expressions, conditions and actions in the IBR are syntactically the same as in the original PDDL problem; what changes is their interpretation. In particular, the value of an expression ξ in a relaxed state s^+ , denoted $\text{val}^+(\xi, s^+)$, is the interval computed using interval operations as defined above. Note that mathematical identities are not necessarily preserved by the relaxed interpretation. For example, if $\text{val}^+(x, s^+) = [-2, 2]$, then $\text{val}^+(x \times x, s^+) = [-4, 4]$, while $\text{val}^+(x^2, s^+) = [0, 4]$. A numeric condition $\langle \xi, \geq, 0 \rangle$ is satisfied in a relaxed state s^+ if $\text{val}^+(\xi, s^+)$ is defined and there exists some value $v \in \text{val}^+(\xi, s^+)$ such that $v \geq 0$. A set of conditions (that can appear both in the goal or action preconditions) is relaxed satisfied iff each condition in the set is. We write $s^+ \models C$ when s^+ satisfies C .

An action a is applicable in s^+ if $s^+ \models \text{pre}(a)$. In the IBR, action effects can only extend the set of possible values of a variable. To define the relaxed successor state, we need the convex union:

Definition 4. *The convex union between two closed intervals x, y is $x \sqcup y = [\min\{\underline{x}, \underline{y}\}, \max\{\bar{x}, \bar{y}\}]$. The extension to open or mixed*

bounded intervals uses open/closed bounds according to those used for x and y .

Definition 5 (Relaxed Action Effects). *Let s^+ be a relaxed state and a an action such that $s^+ \models \text{pre}(a)$. The successor state $s_1^+ = \text{succ}^+(s^+, a)$ is such that $\forall x \in \mathcal{X}$, $\text{val}^+(x, s_1^+) =$*

$$\begin{cases} \text{val}^+(x, s^+) \sqcup [\text{rhs}(e), \text{rhs}(e)] & \text{if } x = \text{lhs}(e), e \in \text{const_assn}(a) \\ \text{val}^+(x, s^+) \sqcup \text{val}^+(\text{rhs}(e), s^+) & \text{if } x = \text{lhs}(e), e \in \text{assn}(a) \\ \text{val}^+(x, s^+) \sqcup (\text{val}^+(x, s^+) + \text{val}^+(\text{rhs}(e), s^+)) & \text{if } x = \text{lhs}(e), e \in \text{incr}(a) \\ \text{val}^+(x, s^+) \sqcup (\text{val}^+(x, s^+) - \text{val}^+(\text{rhs}(e), s^+)) & \text{if } x = \text{lhs}(e), e \in \text{decr}(a) \\ \text{val}^+(x, s_1^+) & \text{otherwise} \end{cases}$$

Applying actions' effects in the IBR can only *monotonically increase* the variables' intervals, because the convex union of two intervals contains both. Hence, this relaxation, just like the classical delete relaxation, is also monotonic with respect to condition satisfaction: what is true before an action is applied is also true after its execution. This property ensures that the relaxation over-approximates the set of reachable conditions, and therefore that it is pruning-safe.

Given a numeric planning problem Π , we denote its relaxation by Π^+ . The initial state s_0^+ of Π^+ assigns to each variable $x \in \mathcal{V}$ the unit interval $[\text{val}(x, s_0), \text{val}(x, s_0)]$, if x is defined in s_0 .

3.1.2 Asymptotic Behavior of Additive Numeric Effects

In numeric planning, there may be no upper bound on the length of the action sequence needed to reach a state (or condition). Because of this, Aldinger et al. [1] considered *asymptotic reachability* in the interval-based relaxation, i.e., the variable intervals that are reachable in the limit of an infinite repetition of actions' effects. The following proposition is slightly adapted from their work. (We present it for increase effects only; the result for decrease effects is analogous.)

Proposition 1. *Let s^+ be a relaxed state, $x \in \mathcal{X}$ a numeric variable, $a \in \mathcal{A}$ an action such that $s^+ \models \text{pre}(a)$, and $x += \xi$ an effect of a . Let s_{lim}^+ be the relaxed state reached in the limit by applying a an unbounded number of times in s^+ . Then the upper and lower open bounds of x in s_{lim}^+ are as follows:*

- if $\exists y' > 0 \in \text{val}^+(\xi, s^+)$ then $\bar{x} = \infty$ in s_{lim}^+ ; and
- if $\exists y' < 0 \in \text{val}^+(\xi, s^+)$ then $\underline{x} = -\infty$ in s_{lim}^+ .

In other words, if the right-hand side of an increase effect affecting variable x can attain a positive value, no matter how small, then the value of x can be made arbitrarily large by repeated application of the action. This follows directly from the monotonicity of the interval-based relaxation, since if the interval of the right-hand side of the effect, ξ , contains a positive value in s^+ then it contains that value in any successor state. The rate of increase is not constant in general, but always greater or equal to the rate in the first state.

The following notion of dependency between numeric variables was defined by Aldinger et al. We repeat it here because it is necessary for stating precisely how we generalise their work.

Definition 6 (Slightly adapted from Aldinger et al. [1]). *A numeric variable x_1 is directly dependent on a numeric variable x_2 in task Π if there exists an action a in \mathcal{A} with a numeric effect $\langle x_1 \circ = \xi \rangle$ ($\circ \in \{=, +, -\}$) such that $x_2 \in \xi$.*

A variable may depend on itself. However, formulating effects with increase and decrease operators can avoid certain cycles. For example, the assignment $x = x + 1$ causes x to self-depend, but the equivalent increase effect $x += 1$ does not.

Aldinger et al. [1] proved that asymptotic reachability in the interval-based relaxation is decidable for planning problems where the dependency relation over the numeric variables has no cycle. In this paper, we *remove this restriction*. Moreover, we also show that the IBR can support an extended set of mathematical functions in the planning language, such as exponentiation and square root, which have not been supported in previous work.

We approach the derivation of a heuristic based on the interval relaxation in two parts: First, we present a procedure to determine if a condition is asymptotically reachable in the relaxed problem; only if that is the case do we then compute an estimate of the number of actions required to achieve it. The second part treated in Section 5.3.

4 The Additive Effects Transformation

Our generalisation starts from the observation that one reason why Aldinger et al. [1] required the acyclic dependency assumption is the presence of state-dependent non-additive action effects. Non-additive effects can be transformed into additive ones by simple reformulation which preserves their semantic in the real (non-relaxed) problem. An assignment $x = \xi$ can be rewritten as $x += \xi - x$. The expressions $x + \xi - x$ and ξ are equivalent in real arithmetic. Formally:

Definition 7 (Additive Effects Transformation). *The additive transformation of an effect e is*

- $\tau(e) := \text{lhs}(e) += \text{rhs}(e) - \text{lhs}(e)$ if $\text{op}(e)$ is an assignment and $\text{rhs}(e)$ is non-constant; and
- $\tau(e) := e$ otherwise.

The additive effects transformation of a numeric planning problem Π is obtained by applying τ to all effects of all actions in Π , and is denoted by $\tau(\Pi)$.

Note that this transformation does not change constant (numeric or propositional) assignments. We refer to the interval-based relaxation applied to the additive effects transformation of Π as the Additive Interval-Based Relaxation (AIBR) of Π .

Definition 8. *The additive interval-based relaxation of a numeric planning problem Π is the interval-based relaxation of $\tau(\Pi)$, and is denoted by Π^{++} .*

A natural question is what is the relation between AIBR and IBR? AIBR is an over-approximation of IBR: The space of reachable states in Π^{++} includes that of Π^+ . This implies that heuristics based on the the AIBR are also pruning-safe. The inclusion is a simple consequence of how the interval arithmetic is defined: $x \subseteq x + y - y$ for any intervals x, y , and consequently $z \sqcup x \subseteq z \sqcup (x + y - y)$, for any z . To see how the AIBR can overestimate (asymptotic) reachability, consider a very simple planning example with two actions $a = \langle \emptyset, x = y \rangle$ and $b = \langle \emptyset, y = 1 \rangle$ and a state $s = \langle x = 0, y = 0 \rangle$. In the IBR, the (asymptotically) reachable value of x is the interval $[0, 1]$. Applying the additive effects transformation to a we get $\tau(a) = \langle \emptyset, x += y - x \rangle$. From Proposition 1 it is easy to see that the asymptotically reachable value of x is now $[0, \infty)$. Applying b we have $\text{val}^+(y, \text{succ}^+(s^+, b)) = [0, 1]$, and, since intervals can only grow, $0 \in \text{val}^+(x, \text{succ}^+(s^+, \dots))$. Hence the right-hand side of the additive effect $(y - x)$ includes 1, and applying the action repeatedly achieves arbitrary large values of x . Of

course, if the task does not include any state-dependent assignment effects, the set of reachable states is exactly the same.

The transformation addresses only partially the problem, since cyclic dependencies can still occur. Crucially, however, those dependencies are only between additive effects.

5 From Numeric Actions to Supporters and the Asymptotic Relaxed Planning Graph

As observed in the previous section, we can identify the asymptotic implications of additive numeric effects on a variable by evaluating the *sign* of its right-hand side. The case of constant assignments ($x = k$ where k is a constant) is simpler as these are idempotent; repeated application of such effects does not change their initial effect.

A difficulty arises when the possible value of the right-hand side of some numeric effect depends on the state in which the effect is applied, since this state depends on the sequence of actions that were executed before it. As noted by Aldinger et al. [1], this is particularly problematic when the planning problem has cyclic dependencies. In this case there is no a-priori ordering of actions that is guaranteed to produce the complete set of reachable values. If, on the other hand, the dependency relation is acyclic, it suffices to apply the asymptotic effect of each action once, in order of the dependency relation.

5.1 Supporters Set

State-dependent numeric effects can be interpreted as a compact way of expressing conditional effects. Each of these conditional effects depends on the specific value of the right hand side of the effect, which in turn depends on the sequence of actions done before. The main idea of our asymptotic reachability analysis is to make explicit the conditions for such indirect effects to occur through the use of auxiliary actions called “supporters”.

Each action a is split into a set of supporters, each modelling one possible asymptotic outcome of an effect of a . As observed in Proposition 1, an additive numeric effect can extend the interval of the affected variable to ∞ or $-\infty$. The idea of supporters is to model explicitly the condition enabling these asymptotic behaviors. A supporter is similar to an action, with the difference that it can express a numeric interval effect $x = (\underline{x}, \infty)$ or $x = (-\infty, \bar{x})$. As per Definition 5, this effect updates the state to the convex union of the interval x in the current state and (\underline{x}, ∞) or $(-\infty, \bar{x})$, respectively.

Definition 9 (Supporters of Action a). *Each additive numeric effect $e \in \text{eff}_{\text{num}}(a)$ generates two supporters e^+ and e^- : If $\text{op}(e)$ is $+=$, then e^+ has the precondition $\text{pre}(a) \cup \{\text{rhs}(e), >, 0\}$ and the effect $\text{lhs}(e) = (\underline{x}, +\infty)$; the precondition of e^- is $\text{pre}(a) \cup \{\text{rhs}(e), <, 0\}$ and the effect $\text{lhs}(e) = (-\infty, \bar{x})$. The supporters generated by a decrease effect are defined analogously, but with the left-hand sides of the effects swapped. A constant assignment effect $x = k$ generates only one supporter, with precondition $\text{pre}(a)$ and the effect unchanged.*

Note that in actual numeric planning, assigning an interval is not possible. The supporters are used only in the AIBR to compactly represent application of an effect an arbitrary number of times.

From the set of actions \mathcal{A} we generate a set of supporters Ω using this mechanism. The size of Ω is no more than $2n|\mathcal{A}|$, where n is the maximum number of effects per action. Supporters have no cyclic dependencies, according to Definition 6, because their effects are all constant assignments. The dependencies are instead captured by their

preconditions. We can therefore compute asymptotic relaxed reachability in a manner analogous to relaxed planning graph construction over Ω , as described in the next section.

5.2 Asymptotic Relaxed Planning Graph

This section describes the construction of the Asymptotic Relaxed Planning Graph (ARPG), \mathbb{G} , from the set of supporters. As in the classical relaxed plan graph construction, \mathbb{G} is a digraph of alternating interval (corresponding to “fact”) and supporter (corresponding to “action”) layers. The construction of \mathbb{G} starts with the unit intervals corresponding to the initial state of the planning task, and expands it with a supporter layer containing all supporters in Ω whose preconditions are relaxed satisfied, followed by a new interval layer updated with the effects of the applied supporters. The process iterates until either the goal condition is relaxed satisfied in the last interval layer, or no new supporters can be added (in which case the goal is unreachable). Algorithm 1 formalises the process. With slight abuse of notation, we write $\text{succ}^+(s^+, \mathcal{S})$ for the (relaxed) state that results from simultaneously applying all actions in supporter set \mathcal{S} . This is well-defined since all supporters are constant assignments and the successor state is defined by taking the convex union.

Algorithm 1: Asymptotic Relaxed Planning Graph (ARPG)

Input: Π^{++}
Output: Is \mathcal{G} reachable?
1 $\Omega = \text{supporters of } \mathcal{A}$.
2 $s^+ = s_0^+$.
3 $\mathcal{S} = \{a \in \Omega : s^+ \models \text{pre}(a)\}$
4 **while** $\mathcal{S} \neq \emptyset$ **and** $s^+ \not\models \mathcal{G}$ **do**
5 $s^+ = \text{succ}^+(s^+, \mathcal{S})$
6 $\Omega = \Omega \setminus \mathcal{S}$
7 $\mathcal{S} = \{a \in \Omega : s^+ \models \text{pre}(a)\}$
8 **return** $s^+ \models \mathcal{G}$

Figure 2 shows an example of the conversion from actions to supporters, and a sketch of the asymptotic reachability analysis.

Proposition 2 (Termination). *The ARPG construction terminates.*

Proof Sketch: The set of supporters is finite, so, since repetitions are not allowed, the number of layers that can be built is finite. In the worst case, the construction will terminate when all applicable supporters have been tried.

Proposition 3 (Safeness). *If ARPG(Π^{++}) returns False, then \mathcal{G} is unreachable in Π .*

Proof Sketch: Π^{++} is a proper relaxation of Π . Assume \mathcal{G} is reachable, by a plan $\pi = b_0, \dots, b_m$ that takes s_0 to a state s' such that $s' \models \mathcal{G}$. The plan is executable in Π^{++} (by monotonicity) and the goal is relaxed satisfied in its final state. From the relaxed execution of π , we can extract the corresponding supporters that are used; as these are relaxed applicable, the ARPG construction returns True.

5.3 Heuristic Estimation

The ARPG construction decides only whether \mathcal{G} is reachable, in the relaxed problem, by application of some, arbitrarily large, number of actions. To compute an estimate of how many actions, we can proceed to build an RPG in the standard fashion, applying (sets of)

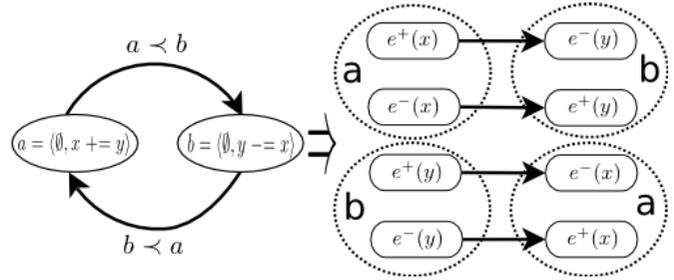


Figure 2: Left: Cyclic dependency ($<$) example. Action a depends on y which is modified by b ; b depends on x which is modified by a . Right: Supporters generated for the two actions. Dependencies between supporters are due to the added preconditions (e.g., $\text{pre}(e^+(x)) = \{y > 0\}$). If $s_0 = \langle x = -5, y = -5 \rangle$, supporters in the first layer are $e^+(y)$ and $e^-(x)$. Asymptotically all values of x and y are reachable both in the AIBR and in the original problem. If we want to reach goal $y < -100$ in the AIBR we have to apply first $e^+(y)$ (i.e., using action b) then $e^+(x)$ (i.e., action a), and then $e^-(y)$ (i.e., action b again).

relaxed applicable actions until the goal is reached and then counting how many were needed. Algorithm 2 shows the relaxed plan computation procedure, for the sake of completeness. Because the goal has already been determined to be relaxed reachable, this process is guaranteed to terminate in a finite number of steps.

Algorithm 2 simply counts the number of actions applied before the goal is achieved. This is the heuristic estimate used in our experiments. Although it can obviously overestimate the size of the relaxed plan substantially, it is very simple to implement and still provides effective heuristic guidance, as shown in Section 7. The heuristic could be made more accurate by removing from \mathcal{A}' actions that do not contribute to achieving \mathcal{G} , in a way that is similar to relaxed plan extraction. How to do that efficiently and in a principled way, as well as how to find an admissible estimate, is a question for future work.

Algorithm 2: Compute AIBR Estimate

Input: Π^{++}
Output: Integer
1 $\mathcal{A}' = \emptyset$
2 $s = s_0^+$
3 **Loop**
4 **foreach** $a \in \mathcal{A}$ **do**
5 **if** a is relaxed applicable in s **then**
6 $s = \text{succ}^+(s, a)$
7 $\mathcal{A}' = \mathcal{A}' \cup \{a\}$
8 **if** $s \models \mathcal{G}$ **then**
9 **return** $|\mathcal{A}'|$

6 Planning with Autonomous Processes in PDDL+

Sequential numeric planning is a crucial building block to supporting more expressive planning formalisms, such as planning with autonomous processes as captured in PDDL+ [12]. In the theory of waiting, proposed by McDermott [22] for planning with autonomous processes, instantaneous actions (the agent decisions) are interleaved with waiting (environment evolution). During a waiting action, the agent observes the evolution of the world from a potentially unbounded

amount of time. During this time, the state changes according to the accumulated, *additive*, effect of active processes. The activation of processes is determined by their conditions on the evolving state.

In principle, the amount of time an agent must wait in a given situation can be arbitrarily large or small. Waiting too long it may miss a window of opportunity to take some action whose precondition is only briefly satisfied, or the dynamics of the world may change as its evolution triggers a change in the set of active processes. Fixing the minimum waiting time to a constant δt results in a time-discretised approximation of the model, and a sequential numeric planning problem. The domain presented in Section 2 is an example of such a model. As shown by Löhr et al. [20, 21], this approximation is good enough to solve a number of challenging realistic hybrid control problems. The “discretise-and-validate” approach to planning with processes, exemplified by the UPMurphi planner [8], also has planning for a time-discretised model at its core, and embeds it into an iterative scheme in which the proposed plan is checked against the continuous-time model and the discretisation refined if it is found to be invalid. Importantly, the time-discretisation of a process model typically results in a sequential planning problem with complex numeric conditions and effects. UPMurphi solves this problem by blind search; a recently developed successor system, DiNo [24], uses a heuristic based on relaxed planning graphs.

We focus here on planning for time-discretised models with instantaneous actions, processes and global constraints². A global constraint is a boolean formula over propositional and numeric conditions that must be satisfied in every state along the plan trajectory, and they are equivalent to *always* constraints of PDDL. We use the global constraint mainly to model the equivalent of PDDL+ events. In many PDDL+ domains, events restrict particular plan trajectories by leading to dead-end states (e.g., blowing up the engine in the CAR domain), playing a role similar to global constraints. Whether events do in fact make the language more expressive than global constraints alone can be an open question, but they are known to be the cause of significant modeling and computational complications (e.g., infinite cascades of events [10]).

We solve the sequential numeric planning problem with a best-first search using the AIBR heuristic. Because the AIBR is monotonic, applying the effect of processes in the relaxed problem becomes optional rather than mandatory (i.e., a “may” instead of a “must” semantic). Likewise global constraints, which are always satisfied in the state being evaluated, cannot be invalidated in the relaxed problem. In spite of these approximations, the heuristic provides effective guidance, as shown in the next section.

7 Experiments

To evaluate the effectiveness of the AIBR heuristic, we perform experiments on a set of time-discretised PDDL+ domains involving autonomous processes and global constraints. We did not consider IPC numeric planning benchmarks. These have only simple (linear or constant) conditions and effects, which are already known to be handled, in most cases quite effectively, by heuristics based on the interval relaxation. Next, we present the setting and domains used, then a summary of results, an in-depth analysis for selected domains, and a comparison with two state of the art numeric planners.

² As observed by Fox et al. [12], durative actions can be compiled away using two instantaneous actions modeling the start and the stop of a process, which in turns captures the continuous effects of the durative action.

7.1 Setting

We implemented a numeric planner using best-first search on $f(n) = g(n) + h(n)$, where $h(n)$ is the AIBR heuristic. As in McDermott’s planner [22], branching is on both (sets of) instantaneous action(s) and waiting. The waiting time is fixed to a constant δt . The successor state of the waiting realises the cumulative effect of all processes active at that particular moment. The planner is correct and complete for the time-discretised problem; of course, like UPMurphi [8] without the validation, the approach is incomplete for the continuous-time PDDL+ semantics (e.g., we can miss opportunities during the δt), and can produce invalid plans as we do not check zero-crossings within the waiting time (e.g., there could be other processes triggered, or global constraints invalidated in that interval). Since our heuristic can overestimate the actual distance to the goal, it is inadmissible and can produce suboptimal plans.

The implementation is in JAVA 1.8, and experiments were run on Ubuntu 14.04 on an Intel I5-vPro with 8 GB of memory. The planner is called Expressive Numeric Heuristic Search Planner (ENHSP), and will be publicly available.

7.2 Planning Domains

The large majority of our test domains require reasoning about autonomous non-linear processes. They include the well-known PDDL+ domains CAR [12] and GENERATOR [3], in both their linear and non-linear versions, the CONVOYS domain introduced by McDermott [22], and two new domains INTERCEPT and HVAC. To these, we add a challenging pure sequential numeric planning domain called COMPLEXPOURING.

Both CAR domain versions involve instantaneous accelerate/decelerate actions and processes updating the distance driven and speed of the car. The non-linear version features a new process, which is active when the speed v is positive and additionally accounts for the drag k , according to $\frac{dv}{dt} = -k \cdot v^2$. Like Bryce et al. [3], we test the scalability of our approach by increasing the maximum acceleration/deceleration (from 1 to 8) to enlarge the branching factor.

GENERATOR describes power generation by an engine which consumes fuel and may need (concurrent) refilling from various fuel tanks. The linear [12] and non-linear [3] versions both have turn-on/turn-off generator and start/stop refuelling actions but different refuelling dynamics. In the non-linear case, the rate of increase of the fuel level f caused by refuelling is given by $\frac{df}{dt} = \alpha \cdot t^2$ where α is a constant and t the refuelling time. Problem instances differ by increasing the number of tanks needed to achieve the goal of generating power for a given duration. Dead-ends appear whenever the planner delays refuelling for too long or when refueling is impossible due to the lack of space in the generator’s tank. As in previous work [3], the largest instance has 8 tanks, all of which are needed.

CONVOYS features a set of convoys that must reach specific locations starting from their initial positions. The time needed by convoy c to move between two positions a and b on the map depends on the speed v_c of the convoy and on the inverse of the square of the traffic $T_{a,b}$ in that specific road segment. There is a process per convoy and road segment, which updates the distance travelled according to $\frac{dD_{c,a,b}}{dt} = -\frac{v_c}{T_{a,b}^2}$ as long as the segment end point is not reached.

Instances in this domain involves up to 8 locations and 4 convoys.

INTERCEPT involves a vampire v and a bird b in the two-dimensional space \mathbb{Q}^2 . Two processes describe their movement, given as a function of the speed in the two dimensions. The task is to launch the vampire (initially stationary) at the right time and with

the right speed for it to intercept the bird. This happens when the Euclidean distance between the two is lower than a given radius r , which is encoded using the constraint $(x_v - x_b)^2 + (y_v - y_b)^2 \leq r^2$. Instances scale according to the initial distance between the vampire and the bird, and by the radius defining the goal condition.

HVAC is an abstraction of a ventilation, air-conditioning and cooling (HVAC) control problem [19]³. For each zone l in a building, two HVAC control parameters, the supply air flow rate a_l^{SA} and supply air temperature T_l^{sa} , must be adjusted from time to time so as to ensure that the zone temperature T_l meets given constraints. These constraints reflect the schedule of activities occurring at the zone. In our abstraction, there are instantaneous actions to increase/decrease T_l^{SA} and a_l^{SA} by a given amount. A single process updates the actual time, and a process per zone computes the rate of temperature change in this zone as $\frac{dT_l}{dt} = \beta \cdot a_l^{SA} \cdot (T_l^{SA} - T_l)$, where β is a constant. Time in this representation is implicitly captured by the progress of the plan, in contrast to the MIP formulation presented in [19], where each variable is explicitly located on the timeline. Moreover, since nonlinear optimisation solvers struggle with this problem, the MIP formulation linearises the bilinear terms in the above equation. Instances in this domain are generated by increasing the number of scheduled activities from 1 to 16.

COMPLEXPOURING is the problem of filling buckets using water tanks arranged in a complex directed network. Each tank and bucket has a given capacity and initial volume of water. This is a purely sequential problem which has no processes but just an action that enables the transfer of water from one container to a connected one. The transfer is modelled by Toriccelli’s law which states that the volume V of water left in a tank with an initial volume U of water after t seconds with an open tap is $V = (-kt - \sqrt{U})$, with $t \in (0, \frac{\sqrt{U}}{k})$, where k is a constant that depends on the cross-sectional area of the tank, the size of the tap, and gravity [18]. In our formulation this law is discretised to model the change happening after 1 second. In order to achieve the goal, we may have to transfer water from different tanks into intermediary ones, and so on. In this domain, we study the complexity arising from two different sources, leading to two different instance sets. In the first set, tanks arranged in a flat structure all feed into a single bucket and we vary the number of tanks from 2 to 10. In the second set, we vary the structure of the network of tanks, which includes from 3 to 10 tanks and 1 or 2 buckets to fill.

7.3 Summary of Results

Table 1 reports number of instances (I), coverage (C), run time (T), plan length (PL), and number of expanded nodes (Exp) for all the domains described in the previous section. The planner was able to solve all the instances provided.

Domain	I	C	T	PL	Exp
CAR	8	8	0.1	17.2	30.6
CAR (NL)	8	8	0.3	20.8	353.9
GENERATOR	8	8	3.7	1005.5	1006.5
GENERATOR (NL)	8	8	4.9	1005.5	1006.5
CONVOYS	6	6	25.1	23	935
INTERCEPT	10	10	1.5	114.4	2750.9
HVAC	16	16	16.5	110.6	9235.4
COMPLEXPOURING	17	17	4.9	14.1	1693

Table 1: Overall picture of the experimental results. PDDL+ domains ran with $\delta_t = 1$ (apart from INTERCEPT, which requires $\delta_t = 0.1$ to make the instances solvable). Time-out is set to 1800 seconds. Plan length includes both actions and waiting actions.

³ We only consider the control part of the problem and ignore external influences such as the temperature of adjacent rooms.

	With Heuristic Guidance				Blind Search		
	Exp	Eval	PL	T	Exp	PL	T
1	34	68	20	0.10	359209	20	96
2	73	148	20	0.14	NA	NA	NA
3	388	948	21	0.34	NA	NA	NA
4	448	1002	21	0.36	NA	NA	NA
5	472	1158	21	0.39	NA	NA	NA
6	472	1158	21	0.39	NA	NA	NA
7	472	1158	21	0.41	NA	NA	NA
8	472	1158	21	0.41	NA	NA	NA

(a) Non-Linear CAR

	With Heuristic Guidance				Blind Search		
	Exp	Eval	PL	T	Exp	PL	T
1	1003	1005	1002	2.70	NA	NA	NA
2	1004	1009	1003	3.05	NA	NA	NA
3	1005	1014	1004	3.77	NA	NA	NA
4	1006	1020	1005	4.60	NA	NA	NA
5	1007	1027	1006	5.01	NA	NA	NA
6	1008	1035	1007	5.93	NA	NA	NA
7	1009	1044	1008	6.86	NA	NA	NA
8	1010	1054	1009	7.65	NA	NA	NA

(b) Non-Linear GENERATOR

	With Heuristic Guidance				Blind Search		
	Exp	Eval	PL	T	Exp	PL	T
1	8	7	7	0.02	7	7	0.06
2	12	22	6	0.05	64	6	0.15
3	386	688	13	0.54	NA	NA	NA
4	1389	1651	15	1.85	NA	NA	NA
5	16	30	8	0.05	269	8	0.27
6	7	12	6	0.05	66	6	0.17
7	26	72	11	0.86	88661	11	30.2
8	204	742	14	1.03	NA	NA	NA
9	309	1131	22	1.01	NA	NA	NA
10	50	144	11	0.19	88563	11	19.7
11	429	1606	16	1.15	NA	NA	NA
12	408	1889	19	1.50	NA	NA	NA
13	1163	6097	21	3.25	NA	NA	NA
14	18356	109777	24	51.79	NA	NA	NA
15	5933	33500	22	19.38	NA	NA	NA
16	35	171	14	0.82	NA	NA	NA
17	50	363	10	1.22	NA	NA	NA

(c) COMPLEXPOURING

Table 2: Detailed results

The low average number of expanded nodes in the domain is indicative of heuristic effectiveness; the next section provides more details. Many domains have non-linear dependencies among numeric variables, and some require (implicit) temporal reasoning (e.g., GENERATOR, INTERCEPT). The heuristic is more inaccurate in these domains, especially HVAC which has many global constraints. Since AIBR is monotone, it does not capture negative effects on them.

7.4 In-Depth Analysis

Next, we present a more in-depth analysis of heuristic guidance for a subset of domains. AIBR informativeness and impact on plan length is evaluated comparing it with blind search ($f(n) = g(n)$, i.e. uniform-cost search). We used uniform-cost search rather than uninformed depth-first search since the latter only solved a few instances of one domain (with plans 10–100 times longer). Since, the search space is usually infinite, depth-first search often fails to terminate also for solvable instances.

Car domain. Table 2a shows the number of expanded (Exp) and evaluated nodes (Eval), plan length (PL) and run-time (T) for the non-linear instances. The increasing spectrum of possible accelerations in instances 1 to 8 only marginally affects ENHSP’s runtime. Blind search finds minimal-length plans, but is not able to scale up beyond instance 1. The plans produced using the heuristic, whilst suboptimal, are still of a good quality.

Non-linear Generator. Table 2b shows data for each instance. Here the number of necessary plan steps depends on the time the generator has to run, and on the number of tanks that must be used. With a

shortest plan length of about a 1000 actions and an average branching factor of 3 actions, even the smallest instance has a huge search space. Despite this, the AIBR heuristic solves all instances, whilst blind search solves none. Surprisingly, the heuristic overestimation favours states where fewer actions are possible (a tank can be used just once in this domain). This causes planner to favour refuelling over waiting, thus avoiding the generator running out of fuel.

Complex Pouring. Table 2c reports the data for each instance. Instances 1-9 belong to the set of instances with a flat network structure (see Subsection 7.2) The largest instance (instance 9) has a larger volume of liquid to be poured so as to increase the length of plans. Instances 10-16 involving a more complex network of tanks. The depth of the network affects the length of plans. Blind search was able to produce plans within the timeout but only for a subset of these instances. This is because, except for the instances requiring more than 11 actions, the state space is small enough to be blindly explored. Interestingly, none of the plans produced with the heuristics is longer than the plan produced by blind search. It turns out that, despite its inadmissibility, our heuristic leads to optimal plans in this domain.

ENHSP outperforms blind search on the other domains. Blind search only solved instances from the linear version of CAR (expanding on the average 387605 nodes), and 3 instances of CONVOYS (average expanded nodes: 10621).

7.5 Comparative Analysis

We compared ENHSP with dReal [3] and UPMurphi [8], both of which follow the planning via model-checking paradigm. Neither of them is strictly designed for sequential numeric planning, but they were the closest comparable planners that we were aware of and that were available at the time of writing. Another planner with similar capabilities was proposed by Bajada et al. [2], and two more systems for planning with non-linear processes have only recently become available [4, 24]. Examining their comparative performance, and that of ENHSP on newly proposed benchmarks [24], is a question for future work.

Table 3 reports the planners' runtimes on the CAR and GENERATOR domains. The UPMurphi runtimes were obtained by running the latest version of the planner⁴ in its standard configuration with a discretisation step of 1 sec, on the same PDDL+ formalisation the we use for ENHSP. The dReal runtimes are taken from Bryce et al.'s paper [3]. This is because dReal does not support PDDL with processes natively; a model-checking formulation of each instance has to be written manually, which includes a mode for each possible combination of processes (note that there could be an exponential number of modes to consider). We strived to make our PDDL+ formulations as close as possible to those in the dReal distribution⁵.

dReal uses some heuristic guidance, but the heuristic is intended to estimate the number of jumps between different modes, and ignores the numeric part of the problem. UPMurphi automatically translates PDDL+ to the model-checking formulation, but unlike dReal, it uses blind search and is not guided by any heuristic. Table 3 shows that the AIBR heuristic provides better search guidance than the mechanisms employed by these two planners.

However, these results should only be considered as indicative, rather than as a fair comparison. There are several differences between the three planners which must be taken into account. For instance, dReal is an approximate planner where preconditions and

Instance	1	2	3	4	5	6	7	8
CAR								
ENHSP	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
dReal	1.1	1.2	1.2	1.2	1.2	1.3	1.3	1.2
UPMur.	0.3	0.6	1.0	1.5	1.8	2.1	2.3	2.6
NL CAR								
ENHSP	0.1	0.1	0.3	0.4	0.4	0.4	0.4	0.4
dReal	16.7	16.7	16.3	16.8	16.6	16.8	17.4	16.6
UPMur.	4.9	42.7	84.3	113	139	171	187	204
GENERATOR								
ENHSP	1.9	2.6	2.7	3.1	4.0	4.4	5.1	5.6
dReal	3.1	15.6	134.7	1699	TO	TO	TO	TO
UPMur.	113.4	MO	MO	MO	MO	MO	MO	MO
NLGENERATOR								
ENHSP	2.7	3.1	3.8	4.6	5.0	5.9	6.9	7.7
dReal	12.8	71.6	1696	TO	TO	TO	TO	TO
UPMur.	128	MO	MO	MO	MO	MO	MO	MO

Table 3: Comparison of the run times (in sec.) of ENHSP, dReal and UPMurphi. Each instance can end with a solved situation, timeout (TO), or out of memory (MO).

goals are satisfied up to an approximation error. ENHSP and UPMurphi may also produce unsound plans, but for a different reason, due to discretisation of time and lack of zero-crossing check. To remedy this would require introducing a plan validation step, and subsequent refinement of the discretisation [8]. dReal only finds plan with a bounded number of steps, specified in advance, so in general must be run several times, with increasing step bounds. The CPU times in Table 3 are for a single run only, in which the planner is given the optimal number of steps for each problem. (CPU times for the whole process, on some instances, can be found in [4].) In contrast, ENHSP does not require any a priori bound.

8 Conclusions

The interval-based relaxation is the natural extension of the principle of monotonic (delete-free) relaxation to numeric planning [17], and the basis of most numeric planning heuristics [13, 6, 7, 5, 24]. However, these apply only to restricted problems, for example having linear conditions and effects. Up to now, it has not been clear if (asymptotic) reachability in the relaxation is even decidable for unrestricted problems, which may have cyclic dependencies [1]. We have shown that relaxed reachability is computable, through a novel asymptotic relaxed planning graph formulation, for problems with additive action effects, and moreover that non-additive state-dependent effects can be removed by a syntactic problem transformation, at the price of a further relaxation. The resulting additive interval-based relaxation (AIBR) and heuristics derived from it are pruning-safe.

Sequential numeric planning problems with complex non-linear effects arise naturally as time-discretisations of planning in the presence of autonomous continuous-time processes [8, 20], and efficient and general numeric planners are a key building block for supporting more expressive forms of planning. Evaluation of a planner using a heuristic obtained from the AIBR on various non-linear sequential planning problems and time-discretised PDDL+ problems with global constraints showed it to be competitive with some planners designed for such problems [3, 8].

In addition to extending the comparison to other recent planners, our future work is aimed at understanding how to make the heuristic more accurate and/or better integrated with the search engine used, for example through the extraction of helpful actions [17].

Acknowledgements This work is supported by ARC project DP140104219, "Robust AI Planning for Hybrid Systems". NICTA is funded by the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

⁴ Available at <http://github.com/gdellapenna/UPMurphi>

⁵ <http://github.com/danbryce/dreal>

REFERENCES

- [1] Johannes Aldinger, Robert Mattmüller, and Moritz Göbelbecker, ‘Complexity of interval relaxed numeric planning’, in *Proc. of KI 2015: Advances in Artificial Intelligence*, pp. 19–31, (2015).
- [2] Josef Bajada, Maria Fox, and Derek Long, ‘Temporal planning with semantic attachment of non-linear monotonic continuous behaviours’, in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pp. 1523–1529, (2015).
- [3] Daniel Bryce, Sicun Gao, David J. Musliner, and Robert P. Goldman, ‘Smt-based nonlinear PDDL+ planning’, in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 3247–3253, (2015).
- [4] Michael Cashmore, Maria Fox, Derek Long, and Daniele Magazzeni, ‘A compilation of the full PDDL+ language into SMT’, in *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, (2016).
- [5] Amanda Jane Coles, Andrew Coles, Maria Fox, and Derek Long, ‘Colin: Planning with continuous linear numeric change’, *Journal of Artificial Intelligence Research (JAIR)*, **44**, 1–96, (2012).
- [6] Amanda Jane Coles, Andrew Coles, Maria Fox, and Derek Long, ‘A hybrid LP-RPG heuristic for modelling numeric resource flows in planning’, *Journal of Artificial Intelligence Research (JAIR)*, **46**, 343–412, (2013).
- [7] Amanda Jane Coles, Andrew I. Coles, Maria Fox, and Derek Long, ‘Forward-chaining partial-order planning’, in *Proc. of International Conference on Automated Planning and Scheduling (ICAPS-10)*, (2010).
- [8] Giuseppe Della Penna, Daniele Magazzeni, Fabio Mercorio, and Benedetto Intrigila, ‘Upmurphi: A tool for universal planning on PDDL+ problems’, in *Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009, Thessaloniki, Greece, September 19-23, 2009*, (2009).
- [9] Patrick Eyerich, Robert Mattmüller, and Gabriele Röger, ‘Using the context-enhanced additive heuristic for temporal and numeric planning’, in *Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009, Thessaloniki, Greece, September 19-23, 2009*, (2009).
- [10] Maria Fox, Richard Howey, and Derek Long, ‘Validating plans in the context of processes and exogenous events’, in *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pp. 1151–1156, (2005).
- [11] Maria Fox and Derek Long, ‘Pddl2.1: An extension to pddl for expressing temporal planning domains’, *Journal of Artificial Intelligence Research*, **20**, 61–124, (2003).
- [12] Maria Fox and Derek Long, ‘Modelling mixed discrete-continuous domains for planning’, *J. Artif. Intell. Res. (JAIR)*, **27**, 235–297, (2006).
- [13] Alfonso Gerevini, Ivan Saetti, and Alessandro Serina, ‘An approach to efficient planning with numerical fluents and multi-criteria plan quality’, *Artificial Intelligence*, **172**(8-9), 899–944, (2008).
- [14] Peter Gregory, Derek Long, Maria Fox, and J. Christopher Beck, ‘Planning modulo theories: Extending the planning paradigm’, in *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling (ICAPS 2012)*, (2012).
- [15] Malte Helmert, ‘Decidability and undecidability results for planning with numerical state variables’, in *Proc. of International Conference on Artificial Intelligence Planning and Scheduling (AIPS 2002)*, pp. 44–53, (2002).
- [16] Malte Helmert and Hector Geffner, ‘Unifying the causal graph and additive heuristics’, in *Proc. of the 18th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 140–147, (2008).
- [17] Jörg Hoffmann, ‘The metric-ff planning system: Translating “ignoring delete lists” to numeric state variables’, *Journal of Artificial Intelligence Research (JAIR)*, **20**, 291–341, (2003).
- [18] Richard Howey and Derek Long, ‘VALs progress: The automatic validation tool for PDDL2.1 used in the international planning competition’.
- [19] BoonPing Lim, Menkes van den Briel, Sylvie Thiébaux, Scott Backhaus, and Russell Bent, ‘Hvac-aware occupancy scheduling’, in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 679–686, (2015).
- [20] Johannes Löhr, Patrick Eyerich, Thomas Keller, and Bernhard Nebel, ‘A planning based framework for controlling hybrid systems’, in *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling (ICAPS)*, (2012).
- [21] Johannes Löhr, Patrick Eyerich, Stefan Winkler, and Bernhard Nebel, ‘Domain predictive control under uncertain numerical state information’, in *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling (ICAPS)*, (2013).
- [22] Drew V. McDermott, ‘Reasoning about autonomous processes in an estimated-regression planner’, in *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS 2003), June 9-13, 2003, Trento, Italy*, pp. 143–152, (2003).
- [23] Ramon E. Moore, R. Baker Kearfott, and Michael J. Cloud, *Introduction to Interval Analysis*, SIAM, 2009.
- [24] Wiktor Piotrowski, Maria Fox, Derek Long, Daniele Magazzeni, and Fabio Mercorio, ‘Heuristic planning for PDDL+ domains’, in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), New York (NY), USA, July 9-15, (2016)*. Shorter version in Proc. AAAI-16 Workshop on Planning for Hybrid Systems.

Randomized Canonical Correlation Discriminant Analysis for Face Recognition

Bo Ma and Hui He and Hongwei Hu and Meili Wei¹

Abstract. As an important technique in multivariate statistical analysis, Canonical Correlation Analysis (CCA) has been widely used in face recognition. But existing CCA based face recognition methods need two kinds of expression for the same face sample, and usually suffers high computational complexity in dealing with large samples. In this paper, we present a supervised method called Randomized Canonical Correlation Discriminant Analysis (RCCDA) based on Randomized non-linear Canonical Correlation Analysis (RCCA) to make up for the shortage of CCA based face recognition methods. We first obtain basis vectors approximately with random features instead of the calculation of kernel matrix to improve the efficiency of computation, then we use these basis vectors to compute random optimal discriminant features which can reduce the dimension of face features while preserving as much discriminatory information as possible. The result of experiments on Extended Yale B, AR, ORL and FERET face databases demonstrates that the performance of our method compares favorably with some state-of-the-art algorithms.

1 Introduction

Face recognition, which is one of the most important fields in computer vision, has been widely used in military and commercial applications. In the past 40 years, many face recognition methods have been presented to solve the problem of the illumination, occlusion, expression and position diversification. But face recognition is still such a challenging and interesting problem that it has attracted researchers who have different backgrounds: psychology, pattern recognition, neural networks and computer vision[46].

According to the type of feature, face recognition methods can be classified into three categories: holistic matching methods, local matching methods and hybrid methods. In holistic matching methods, a whole face region is used as the raw input for a recognition system, such as eigenfaces (using Principal Component Analysis (PCA))[37], Fisherfaces (using Fisher linear Discriminant Analysis (FDA))[3], Kernel Principal Component Analysis (KPCA)[16], 2D-PCA[40], Local Preserving Projection (LPP)[23], sparsity preserving projections[28]. Although the holistic methods perform well under certain conditions, they need a large and representative training set to achieve high accuracy. Moreover, the misalignment of face image may degrade the performance of recognition.

Contrary to holistic matching methods, local matching methods first extract local features such as eyes, nose, and mouth, then put their locations and local statistics (geometric and/or appearance) into a

structural classifier. It has been proved that early local methods based on the geometric characteristics matching perform worse than global methods[4]. However, other local feature representations, which use patches instead to extract face features, has achieved promising performance in face recognition[11] because of their robustness on illumination, expression and occlusion, such as Gabor wavelets[1], Local Binary Patterns (LBP)[2] and Local Ternary Patterns (LTP)[36], local gabor binary pattern[45], local graph structure[13].

Hybrid methods combine the advantages of the holistic feature representation and local feature representation. Similar to human perception system, both local features and the whole face region are used to recognize a face in machine recognition system, such as Gabor+FDA[21], Gabor+KPCA[20], LBP+KPCA[35]. Considering the superiority of hybrid method, in this paper, we use this method to obtain more local discriminative information and complete facial representation.

CCA, which is first applied in data mining, aims to find the correlation between two sets of data. After the establishment of CCA framework for image recognition presented by Sun et al. [32], multi-modal identification face recognition methods based on CCA have been attracting extensive attention. Many algorithms have been proposed to improve the performance of CCA, for example, KPCA+CCA[10], generalized CCA[31], discriminative CCA[34], 2D-CCA[17].

Although CCA based face recognition methods achieve good results, there still exists some drawbacks: 1) CCA needs two views of the same face. In the methods mentioned above, they extract two types of feature to compute CCA, then fuse features to train classifiers. However, the chosen features and the fusion methods are hand-crafted, which are hard to be optimized. 2) High computational cost. In the process of solving small sample size problem, CCA generally requires suitable methods to reduce the sample dimension such as PCA, which needs extra calculation. Kernel Canonical Correlation Analysis (KCCA) can project visual features into higher space by using kernel tricks to deal with the data involving non-linear correlation, it has a high computational complexity.

Randomization, which can yield comparable generalization performance at a fraction of the computational cost, has recently been considered as an alternative to optimize kernel methods[39]. Random features can be used to approximate kernel matrix when the scale of sample is very large, instead of intractable kernel computation.

As an effective feature in face recognition, Fisher optimal discriminant vectors, which can reduce the dimension of face features to a more manageable size while preserving more discriminatory information before classification, can be obtained by FDA. But in order to find out the optimal non-linear discriminant function, FDA still has high computational cost.

This paper presents a new method for face recognition, Random-

¹ Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. email: {bma000, hehui, huhongwei}@bit.edu.cn, meiliwei1314@163.com.

ized Canonical Correlation Discriminant Analysis (RCCDA), that combines the advantages of stochastic methods and FDA to address both problems of CCA simultaneously. We first extract local feature as a representation of face image, which is more robust for illumination and occlusion, then use RCCDA to obtain random optimal discriminant features, which reduces computational complexity and preserves discriminatory information as much as possible at the same time.

The method consists two main contributions. First, we bridge a gap between KCCA and Kernel Fisher Discriminant Analysis (KFDA)[30] called Kernel Canonical Correlation Discriminant Analysis (KCCDA), thus we can obtain optimal discriminant vectors more efficiently with only a set of labeled samples, which is easy to be optimized. Second, we use Randomized non-linear CCA (RCCA)[22] to achieve a similar accuracy to KCCDA with greatly computational speed-ups (more than 10 times).

2 Random Optimal Discriminant Features

In this section, we first review the CCA and KCCA method, then show the details of KCCDA and introduce randomized method with it to form RCCDA.

2.1 Kernel Canonical Correlation Analysis

CCA is introduced by Hotelling[12] to describe the linear relation between two multi-dimensional(or two sets of) variables. The problem of CCA is finding basis vectors for each set such that the projections of the two variables on their respective basis vectors are maximally correlated.

In another words, CCA processes two different views of the same object, such as speech audio signals, paired video frames and two viewing images of the same object, and returns their maximally correlated linear transformations[18].

In the form of matrix, given two sets $\mathbf{X} \in \mathbb{R}^{n \times d_x}$ and $\mathbf{Y} \in \mathbb{R}^{n \times d_y}$, the CCA projections $\mathbf{a} \in \mathbb{R}^{d_x \times k}$ and $\mathbf{b} \in \mathbb{R}^{d_y \times k}$ are the solution to

$$\begin{aligned} \max \quad & \mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b} \\ \text{s.t.} \quad & \mathbf{a}^T \mathbf{X}^T \mathbf{X} \mathbf{a} = 1 \\ & \mathbf{b}^T \mathbf{Y}^T \mathbf{Y} \mathbf{b} = 1 \end{aligned} \quad (1)$$

The canonical correlations ρ_1, \dots, ρ_k and basis vectors \mathbf{a} and \mathbf{b} form the eigensystem of the generalized eigenvalue problem[7]:

$$\begin{aligned} & \begin{pmatrix} \mathbf{0} & \mathbf{C}_{XY} \\ \mathbf{C}_{YX} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} = \\ \rho \begin{pmatrix} \mathbf{C}_{XX} + \gamma_x \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{YY} + \gamma_y \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \end{aligned} \quad (2)$$

where \mathbf{C}_{XY} is the covariance $\text{cov}(\mathbf{X}, \mathbf{Y})$, $\gamma_x \mathbf{I}$, $\gamma_y \mathbf{I}$ are regularization terms.

To extend CCA to non-linear mappings, the data can be mapped to a new feature space via function Φ , $\Phi_x: \mathbb{R}^{n \times d_x} \rightarrow \mathbb{R}^{n \times p}$, $\Phi_y: \mathbb{R}^{n \times d_y} \rightarrow \mathbb{R}^{n \times q}$. In this new feature space, the function need to be maximized is:

$$\begin{aligned} \max \quad & \mathbf{a}^T \Phi_x(\mathbf{X})^T \Phi_y(\mathbf{Y}) \mathbf{b} \\ \text{s.t.} \quad & \mathbf{a}^T \Phi_x(\mathbf{X})^T \Phi_x(\mathbf{X}) \mathbf{a} = 1 \\ & \mathbf{b}^T \Phi_y(\mathbf{Y})^T \Phi_y(\mathbf{Y}) \mathbf{b} = 1 \end{aligned} \quad (3)$$

According to the theory of reproducing kernel, we can use $\mathbf{a} = \Phi_x(\mathbf{X})^T \mathbf{c}$ and $\mathbf{b} = \Phi_y(\mathbf{Y})^T \mathbf{d}$ to replace \mathbf{a} , \mathbf{b} in the equation above.

Let $\mathbf{K} \in \mathbb{R}^{n \times n}$ be the kernel matrix and $[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. Thus, the function needed to be maximized is:

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{K}_x \mathbf{K}_y \mathbf{d} \\ \text{s.t.} \quad & \mathbf{c}^T \mathbf{K}_x \mathbf{K}_x \mathbf{c} = 1 \\ & \mathbf{d}^T \mathbf{K}_y \mathbf{K}_y \mathbf{d} = 1 \end{aligned} \quad (4)$$

This problem can also be translated to the generalized eigenvalue problem:

$$\begin{aligned} & \begin{pmatrix} \mathbf{0} & \mathbf{K}_x \mathbf{K}_y \\ \mathbf{K}_y \mathbf{K}_x & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix} = \\ \rho \begin{pmatrix} \mathbf{K}_x \mathbf{K}_x + \gamma_x \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_y \mathbf{K}_y + \gamma_y \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix} \end{aligned} \quad (5)$$

where $\mathbf{K}_x, \mathbf{K}_y$ are kernel matrix of \mathbf{X}, \mathbf{Y} , (γ_x, γ_y) are positive regularizers mandatory to avoid spurious ± 1 correlations and \mathbf{c}, \mathbf{d} are basis vectors in feature space.

2.2 Kernel Canonical Correlation Discriminant Analysis

In this section, we will give a brief description of KFDA to help clarify the relationship between KFDA and KCCA, then combine it with randomized method.

FDA[9] is a variance preserving approach with the goal of finding the optimal linear discriminant function. By extending FDA to non-linear mappings, KFDA maps the data into a high dimensional feature space, then executes linear discriminant analysis in it, which is equal to execute non-linear discriminant analysis in original space.

The objective function of FDA is designed to be maximized by a projection that maximizes the inter class scatter and minimizes the intra class scatter. An optimal discriminant vector ω can be obtained by solving the following:

$$\max \frac{\omega^T \mathbf{S}_b \omega}{\omega^T \mathbf{S}_w \omega} \quad (6)$$

where \mathbf{S}_b is the inter scatter matrix and \mathbf{S}_w is the total intra scatter matrix of data.

After mapped into high-dimensional space by function Φ and replaced the dot product in the new feature space with a kernel function, the objective function transforms to:

$$\max \frac{\omega^T \mathbf{S}_b^\Phi \omega}{\omega^T \mathbf{S}_w^\Phi \omega} \quad (7)$$

where \mathbf{S}_b^Φ and \mathbf{S}_w^Φ are between-class and within covariance matrix in feature space.

In order to replace the dot product in the new feature space with a kernel function, we use $\alpha^T \Phi(\mathbf{X})$ to substitute ω . The numerator of objective function can then be written as:

$$\begin{aligned} \omega^T \mathbf{S}_b^\Phi \omega &= \alpha^T \mathbf{K}_b \alpha \\ \mathbf{K}_b &= \sum_{k=1}^c \frac{n_k}{n} \mathbf{U}_k \mathbf{U}_k^T = \frac{1}{n} \mathbf{K} \mathbf{H} \mathbf{K}^T \\ \mathbf{U}_k &= \left(\frac{1}{n_k} \sum_{i=1}^{n_k} k(\mathbf{x}_1, \mathbf{x}_{ki}), \dots, \frac{1}{n_k} \sum_{i=1}^{n_k} k(\mathbf{x}_n, \mathbf{x}_{ki}) \right)^T \end{aligned} \quad (8)$$

where α is optimal discriminant vector, $[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel matrix, $\mathbf{H} = \text{diag}(\mathbf{H}_1 \dots \mathbf{H}_c)$ and $\mathbf{H}_k = \frac{1}{n_k} \mathbf{I}_{n_k \times n_k}$ ($k = 1, \dots, c$), n_k the number of samples in class k .

Similarly, the denominator can be written as:

$$\begin{aligned} \omega^T S_w^\Phi \omega &= \alpha^T K_w \alpha \\ K_w &= \frac{1}{n} (K K^T - K H K^T) \end{aligned} \quad (9)$$

We can solve the problem of KFDA by the generalized eigenvalue problem:

$$K_b \alpha = \lambda K_w \alpha \quad (10)$$

Substituting K_b and K_w with K and H in equation above, the goal of KFDA becomes:

$$K H K \alpha = \frac{\lambda}{1 + \lambda} K K \alpha \quad (11)$$

Reviewing the solution of KCCA in Eq. (5), with regularizers ignored, the base vectors c can be calculate as an eigenvalue problem:

$$K_x K_y (K_y K_y)^\dagger K_y K_x c = \rho^2 K_x K_x c \quad (12)$$

where \dagger indicates a generalized inverse.

Let \mathbf{X} be the $n \times p$ matrix of predictor variables, and \mathbf{Y} be the $n \times c$ matrix of dummy response variables encoded by one-of- k label[14]:

$$\mathbf{Y} = \begin{pmatrix} \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \mathbf{0}_{n_1} & \cdots & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \mathbf{0}_{n_2} & \cdots & \mathbf{0}_{n_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{n_k} & \mathbf{0}_{n_k} & \mathbf{1}_{n_k} & \cdots & \mathbf{0}_{n_k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{n_c} & \mathbf{0}_{n_c} & \mathbf{0}_{n_c} & \cdots & \mathbf{1}_{n_c} \end{pmatrix}_{n \times c} \quad (13)$$

where $\mathbf{1}_{n_k}$ and $\mathbf{0}_{n_k}$ are column vectors with n_k number of 1 and 0 respectively, n_k the number of samples in class k .

Considering non-linear mapping $\Phi_x : \mathbf{x} \rightarrow \Phi_x(\mathbf{x})$ and self-mapping $\Phi_y : \mathbf{y} \rightarrow \mathbf{y}$, the canonical direction c of X can be obtain by solving KCCA. In this example, K_x is a kernel matrix and $K_y = \mathbf{Y}\mathbf{Y}^T$. Replacing these in Eq. (12), the equation can be rewritten as:

$$K_x \mathbf{Y}\mathbf{Y}^T (\mathbf{Y}\mathbf{Y}^T \mathbf{Y}\mathbf{Y}^T)^\dagger \mathbf{Y}\mathbf{Y}^T K_x c = \rho^2 K_x K_x c \quad (14)$$

The key point is that the matrix $\mathbf{Y}\mathbf{Y}^T (\mathbf{Y}\mathbf{Y}^T \mathbf{Y}\mathbf{Y}^T)^\dagger \mathbf{Y}\mathbf{Y}^T$ is equal to matrix H defined in Eq. (8). Then the equation above transforms to:

$$K_x H K_x c = \rho^2 K_x K_x c \quad (15)$$

It is easy to find the equivalency of Eq. (11) and Eq. (15) by setting $\rho^2 = \frac{\lambda}{1+\lambda}$. This shows that the vector c giving the maximum correlation also maximizes the ratio of between class sum of squares to within class sum of squares of the projected data. Thus, we can use KCCA to compute the optimal discriminant direction of KFDA, which is called KCCDA method.

2.3 Randomized Canonical Correlation Discriminant Analysis

In order to handle the high complexity of KCCA, RCCA has been presented in [22] as a low-rank approximation of KCCA. Thus, comparable generalization performance can be yielded by selecting the appropriate scale random features. By using RCCA method, RCCDA has been proposed as an approximation of KCCDA.

The idea of random features is to define a lower-dimensional mapping, $z : \mathbb{R}^d \rightarrow \mathbb{R}^m$ through a random sampling scheme such that

$[K]_{ii'} \approx z(\mathbf{x}_i)^T z(\mathbf{x}_{i'})$ [39]. Thus, using random features, non-linear functions with respect to \mathbf{x} can be learned as linear functions in $z(\mathbf{x})$ leading to significant computational speed-ups[25].

Fix $m \ll n$ and randomly(uniformly) sample a subset $M = \{\hat{\mathbf{x}}_i\}_{i=1}^m$ of m points from the data $\{\mathbf{x}_i\}_{i=1}^n$. Let \tilde{K} denote the Gram matrix $[\tilde{K}]_{ii'}$ where $i, i' \in M$. The Nyström method[43], which uses random subsampling to approximate the Gram matrix, constructs a low-rank approximation to the Gram matrix as

$$K \approx \tilde{K} = \sum_{i=1}^n \sum_{i'=1}^n z(\mathbf{x}_i)^T z(\mathbf{x}_{i'}) \quad (16)$$

$$z(\mathbf{x}_i) = \hat{D}^{-\frac{1}{2}} \hat{V}^T [\kappa(\mathbf{x}_i, \hat{\mathbf{x}}_1), \dots, \kappa(\mathbf{x}_i, \hat{\mathbf{x}}_m)]^T$$

where $z(\mathbf{x}_i)$ is random feature, the columns of \hat{V} are the eigenvectors of \tilde{K} , \hat{D} is a diagonal matrix whose entries are the corresponding eigenvalues. Constructing features in this way reduces the time complexity of learning a non-linear prediction function from $O(n^3)$ to $O(n)$ [8].

Let \tilde{K}_x and \tilde{K}_y , which are obtained by random method, be the approximations to K_x and K_y (as defined in Eq. (5)). The objective function of RCCA is:

$$\begin{aligned} \max \quad & c^T \tilde{K}_x \tilde{K}_y d \\ \text{s.t.} \quad & c^T \tilde{K}_x \tilde{K}_x c = 1 \\ & d^T \tilde{K}_y \tilde{K}_y d = 1 \end{aligned} \quad (17)$$

Replacing \tilde{K}_x, \tilde{K}_y with $z_x(\mathbf{X})^T z_x(\mathbf{X}), z_y(\mathbf{Y})^T z_y(\mathbf{Y})$ and let $\mathbf{a} = z_x(\mathbf{X})c, \mathbf{b} = z_y(\mathbf{Y})d$, then what we need to maximize is

$$\begin{aligned} \max \quad & \mathbf{a}^T z_x(\mathbf{X}) z_y(\mathbf{Y})^T \mathbf{b} \\ \text{s.t.} \quad & \mathbf{a}^T z_x(\mathbf{X}) z_x(\mathbf{X})^T \mathbf{a} = 1 \\ & \mathbf{b}^T z_y(\mathbf{Y}) z_y(\mathbf{Y})^T \mathbf{b} = 1 \end{aligned} \quad (18)$$

It is easy to find that we can compute the RCCA(\mathbf{X}, \mathbf{Y}) just by CCA($z_x(\mathbf{X}), z_y(\mathbf{Y})$). That is to say, RCCA can be understood as linear CCA performed on a pair of randomized non-linear mappings: $z_x : \mathbb{R}^{n \times p} \rightarrow \mathbb{R}^{n \times m_x}, z_y : \mathbb{R}^{n \times q} \rightarrow \mathbb{R}^{n \times m_y}$ of the data $\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{Y} \in \mathbb{R}^{n \times q}$. Schematically,

$$\begin{aligned} \text{RCCA}(\mathbf{X}, \mathbf{Y}) &= \text{CCA}(z_x(\mathbf{X}), z_y(\mathbf{Y})) \\ &\approx \text{KCCA}(\mathbf{X}, \mathbf{Y}). \end{aligned} \quad (19)$$

As mentioned in section 2.2, we can use KCCA to calculate KFDA. Similar to KCCA, RCCA can be used to accelerate the computation of KFDA. As an approximation of KCCDA, the equation of RCCDA defined as:

$$\begin{aligned} \text{RCCDA}(\mathbf{X}_y) &= \text{RCCA}(\mathbf{X}, \mathbf{Y}) \\ &= \text{CCA}(z_x(\mathbf{X}), z_y(\mathbf{Y})) \\ &\approx \text{KCCDA}(\mathbf{X}_y) \\ &= \text{KFDA}(\mathbf{X}_y) \end{aligned} \quad (20)$$

where \mathbf{Y} is a hand-crafted class matrix, \mathbf{X}_y means \mathbf{X} depend on \mathbf{Y} implicitly. z_x is a randomized non-linear mappings: $z_x : \mathbb{R}^{n \times p} \rightarrow \mathbb{R}^{n \times m_x}$ and z_y is a self-mapping, $z_y : \mathbf{y} \rightarrow \mathbf{y}$.

Algorithm 1 details our method to obtaining random optimal discriminant features, which can reduce the dimension of face features while preserving as much discriminatory information as possible. The setting is that we have labeled data $\{\mathbf{X}, \mathbf{y}\}$ and dimension of random feature m .