and $\frac{1}{2}max(X_1 \cup X_{1,2})' < \frac{1}{2}max(X_1 \cup X_{1,2})$. Therefore $y_1' = \frac{1}{2}max(X_1 \cup X_{1,2})' < \frac{1}{2}max(X_1 \cup X_{1,2}) = y_1$, also implying an increase in his cost. Therefore no agent has the incentive to lie.

Case 3. $p_1 = \{F_2\}, p_n = \{F_1\}$. It is not hard to see that we have the same result as Case 2 by symmetry.

Therefore Mechanism 3 is strategy-proof. □

Theorem 14 *Mechanism 3 is optimal.*

Proof. We will analyze by the same cases in the proof for Theorem 13. For Case 1, the maximum cost comes from agent 1 or agent n and we have the maximum cost $mc = L/2$. We can see that any other output would have the new maximum cost $mc' > L/2 = mc$. For Case 2, the maximum cost mc comes from agent 1 and the agent at $max(X_1 \cup X_{1,2})$, or agent n and the agent at $min(X_2 \cup X_{1,2})$. We will focus on the former one as the analysis for the latter one would be similar. Assume there is a better output denoted as y_1' and y_2'. If $y_1' > y_1$, we can see the cost of agent 1 increases, which means that the maximum cost increases. If $y_1' < y_1$, indeed the cost of agent 1 decreases. However for the agent at $max(X_1 \cup X_{1,2})$ we would have his cost mc' be the maximum cost and $mc' = max(X_1 \cup X_{1,2}) - y_1' > \frac{1}{2}max(X_1 \cup X_{1,2}) = mc$, which means that the maximum cost also increases. Therefore no such output exists and Mechanism 3 is optimal. □

4.2 Sum cost

For minimizing sum cost, we will present a strategy-proof mechanism with approximation ratio of 2.

Mechanism 4 *Given a profile, locate F_1 at the median location of $X_1 \cup X_{1,2}$ and F_2 at the median location of $X_2 \cup X_{1,2}$.*

Theorem 15 *Mechanism 4 is strategy-proof.*

Proof. Let agent i be the lying agent. If he has single preference, from Mechanism 4 we can see that he cannot change the output for his preferred facility by lying to have preference $\{F_1, F_2\}$, and can only possibly push it away by lying to have preference for the other facility since the mechanism is based on median location. If he has optional preference, by lying to have preference $\{F_1\}$, he can only possibly push F_2 away from him and cannot change the location of F_1, which does not reduce his cost. Lying to have preference F_2 will have similar effect. Therefore, no agent has the incentive to lie and Mechanism 4 is strategy-proof. □

To prove the approximation ratio of Mechanism 4, let us first investigate a different model, referred to as the *fixed preference model*. In this model, the cost of agent i is defined as $cost_i = \sum_{F_k \in p_i} d(i, F_k)$. Without considering the truthfulness, we can also apply Mechanism 4 to this model. And we will have the following lemma.

Lemma 16 *Mechanism 4 is optimal for the fixed preference model.*

Proof. Given that the cost of agents with preference $\{F_1, F_2\}$ depends on the sum of distances from both facilities in *fixed preference model*, we can split them into an agent with preference $\{F_1\}$ and an agent with preference $\{F_2\}$ located at the same place and we will have a new profile containing only agents with preference for a single facility. It is not hard to see that the new profile is equivalent to the original profile for analyzing the cost. Meanwhile, for any two agents with preference $\{F_1\}$, the sum of their cost is minimized when $\{F_1\}$ is located between them, which is the distance between them. Combining every pair of agents with preference $\{F_1\}$ from out most to the center, we can see that the sum cost is minimized when $\{F_1\}$ is located at the median location of all agents with preference $\{F_1\}$, which is exactly the output of Mechanism 4. The analysis for agents with preference $\{F_2\}$ is similar. Therefore, Mechanism 4 is optimal for the fixed preference model. □

Now we present the proof for the approximation ratio of Mechanism 4.

Theorem 17 *Mechanism 4 is 2-approximation.*

Proof. For our optional preference (Max) model, we have the cost as:

$$sc_{Max} = \sum_{k \in \{1,2\}} \sum_{p_i = \{F_k\}} d(i, F_k) + \sum_{p_i = \{F_1, F_2\}} \max(d(i, F_1), d(i, F_2))$$

while for the *fixed preference model*, we have the cost as

$$sc_{Sum} = \sum_{k \in \{1,2\}} \sum_{p_i = \{F_k\}} d(i, F_k) + \sum_{p_i = \{F_1, F_2\}} (d(i, F_1) + d(i, F_2))$$

Since $sc_{Max}(\mathbf{s}) \geq \sum_{p_i = \{F_1, F_2\}} \min(d(i, F_1), d(i, F_2))$ we have:

$$2sc_{Max} \geq sc_{Sum}$$

Based on this equation, given a profile, let $\mathbf{s}$ be the solution returned by Mechanism 4, and $\mathbf{s}^*$ be the optimal solution, we now show that $sc_{Max}(\mathbf{s}) \leq 2sc_{Max}(\mathbf{s}^*)$.

By Lemma 16 , we have $sc_{Sum}(\mathbf{s}) \leq sc_{Sum}(\mathbf{s}^*)$. Therefore

$$sc_{Max}(\mathbf{s}) \leq sc_{Sum}(\mathbf{s}) \leq sc_{Sum}(\mathbf{s}^*) \leq 2sc_{Max}(\mathbf{s}^*)$$

Eventually, the theorem is proved.

□

5 CONSLUSION AND DISCUSSION

We studied the two heterogeneous facility location game with optional preference and we mainly focused on deterministic mechanisms. This is a new model which covers more real life scenarios. A table summarizing our results is listed below.

Table 1. A summary of our results.

Variant	Objective	Upper Bound	Lower Bound
Min	maximum cost	2	4/3
	sum cost	$n/2+1$	2
Max	maximum cost	1 (optimal)	
	sum cost	2	

Besides the above results, we also found that the approximation ratio can be better if randomized mechanisms are allowed. For example, the ratio of the Min variant for maximum cost can be lowered to 3/2 and is likely to be a constant for sum cost. On the other hand, in our setting the two facilities can be located at any point on the continuous line and can be located together, which is well justified. However, it is also an interesting direction to study the case when facilities cannot be put on the same point. Some of the mechanisms proposed in our paper can be applied to that discrete case. For example, for the Min variant minimizing sum cost, the mechanism we proposed will never locate the two facilities together unless all agents are located together, and this mechanism could potentially be extended to the k-facility model.

Acknowledgment

This research was partially supported by NSFC 61433012 and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 117913].

REFERENCES

[1] Noga Alon, Michal Feldman, Ariel D Procaccia, and Moshe Tennenholtz, 'Strategyproof approximation of the minimax on networks', *Mathematics of Operations Research*, **35**(3), 513–526, (2010).
[2] Yukun Cheng, Wei Yu, and Guochuan Zhang, 'Mechanisms for obnoxious facility game on a path', in *Combinatorial Optimization and Applications*, 262–271, Springer, (2011).
[3] Yukun Cheng, Wei Yu, and Guochuan Zhang, 'Strategy-proof approximation mechanisms for an obnoxious facility game on networks', *Theoretical Computer Science*, **497**, 154–163, (2013).
[4] Elad Dokow, Michal Feldman, Reshef Meir, and Ilan Nehama, 'Mechanism design on discrete lines and cycles', in *Proceedings of the 13th ACM Conference on Electronic Commerce*, pp. 423–440. ACM, (2012).
[5] Itai Feigenbaum and Jay Sethuraman, 'Strategyproof mechanisms for one-dimensional hybrid and obnoxious facility location models', in *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, (2015).
[6] Michal Feldman and Yoav Wilf, 'Strategyproof facility location and the least squares objective', in *Proceedings of the fourteenth ACM conference on Electronic commerce*, pp. 873–890. ACM, (2013).
[7] Aris Filos-Ratsikas, Minming Li, Jie Zhang, and Qiang Zhang, 'Facility location with double-peaked preferences', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, eds., Blai Bonet and Sven Koenig, pp. 893–899. AAAI Press, (2015).
[8] Dimitris Fotakis and Christos Tzamos, 'Winner-imposing strategyproof mechanisms for multiple facility location games', in *Internet and Network Economics*, 234–245, Springer, (2010).
[9] Dimitris Fotakis and Christos Tzamos, 'On the power of deterministic mechanisms for facility location games', *ACM Transactions on Economics and Computation*, **2**(4), 15, (2014).
[10] Pinyan Lu, Xiaorui Sun, Yajun Wang, and Zeyuan Allen Zhu, 'Asymptotically optimal strategy-proof mechanisms for two-facility games', in *Proceedings of the 11th ACM conference on Electronic commerce*, pp. 315–324. ACM, (2010).
[11] Pinyan Lu, Yajun Wang, and Yuan Zhou, 'Tighter bounds for facility games', in *Internet and Network Economics*, 137–148, Springer, (2009).
[12] Hervé Moulin, 'On strategy-proofness and single peakedness', *Public Choice*, **35**(4), 437–455, (1980).
[13] Ariel D. Procaccia and Moshe Tennenholtz, 'Approximate mechanism design without money', in *Proceedings of the 10th ACM Conference on Electronic Commerce*, EC '09, pp. 177–186, New York, NY, USA, (2009). ACM.
[14] Paolo Serafino and Carmine Ventre, 'Heterogeneous facility location without money on the line'. ECAI, (2014).
[15] Paolo Serafino and Carmine Ventre, 'Truthful mechanisms without money for non-utilitarian heterogeneous facility location', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, eds., Blai Bonet and Sven Koenig, pp. 1029–1035. AAAI Press, (2015).
[16] Olivier Spanjaard, Fanny Pascual, Nguyen Kim Thang, Laurent Gourvès, and Bruno Escoffier, 'Strategy-proof mechanisms for facility location games with many facilities', (2011).
[17] Deshi Ye, Lili Mei, and Yong Zhang, 'Strategy-proof mechanism for obnoxious facility location on a line', in *Computing and Combinatorics*, 45–56, Springer, (2015).
[18] Qiang Zhang and Minming Li, 'Strategyproof mechanism design for facility location games with weighted agents on a line', *Journal of Combinatorial Optimization*, **28**(4), 756–773, (2014).
[19] Shaokun Zou and Minming Li, 'Facility location games with dual preference', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, eds., Gerhard Weiss, Pinar Yolum, Rafael H. Bordini, and Edith Elkind, pp. 615–623. ACM, (2015).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1528

Iterative Judgment Aggregation

Marija Slavkovik[1] and **Wojciech Jamroga**[2]

Abstract. Judgment aggregation problems form a class of collective decision-making problems represented in an abstract way, subsuming some well known problems such as voting. A collective decision can be reached in many ways, but a direct one-step aggregation of individual decisions is arguably most studied. Another way to reach collective decisions is by iterative consensus building – allowing each decision-maker to change their individual decision in response to the choices of the other agents until a consensus is reached. Iterative consensus building has so far only been studied for voting problems. Here we propose an iterative judgment aggregation algorithm, based on movements in an undirected graph, and we study for which instances it terminates with a consensus. We also compare the computational complexity of our itterative procedure with that of related judgment aggregation operators.

1 Introduction

Social choice aggregation methods, such as voting [32], are of interest to artificial intelligence as methods for collective decision-making among humans and automated agents alike [3]. Judgment aggregation problems [26] are problems of aggregating individual judgments on a fixed set of logically related issues, called an agenda. Intuitively, an issue is a question that can be answered "yes" or "no." Alternatively, an issue is a proposition that can be accepted or rejected. A judgment is a consistent collection of "yes" and "no" answers, one for each issue. Judgment aggregation has been used to model collective decision-making in multi-agent systems [2, 39, 41]. It is also interesting because it generalises voting problems, i.e., problems of choosing one option from a set of available options by aggregating agents' individual preferences over these options. A voting problem can be represented as a judgment aggregation problem under some mild conditions, see e.g. [6, 23].

Aggregation methods produce a joint decision for a group of agents by aggregating the set of their individual decisions, called a profile, using an aggregation operator. Another approach to collective decision-making is deliberation, when agents negotiate which decisions to make. In multi-agent systems, deliberation procedures are constructed using an abstract argumentation framework to model relations between decisions, cf. e.g. [35, 20]. A third, comparatively less explored method to reach collective decisions is by *iterative consensus building*: each agent starts with an individual decision which she then iteratively changes in response to the individual decisions of the other agents until all agents end up supporting the same decision, i.e., until a *consensus* is reached. While in standard aggregation all individual decisions are elicited once, forming a profile, and after the elicitation the agents can no longer change the profile, an iterative procedure allows agents to change their decisions many times, even back and forth.

The existence of judgment transformation functions, i.e., functions that transform one profile of individual judgments into another profile of individual judgments (possibly towards consensus) has been considered by List [25]. It was shown that under a set of reasonable and minimal desirable conditions no transformation function can exist. Social choice aggregation theory is rife with impossibility results such as this, yet few specific aggregation operators (that violate some of the desirable conditions) are proposed. There are more voting operators than judgment aggregation operators, which is unsurprising since voting is a much older discipline, but the number of judgment aggregation operators is also on the rise. Those include: quota-based rules [7, 9], distance-based rules [11, 13, 14, 28, 37, 40], generalisations of Condorcet-consistent voting rules [21, 29, 30], and rules based on the maximisation of some scoring function [5, 21, 42]. Deliberation and iterative consensus reaching procedures for voting problems have been explored, e.g., in [27, 24, 17, 18, 34]. However, to the best of our knowledge, there are no iterative procedures for aggregating judgments. With this work we aim to fill in the gap.

We consider all possible judgments for an agenda as vertices in a graph. The existence of an edge between judgments in the graph depends on the relations between the truth-value assignments on the same issue in the connected judgments. We define three intuitive agenda graphs. We design an iterative consensus building algorithm which reaches consensus in the following way: In the first step of the algorithm, each agent chooses a vertex and lets the other agents know what she has chosen. In each subsequent step each agent independently from the other agents moves to an adjacent vertex if this move reduces her path distance to the other vertices in the profile. The agents are only allowed to move along a shortest path towards some other agent. The moving continues until a consensus is reached (i.e., when all agents end up on the same vertex). We then exploit properties of graphs to study for which subgraphs corresponding to a profile of judgments the algorithm terminates with a consensus.

Judgment aggregation operators suffer from two shortcomings. First, they are often irresolute, i.e., more than one collective decision is produced Unlike in voting, tie-breaking in judgment aggregation is not straightforward and virtually unexplored. Secondly, deciding if a given judgment is among the possible "winners" of the aggregation is often intractable [23, 12]. The set of tractable aggregation functions is very limited, exceptions being [7, 9, 13]. An iterative procedure clearly avoids ties when it reaches a consensus, and this is one advantage of our proposal. We also show that our consensus-oriented procedure may offer some computational benefits when compared to standard judgment aggregation rules.

The motivation for our iterative procedure is both descriptive and prescriptive. On one hand, our algorithm is meant to approximate

[1] Department of Information Science and Media Studies, University of Bergen, Norway, email: Marija.Slavkovik@infomedia.uib.no
[2] Institute of Computer Science, Polish Academy of Sciences, email: w.jamroga@ipipan.waw.pl

	φ_1	φ_2	φ_3
agent 1	0	1	0
agent 2	1	0	0
agent 3	1	1	1
majority	1	1	0

Figure 1. Doctrinal paradox

consensus formation that happens in human societies. On the other hand, our procedure can be useful for implementing artificial agents, as producing a consensual judgment is in some cases distinctly cheaper than computing the collective opinion in one step elicitation by a standard judgment aggregation procedure.

The paper is structured as follows. In Section 2 we introduce the judgment aggregation framework. In Section 3 we define three relevant agenda graphs and recall some useful concepts from graph theory. In Section 4 we present the algorithm for iterative judgment aggregation, and discuss necessary conditions for its termination with a consensus. In Section 5 we investigate sufficient termination conditions for each of the three agenda graphs. In Section 6 we briefly discuss the quality of the reached consensus with respect to some judgment aggregation operators, and study the computational complexity of the algorithm. In Section 7 we discuss the related work. In Section 8 we present our conclusions and discuss future work.

2 Preliminaries

We first introduce the basic definitions of judgment aggregation.

Judgments. Let $\mathcal{L}$ be a set of propositional variables. An *agenda* $\mathcal{A} = \{\varphi_1, \ldots, \varphi_m\}$ is a finite set $\mathcal{A} \subseteq \mathcal{L}$. The elements of $\mathcal{A}$ are called *issues*. A *judgment* is a (truth assignment) function $J : \mathcal{A} \to \{0,1\}$ mapping each issue to either 0 (reject) or 1 (accept). We write $\{0,1\}^{\mathcal{A}}$ as a shorthand for $\mathcal{A} \to \{0,1\}$, the space of all possible judgments for m issues, i.e., all sequences of length m comprised of 0s and 1s. We use $J(\varphi)$ to denote the value assigned to $\varphi \in \mathcal{A}$. The Hamming distance between two judgments is defined as the number of issues on which the judgments differ, i.e., $d_h(J^a, J^b) = \#\{\varphi \in \mathcal{A} \mid J^a(\varphi) \neq J^b(\varphi)\}$.

With each agenda we associate a *constraint* $\Gamma \in \mathcal{L}_{\mathcal{A}}$, where $\mathcal{L}_{\mathcal{A}}$ is the set of well formed formulas constructed with variables from $\mathcal{A}$ and the logical connectives $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), and $\to$ (implication). The formula Γ is assumed not to be a contradiction. A judgment from $\{0,1\}^{\mathcal{A}}$ is *rational* for Γ if and only if it is a model for Γ in the sense of classical propositional logic. For a given $\Gamma \in \mathcal{L}_{\mathcal{A}}$, we define $\mathcal{J}_{\mathcal{A},\Gamma} \subseteq \{0,1\}^{\mathcal{A}}$ to be the set of all *rational judgments* for $\mathcal{A}$ and Γ.

Agents and profiles. Let $N = \{1, 2, \ldots, n\}$ be a finite set of *agents*. A profile $P = (J_1, \ldots, J_i, \ldots, J_n) \in \mathcal{J}^n_{\mathcal{A},\Gamma}$ is a list of rational judgments, one for each agent. We denote $P[i] = J_i$ and $P_{-i} = (J_1, \ldots, J_{i-1}, J_{i+1}, \ldots, J_n)$. Further let $\{P\}$ be the set of all distinct judgments that are in P. We often abuse notation and write $J_i \in P$ when $P[i] = J_i$. We reserve subscripted judgments, e.g., J_i, to denote judgments that belong to some profile and the superscripted judgments, e.g., J^a, J^b, to denote rational judgments that may not belong to some profile. A profile is *unanimous* if $\{P\} = \{J\}$, for some $J \in \mathcal{J}_{\mathcal{A},\Gamma}$. A judgment J is a *plurality judgment* in P if and only if $\#\{i \mid P[i] = J\} \geq \#\{i \mid P[i] \neq J\}$.

Example 1 *The quintessential example in judgment aggregation is the "doctrinal paradox" which is described with* $\mathcal{A} = \{\varphi_1, \varphi_2, \varphi_3\}$ *and* $\Gamma = (\varphi_1 \wedge \varphi_2) \leftrightarrow \varphi_3$. *The* $\mathcal{J}_{\mathcal{A},\Gamma} = \{(0,0,0), (0,1,0), (1,0,0), (1,1,1)\}$. *The doctrinal paradox profile is* $P = ((0,1,0), (1,0,0), (1,1,1))$*, see also Figure 1. Note that all the three profile judgments are rational, but the collective judgment obtained by taking the value for each issue assigned by a strict majority of agents, the so called majority rule, is not rational.*

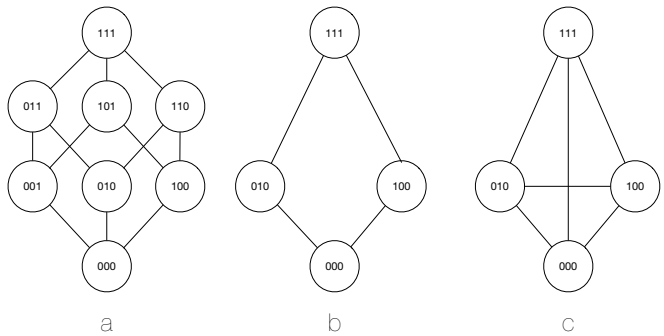


Figure 2. Agenda graphs for the doctrinal paradox: (a) the Hamming graph $G^h_{\mathcal{A}}$, (b) the Model graph $G^m_{\mathcal{A},\Gamma}$, (c) the Complete graph $G^c_{\mathcal{A},\Gamma}$ for $\mathcal{A}$ and Γ in Example 1.

3 Agenda Graphs

An *agenda graph* is a graph $G_{\mathcal{A},\Gamma} = \langle V, E\rangle$ whose nodes are judgments for some agenda $\mathcal{A}$, namely $V \subseteq \{0,1\}^{\mathcal{A}}$. Given an agenda $\mathcal{A}$ and constraints Γ we define three agenda graphs. The *Hamming graph* $G^h_{\mathcal{A}}$ is the graph over all possible (not necessarily rational!) judgments, that connects vertices which differ on exactly one issue. The *Model graph* $G^m_{\mathcal{A},\Gamma}$ is the graph over all *rational* judgments, where two judgments are adjacent iff no "compromise" exists between them. The *Complete graph* $G^c_{\mathcal{A},\Gamma}$ is the fully connected graph over all rational judgments. Formally:

- The **Hamming graph** is $G^h_{\mathcal{A}} = \langle \{0,1\}^{\mathcal{A}}, E^h\rangle$ where $(J^a, J^b) \in E^h$ iff $d_h(J^a, J^b) = 1$.
- The **Model graph** is $G^m_{\mathcal{A},\Gamma} = \langle \mathcal{J}_{\mathcal{A},\Gamma}, E^m\rangle$ where $(J^a, J^b) \in E^m$ iff there exists no judgment $J^c \in \mathcal{J}_{\mathcal{A},\Gamma}$ *between* J^a and J^b. A judgment $J^c \in \mathcal{J}_{\mathcal{A},\Gamma}$ is between judgments $J^a \in \mathcal{J}_{\mathcal{A},\Gamma}$ and $J^b \in \mathcal{J}_{\mathcal{A},\Gamma}$ when $J^c \neq J^a$, $J^c \neq J^b$, $J^a \neq J^b$ and for every $\varphi \in \mathcal{A}$ if $J^a(\varphi) = J^b(\varphi)$, then $J^c(\varphi) = J^a(\varphi) = J^b(\varphi)$.
- The **Complete graph** is $G^c_{\mathcal{A},\Gamma} = \langle \mathcal{J}_{\mathcal{A},\Gamma}, E^c\rangle$ where $E^c = \mathcal{J}_{\mathcal{A},\Gamma} \times \mathcal{J}_{\mathcal{A},\Gamma}$.

The agenda graphs for the doctrinal paradox of Example 1 are shown in Figure 2.

We use d_x to denote the (shortest) path distance on an agenda graph $G^x_{\mathcal{A},\Gamma}$. The path distance on $G^c_{\mathcal{A},\Gamma}$ is also known as the *drastic distance*: the distance between two judgments is 0 iff they are the same on all issues and 1 iff they differ on at least one issue. The path distance on $G^h_{\mathcal{A}}$ is the Hamming distance, and the path distance on $G^m_{\mathcal{A},\Gamma}$ is the Model distance introduced and formally characterized in [11]. Recall that a path distance on a graph $G_{\mathcal{A},\Gamma}$, as on any graph, is a distance function in the true sense since for every $J^a, J^b, J^c \in V$ it satisfies: $d(J^a, J^b) = 0$ iff $J^a = J^b$, $d(J^a, J^b) = d(J^b, J^a)$, and $d(J^a, J^c) \leq d(J^a, J^b) + d(J^b, J^c)$ (triangle inequality).

A graph $G' = \langle V', E'\rangle$ is a subgraph of graph $G = \langle V, E\rangle$, denoted $G' \subseteq G$, if $V' \subseteq V$ and $E' \subseteq E$. The V'-induced subgraph of a graph G is the graph $G' \subseteq G$ with vertices V' and edges E' which satisfies that, for every pair of vertices in V', they are adjacent in G if and only if they are adjacent in G'.

We consider profile-induced subgraphs of $G_{\mathcal{A},\Gamma}$ and make use of their "geometry". Therefore we define some useful concepts for a

given agenda graph $G_{\mathcal{A},\Gamma} = \langle V, E\rangle$ and agents $N = \{1, 2, \ldots, n\}$, following the terminology from graph theory [36].

- The *interval* between a pair of vertices $J^a, J^b \in V$, denoted $I_{\mathcal{A},\Gamma}[J^a, J^b]$, is the set of all the judgments on all the shortest paths in $G_{\mathcal{A},\Gamma}$ from J^a to J^b.
- A subset $S \subseteq V$ is *convex* if it is closed under $I_{\mathcal{A},\Gamma}$, namely when it includes all shortest paths between two vertices in S.
- The *convex hull of* P on $G_{\mathcal{A},\Gamma}$, denoted $\mathrm{CH}(P)$, is the smallest convex subset of V from $G_{\mathcal{A},\Gamma}$ that contains $\{P\}$.
- The *eccentricity* of a judgment $J^a \in S \subseteq V$ is $e_S(J^a) = \max\{d(J^a, J^b) \mid J^b \in S\}$, i.e., the farthest that J^a gets from any other judgment in S.
- A *diameter* of a set $S \subseteq V$ is $mx_d(S) = \max\{e_S(J) \mid J \in S\}$, namely the maximal eccentricity of a vertex in S. All judgments for which $mx_d(S) = e_S(J)$ are called *peripheral judgments* for S. For $S = \{P\}$ we call these judgments *peripheral judgments of a profile* P. If for two judgments J^a, J^b it holds that $d(J^a, J^b) = mx_d(P)$, then these are called *antipodal judgments of a profile* P.

Example 2 *Consider the Hamming graph $G^h_{\mathcal{A}}$ for the doctrinal paradox, presented in Figure 2a, and take the profile P as defined in Example 1. The $\mathrm{CH}(P)$-induced subgraph of $G^h_{\mathcal{A}}$ is given in Figure 3; this graph contains all the shortest paths, and nodes from $\mathcal{J}_{\mathcal{A},\Gamma}$, between $(0,1,0)$ and $(1,0,0)$, between $(0,1,0)$ and $(1,1,1)$, and between $(1,0,0)$ and $(1,1,1)$. Node $(0,0,1)$ is not in this $\mathrm{CH}(P)$-induced subgraph of $G^h_{\mathcal{A}}$ because this node is not on any of the shortest paths between the profile judgments.*

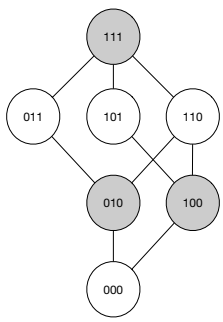


Figure 3. The $\mathrm{CH}(P)$-induced subgraph of $G^h_{\mathcal{A}}$ for P in Example 1.

We have defined the convex hull of P because we will build our iteration algorithm on the principle of only allowing the agents to move from their current judgment to an adjacent judgment in the hull of P. By doing this we ensure that the agents do not disperse away from each other. The Proposition 1 shows that if agents in profile P each move to a judgment in $\mathrm{CH}(P)$, thus creating profile P', the diameter of the P' profile cannot be bigger than that of the P.

Proposition 1 *For an agenda graph $G_{\mathcal{A},\Gamma}$ and profile $P \in \mathcal{J}^n_{\mathcal{A},\Gamma}$, if $S \subseteq \mathrm{CH}(P)$, then $\mathrm{CH}(S) \subseteq \mathrm{CH}(P)$ and $mx_d(\mathrm{CH}(S)) \leq mx_d(\mathrm{CH}(P))$.*

Proof This property follows from the fact that CH is a finitary closure operator [36, p. 6, Theorem 1.3]. Thus, for the convex hull $\mathrm{CH}(P)$, it holds that if $S \subseteq \mathrm{CH}(P)$, then $\mathrm{CH}(S) \subseteq \mathrm{CH}(\mathrm{CH}(P))$, and $\mathrm{CH}(\mathrm{CH}(P)) = \mathrm{CH}(P)$. ■

Definition 1 *We say that the profile P has a k-cycle in $G_{\mathcal{A},\Gamma}$ if and only the $\mathrm{CH}(P)$-induced subgraph of $G_{\mathcal{A},\Gamma}$ has a simple cycle of length k. We say that P is a k-cycle in $G_{\mathcal{A},\Gamma}$ if and only if the $\mathrm{CH}(P)$-induced subgraph of $G_{\mathcal{A},\Gamma}$ is a simple cycle in $G_{\mathcal{A},\Gamma}$ of length k.*

The doctrinal paradox profile from Example 1 is a 4-cycle in $G^m_{\mathcal{A},\Gamma}$, and it has a 6-cycle in $G^h_{\mathcal{A}}$, as can be inferred from Figure 2.

We make the following observation.

Observation 2 *No profile $P \in \mathcal{J}^n_{\mathcal{A},\Gamma}$ has a 3-cycle in $G^h_{\mathcal{A}}$ or in $G^m_{\mathcal{A},\Gamma}$, regardless of $\mathcal{A}$ and Γ.*

This exclusion of 3-cycles is due to the *betweenness* property of path distances: if J^b is between J^a and J^c in the graph, then $d(J^a, J^c) = d(J^a, J^b) + d(J^b, J^c)$.

Definition 2 *An interval $I_{\mathcal{A},\Gamma}[J^a, J^b]$ is* ϵ-connected *in $G_{\mathcal{A},\Gamma}$ if and only if the maximal path distance between two adjacent* **rational** *judgments in the $I_{\mathcal{A},\Gamma}[J^a, J^b]$-induced subgraph of $G_{\mathcal{A},\Gamma}$ is at most $\epsilon \in \mathbb{N}$. A profile P is* ϵ-connected *in $G_{\mathcal{A},\Gamma}$ if and only if every interval $I_{\mathcal{A},\Gamma}[P[i], P[j]]$ for $i, j \in N$ is ϵ-connected in $G_{\mathcal{A},\Gamma}$.*

While every interval in $G^c_{\mathcal{A},\Gamma}$ and $G^m_{\mathcal{A},\Gamma}$ is ϵ-connected for every $\epsilon \geq 1$, this may not be the case for intervals in $G^h_{\mathcal{A}}$. As an example, consider the doctrinal paradox profile P from Example 1 and graph $G^h_{\mathcal{A}}$ in Figure 2a. Here, the interval $I_{\mathcal{A},\Gamma}[(1,1,1),(0,1,0)]$ is 2-connected because every shortest path between $(1,1,1)$ and $(0,1,0)$ passes only through judgments that are not rational.

4 Iteration Algorithm

Collective opinions in human societies are often formed not in one step, but rather in an intricate process that involves mutual information exchange, argumentation, persuasion, and opinion revision. Typically, social agents are motivated by two somewhat conflicting needs: on one hand, they want to form a unified stance with a significant enough part of the community; on the other hand, they do not want to concede too much of their own opinion. Here, we try to mimic this kind of behaviour – obviously, in a very simplified way. To this end, we design an iteration algorithm, Algorithm 1, based on an agenda graph $G_{\mathcal{A},\Gamma}$. As it is standard in judgment aggregation, we assume that the agents can only chose rational judgments at each iteration step. For an agenda $\mathcal{A}$ and constraints Γ, each agent's judgment is a node in the graph $G_{\mathcal{A},\Gamma}$. In the first step of the iterative procedure each agent announces which node she has chosen. Two or more agents may choose the same node. The agents do not have a constructed $G_{\mathcal{A},\Gamma}$ available. At each subsequent step, the agents (simultaneously) compute their adjacent nodes and try to "move" to one of these adjacent nodes along some shortest path towards the other agents. A move is possible if and only if the adjacent judgment is rational and it brings the agent closer to the rest of the profile, i.e., it decreases its aggregated path distance to the other judgments. More precisely, an agent $i \in N$ will move from $P[i]$ to a J iff there exist a rational $J \in \mathrm{CH}(P)$ s.t. $d(J, P[i]) = 1$ and $\sum_{j\in N, j\neq i} d(P[i], P[j]) < \sum_{j\in N, j\neq i} d(J, P[j])$, where d is a path distance for a given $G_{\mathcal{A},\Gamma}$. Given a choice between two moves the agent chooses the one which better reduces the distance to the rest of the profile. If more than one move reduces the distance to the same extent, the agent chooses using some internal decision-making procedure which we do not model. We take it that in this case the agent chooses non-deterministically, with all move options being probable. The agents continue moving along the nodes of $G_{\mathcal{A},\Gamma}$ until no agent in the profile can move, or all of the agents "sit" on the same node, namely until a *consensus* is reached.

Data: $\epsilon > 0$, $\mathcal{J}_{\mathcal{A},\Gamma}$, N, own identifier $i \in N$, initial judgment J_i^0
Result: $P \in \mathcal{J}_{\mathcal{A},\Gamma}^{|N|}$
$t := 0$; MOVES $:= \emptyset$; $J_i := J_i^0$; $P :=$ *empty list*;
repeat
  $P' := P$;
  $P[i] := J_i$;
  for $j \in N, j \neq i$ **do**
    $send(J_i, j), receive(J_j, i), P[j] := J_j$;
  end
  MOVES $:=$ $\underset{J \in \mathrm{CH}(P) \cap \mathcal{J}_{\mathcal{A},\Gamma}}{\operatorname{argmin}} D(i, J, P) \cap$
    $\{J \mid 0 < d(J, J_i) \leq 1$ and $D(i, J, P) < D(i, P[i], P)\}$;
  if MOVES $\neq \emptyset$ **then**
    select $J \in$ MOVES, $J_i := J$;
  end
  $t := t + 1$;
until P *is unanimous or* $P' = P$;
return P;

Algorithm 1: Iteration algorithm

In Algorithm 1, $send(J_i, j)$ informs agent $j \in N$ that agent $i \in N$ has chosen to move to node $J_i \in V$, while $receive(J_j, i)$ denotes that the agent $i \in N$ has been informed that agent $j \in N$ has chosen to move to node $J_j \in V$. To ease readability we use $D(i, J, P) = \sum_{j \in N, j \neq i} d(J, P[j])$. We call $D(i, J, P)$ the distance of J to the profile P_{-i}. In Algorithm 1, at each iteration t, MOVES is the set of judgments that are strictly closer to P than the current judgment J_i. Note that the algorithm is fully decentralised, in the sense that there is no need for any central authority to take over the iteration at any point of the process.

The starting profile P^0 collects the initial individual opinions of the agents. That is, it is the profile that would be aggregated under classical one-step social choice. We say that the algorithm *reaches consensus* J for P^0, if it terminates starting from P^0 by returning the unanimous profile $\{P\} = \{J\}$. We first observe a necessary condition for reaching consensus.

Proposition 3 *If Algorithm 1 reaches consensus, then* P^0 *is* ϵ*-connected for* $\epsilon = 1$.

Proof Assume that the algorithm terminates with a J^*-unanimous P at some t. In each $t' < t$, every agent either keeps her own judgment $P[i]$, or moves to a new $J \in \mathcal{J}_{\mathcal{A},\Gamma}$ with $d(P[i], J) = 1$. Since a J^* is reached by every agent, there must exist a 1-connected path between any two judgments in P^0. Thus P^0 must be 1-connected. ■

Note also that if P^0 is ϵ-connected, so is P at any step $t > 0$. The interesting question is: what are the sufficient conditions for reaching consensus by Algorithm 1? We address the question in Section 5.

5 Reaching Consensus

In this section, we examine the sufficient conditions for reaching consensus by Algorithm 1. We begin by looking at the iteration over the complete agenda graph $G^c_{\mathcal{A},\Gamma}$, and then we move on to the more interesting cases of $G^h_{\mathcal{A}}$ and $G^m_{\mathcal{A},\Gamma}$.

5.1 Iteration with $G^c_{\mathcal{A},\Gamma}$

Theorem 4 *If* P *contains a* unique plurality judgment J*, then Algorithm 1 always reaches consensus in one step.*

Proof On $G^c_{\mathcal{A},\Gamma}$, the path distance d_c between any two judgments that are different is 1. Let J be the unique plurality judgment in $P \in \mathcal{J}^n_{\mathcal{A},\Gamma}$, selected by k agents. For every $J_i = J \neq J_j$, we have $D(i, J, P) = n - k \leq D(i, P[j], P)$, so the agents selecting J can not change their judgments. Moreover, switching from J_j to J decreases the distance to P_{-j} most, so all the other agents will switch to J in the first iteration. ■

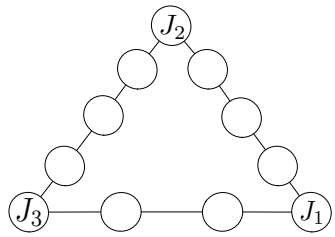


Figure 4. An agenda graph of a non-equidistant profile $P = (J_1, J_2, J_3)$

What about starting profiles with several plurality judgments? They converge towards consensus under reasonable conditions.

Theorem 5 *If* N *consists of an odd number of agents, then Algorithm 1 probabilistically reaches consensus, i.e., it reaches consensus with probability 1.*

Proof If there is a single plurality judgment in P, then the algorithm converges in one step. If there are two or more plurality judgments $J^1, \ldots, J^k \in \mathcal{J}_{\mathcal{A},\Gamma}$, then those agents swap nondeterministically between $J^1, \ldots, J^k$, and the other agents move to one of $J^1, \ldots, J^k$. In the next round, the same argument applies. Eventually, P converges either to the unanimous profile P' such that $\{P'\} = \{J\}$ for some $J \in \{J^1, \ldots, J^k\}$, or to a profile P'' such that $\{P''\} = \{J^1, \ldots, J^m\}$ for an odd m, each favoured by the same amount of agents. From then on, the agents keep swapping judgments until one judgment gets plurality in the profile, and wins in the next round.

Formally, let $\text{MOVES}_{i,t}$ be the set of moves available to agent i at the step t of Algorithm 1. We assume that there is some $\delta > 0$ such that, for each step t, agent i selects judgment $J \in \text{MOVES}_{i,t}$ with probability $p_i(J) \geq \delta$. Then, there exists $\delta' > 0$ such that the probability of all the agents "hitting" a profile with no plurality in the next round is at most $1 - \delta'$. Hence, the probability that the profile stays with no plurality in m steps is at most $(1 - \delta')^m$, which converges to 0 as m increases. ■

The Algorithm 1 has good convergence properties on $G^c_{\mathcal{A},\Gamma}$ but the consensus it reaches is limited to the judgments that are already in the starting profile. On the Hamming and Model agenda graphs Algorithm 1 surpasses this limitation. However, its convergence becomes a subtler issue.

5.2 Iteration with $G^h_{\mathcal{A}}$ and $G^m_{\mathcal{A},\Gamma}$

In this section, we will use $G_{\mathcal{A},\Gamma}$ to refer to one of $G^m_{\mathcal{A},\Gamma}, G^h_{\mathcal{A}}$ in order to avoid stating and proving the same properties for $G^m_{\mathcal{A},\Gamma}$ and $G^h_{\mathcal{A}}$ separately when the same proof can be applied.

We start with a negative result. Let us call *equidistant* those profiles P such that, for any $i, j, r \in N$ with $i \neq j$, $i \neq r$, $j \neq r$, it holds that $d(P[i], P[j]) = d(P[i], P[r])$.

Proposition 6 *Consider* $N = \{1, 2, 3\}$ *agents and a* 1*-connected* P^0*. If Algorithm 1 reaches a* P *that is an equidistant* k*-cycle, then* $Algorithm\ 1$ *will not terminate with a consensus.*

Proof If P is an equidistant k-cycle then no agent can reduce the distance to one agent by 1 without increasing the distance to the other agent by 1. Thus no agent has a possible move. ■

Note that the same applies to some non-equidistant k-cycles. For example, the profile in Figure 4 is not equidistant, but it is easy to check that each agent has an empty set of moves.

3-agent profiles that form a simple cycle are problematic because an agent may not be able to get closer to one of the other agents without distancing itself from the third. For profiles of more than three agents that form a simple cycle, the judgments cannot be equidistant, and this is no longer a problem.

Lemma 7 *If P is a (1-connected) k-cycle for $n > 3$ agents at step t of Algorithm 1, then the set* MOVES *at t is nonempty for some $i \in N$.*

Proof Take any judgment $P[i]$ which is peripheral in $\mathrm{CH}(P)$, and consider its antipodal judgment $P[j]$. Let $\mathbf{p_1}, \mathbf{p_2}$ be the two paths from $P[i]$ to $P[j]$ in $\mathrm{CH}(P)$, and let $J_m \in \mathrm{CH}(P)$ be the node adjacent to $P[i]$ on path $\mathbf{p_m}$, $m = 1, 2$. We have that $d(J_1, P[r]) = d(P[i], P[r]) - 1$ for every $P[r]$ on $\mathbf{p_1}$, while $d(J_2, P[s]) = d(P[i], P[s]) + 1$ for every $P[s]$ on $\mathbf{p_2}$. If there are more profile judgments on $\mathbf{p_1}$ than on $\mathbf{p_2}$, then $J_1 \in$ MOVES, otherwise $J_2 \in$ MOVES. If there are exactly as many judgments on $\mathbf{p_1}$ as there are on $\mathbf{p_2}$, then both $J_1 \in$ MOVES and $J_2 \in$ MOVES because in that case $d(J_1, P[j]) = d(J_2, P[j]) = d(P[i], P[j]) - 1$ and consequently $D(i, J, P) = D(i, J', P) < D(i, P[i], P)$. ■

Let us consider the case of a 1-connected P^0 for $|N| > 3$. Let P be the profile produced by Algorithm 1 at step $t \geq 0$, and let P' be the profile produced by Algorithm 1 at step $t + 1$. We begin by showing that for graphs in which no judgment has a degree higher than 2, it is never the case that $P = P'$, i.e., there exist at least one agent for which MOVES $= \emptyset$ for P.

Lemma 8 *Let P be a profile for $n > 3$ agents, produced by Algorithm 1 at step $t \geq 0$, and let P' be the profile produced by Algorithm 1 at step $t + 1$. Assume that P is 1-connected on $G_{\mathcal{A},\Gamma}$. If the $\mathrm{CH}(P)$-induced subgraph of $G_{\mathcal{A},\Gamma}$ is such that no vertex in it has a degree higher than two, and $mx_d(P) > 1$, then $P \neq P'$.*

Proof We show that at least the agent $i \in N$ with a peripheral judgment for P has a possible move in P.

Case a. There exists a peripheral judgment in P with degree 1, belonging to $i \in N$.
Let $P[j]$ be an antipodal of $P[j]$. Since $d(P[i], P[j]) > 1$ and P is 1-connected, there must exist exactly one judgment $J \in \mathcal{J}_{\mathcal{A},\Gamma}$, such that $d(P[i], J) = 1$ and that is between judgments $P[i]$ and $P[j]$. For every $r \in N$, $r \neq i$ it holds that $d(J, P[r]) = d(P[i], P[r]) - 1$. Thus J is a move for $P[i]$.

Case b. All peripheral judgments in P are with degree 2. Consider the antipodal judgments $P[i]$ and $P[j]$. There are exactly two shortest paths connecting them: $\mathbf{p_1}$ and $\mathbf{p_2}$. All other profile judgments $P[r]$ are: either on $\mathbf{p_1}$, or on $\mathbf{p_2}$, or have a shortest path to $P[j]$ that intersects either $\mathbf{p_1}$ or $\mathbf{p_2}$, possibly both. We can apply the same reasoning as in the proof of Lemma 7.
Consider $J \in \mathrm{CH}(P)$ adjacent to $P[i]$ on $\mathbf{p_1}$ and $J' \in \mathrm{CH}(P)$ adjacent to $P[i]$ on $\mathbf{p_2}$. We have that $d(J, P[r]) = d(P[i], P[r]) - 1$ for every $P[r]$ on $\mathbf{p_1}$ or whose shortest path to $P[j]$ intersects $\mathbf{p_1}$, while $d(J, P[s]) = d(P[i], P[s]) + 1$ for every $P[s]$ on $\mathbf{p_2}$ or whose shortest path to $P[j]$ intersects $\mathbf{p_2}$, but does not intersect $\mathbf{p_1}$. If there are more agents r than agents s, then $J \in$ MOVES, otherwise $J' \in$ MOVES. If there are exactly as many agents r as agents s, then both $J \in$ MOVES and $J' \in$ MOVES because in that case $d(J, P[j]) = d(J', P[j]) = d(P[i], P[j]) - 1$ and consequently $D(i, J, P) = D(i, J', P) < D(i, P[i], P)$.

■

Observe that if the $\mathrm{CH}(P^0)$-induced subgraph on $G_{\mathcal{A},\Gamma}$ is such that every vertex in it is of degree at most two, then for every subsequently constructed P in Algorithm 1, it will hold that the $\mathrm{CH}(P)$-induced subgraph on $G_{\mathcal{A},\Gamma}$ is such that every vertex in it is of degree at most two. This is due to the fact that, at each step of Algorithm 1, the agents can only chose judgments from the $\mathrm{CH}(P^0)$.

From Observation 2 we know that a $\{P\}$-induced subgraph of $G^h_{\mathcal{A}}$ and $G^m_{\mathcal{A},\Gamma}$ does not have 3-cycles. If the $\mathrm{CH}(P^0)$-induced subgraph of $G^h_{\mathcal{A}}$, respectively $G^m_{\mathcal{A},\Gamma}$ contains no k-cycles for $k > 3$, then this induced subgraph contains no cycles and it is by definition a tree. From the Case a. of the proof of Lemma 8, we immediately obtain the following corollary.

Corollary 9 *Let P be a profile produced in Algorithm 1 at step $t \geq 0$ and let P' be profile produced in Algorithm 1 at step $t+1$. Assume that P is 1-connected on $G_{\mathcal{A},\Gamma}$. If the $\mathrm{CH}(P)$-induced subgraph of $G_{\mathcal{A},\Gamma}$ is a tree, and $mx_d(P) > 1$, then $P \neq P'$.*

Proof The proof follows from the Case a. of the proof of Lemma 8 and the observation that: all the subgraphs of a tree are trees, and the peripheral vertices of a tree have a degree 1. ■

We now need to show that not only the profile changes in each iteration, it also changes towards a consensus.

From Proposition 1 we have that $mx_d(\mathrm{CH}(P))$ does not increase with each step of the Algorithm 1. It is possible that $mx_d(\mathrm{CH}(P)) = mx_d(\mathrm{CH}(P'))$ for P' being constructed immediately after P in Algorithm 1. From the proof of Lemma 8 we have the following corollary.

Corollary 10 *Let $P \in \mathcal{J}^n_{\mathcal{A},\Gamma}$ be a profile produced in Algorithm 1 at step $t \geq 0$ and let $P' \in \mathcal{J}^n_{\mathcal{A},\Gamma}$ be profile produced in Algorithm 1 at step $t + 1$. If $\{P\} = \{P'\}$, then the $\{P\}$-induced graph of $G_{\mathcal{A},\Gamma}$ has at least one k-cycle, where $2m + 2 \geq k \geq 2m$.*

Clearly if the agents whose judgments are antipodal in P can choose to move towards each other via two different shortest paths between their judgments causing $\{P\} = \{P'\}$. These agents however, also have the possibility to chose to move towards each other on the same shortest path between their judgments. As soon as two agents use the same shortest path, the k-cycle will be broken in the next step of the algorithm and $\{P\} \neq \{P'\}$.

Let us consider the case when $mx_d(P) = 1$.

Lemma 11 *Let P be a 1-connected profile for $n > 3$ agents at step t with $mx_d(P) = 1$ and let P' be a profile obtain from it by Algorithm 1 at step $t + 1$. If n is odd then $\{P\} \neq \{P'\}$.*

Proof In this case the Algorithm 1 behaves as on the $G^c_{\mathcal{A},\Gamma}$ graph, see Theorems 4 and 5, except the P-induced subgraf of $G_{\mathcal{A},\Gamma}$ will have no 3-cycles (or any size cycles since $mx_d(P) = 1$). Namely, if there is one plurality judgment J in P, all the agents can reach it, because $mx_d(P) = 1$ and P is 1-connected. Consequently $\{P'\} = 1$. If more than one plurality judgment exists, the agents whose judgment is this plurality judgment will not have a move, while and all the other agents will move to their choice of a plurality judgment. If n is odd P' will have exactly one plurality judgment and the profile P'' constructed by Algorithm 1 in step $t + 1$ is a consensus. If however n is even, as with $G^c_{\mathcal{A},\Gamma}$, P can be such that half of the agents have a judgment J, while the other half have an adjacent judgments J'. Namely $\{P\} = \{J, J'\}$ and $d(J, J') = 1$. If such P is reached the Algorithm 1 forces the agents to infinitely "swap" between J and J'. ■

Lemma 12 *Assume an odd number of agents $n > 3$. that in the P-induced subgraph on $G_{\mathcal{A},\Gamma}$ each vertex has a degree at most 2. Let $P \in \mathcal{J}^n_{\mathcal{A},\Gamma}$ be s.t. $\mathrm{CH}(P)$ has at least one k-cycle for $k > 3$. Let $p_i(J) > 0$ be the probability that an agent i will choose a possible move J from the set* MOVES *at a step t_1 in the Algorithm 1. Then the algorithm will reach a point $t_2 > t_1$ where $P' \in \mathcal{J}^n_{\mathcal{A},\Gamma}$ is obtained s.t. $\mathrm{CH}(P) \subset \mathrm{CH}(P')$ with probability 1.*

Proof If a profile P'' is reached such that all antipodal judgments have degree two, it is sufficient that only one antipodal pair "breaks" the cycle for a profile P' to be reached. To do so, two agents with antipodal judgments have to chose to move along the same path towards each other. Consider a pair of antipodal judgments in P, $P[i]$ and $P[j]$. Assume that at the non-deterministic step of the algorithm there exists a probability $1 > p_i(J) > 0$ that the agent i selects $J \in$ MOVES that is on a shortest path **p** between $P[i]$ and $P[j]$ and probability $p_i(J') = 1 - p_i(J)$ that she selects $J' \in$ MOVES that is on a different path **q** between $P[i]$ and $P[j]$. Similarly, let those probabilities be $1 > p_j(J'') > 0$ that agent j selects to move to J'' on path **p** and $p_j(J''') = 1 - p_j(J'')$ for the probability that j moves to J''' on some other path **q'** (**q** and **q'** may not be the same). Since the agents decide on their moves independently, the probability that agent i will chose the same path as j is $p_i(J) \cdot p_j(J'') > 0$. Since the two peripheral judgments $J[i]$ and $J[j]$ are no longer part of the new profile P', $\{P'\} \subset \mathrm{CH}(P)$ and from Corollary 10 we get that $\mathrm{CH}(P) \subset \mathrm{CH}(P')$ is reached after a finite time with probability 1. ■

Let us call **Class A for** $G_{\mathcal{A},\Gamma}$ the set of all $\mathrm{CH}(P)$-induced subgraphs of $G_{\mathcal{A},\Gamma}$ that are tree graphs. Let us call **Class B for** $G_{\mathcal{A},\Gamma}$ the set of all $\mathrm{CH}(P)$-induced subgraphs of $G_{\mathcal{A},\Gamma}$ whose vertices have a degree of at most 2. For instance, the doctrinal paradox profile from Example 1 is in Class B for $G^m_{\mathcal{A},\Gamma}$, see Graph b in Figure 2. On the other hand, it is neither in Class B nor in Class A for $G^h_{\mathcal{A}}$, see Example 2 and Figure 3.

We can now state the following theorem whose proof follows from Lemma 8, Corollary 10, Corollary 9, Lemma 11 and Lemma 12.

Theorem 13 *Let $P^0 \in \mathcal{J}^n_{\mathcal{A},\Gamma}$ be a 1-connected profile belonging to Class A or to Class B for $G^h_{\mathcal{A}}$ or $G^m_{\mathcal{A},\Gamma}$. If $n > 3$ is odd, and each element of* MOVES *has a non-null probability of being selected in the non-deterministic choice step, then the Algorithm 1 reaches consensus with probability 1 on $G^h_{\mathcal{A}}$, respectively $G^m_{\mathcal{A},\Gamma}$.*

6 Properties of Consensus

In this section, we compare the output and performance of our iteration procedure to those of standard distance-based judgment aggregation rules. We first discuss the "quality" of the consensual decision. Then, we look at the computational complexity of the procedure.

6.1 Consensus Quality

Distance-based judgment aggregation [28, 39, 21, 19] combines an algebraic aggregation function $\star$ with a distance function d (not necessarily a path distance in some agenda graph) in order to select the collective opinion that is closest to the given profile. Given $P \in \mathcal{J}^n_{\mathcal{A},\Gamma}$, the distance-based aggregation function $F^{d,\star} : \mathcal{J}^n_{\mathcal{A},\Gamma} \to 2^{\mathcal{J}_{\mathcal{A},\Gamma}} \setminus \emptyset$ is defined as

$$F^{d,\star}(P) = \operatorname*{argmin}_{J \in \mathcal{J}_{\mathcal{A},\Gamma}} \star (d(P[1], J), \ldots, d(P[n], J)).$$

Natural questions to ask are:

- How does Algorithm 1 perform in comparison to $F^{d,\Sigma}$ when d is a path distance in an agenda graph?
- How do the collective judgments $F^{d,\Sigma}(P^0)$ compare to the consensus judgment reached by Algorithm 1 for a given starting profile P^0?

The questions cannot be fully explored within the scope of this paper. However, we establish some initial properties below.

A property generally deemed desirable in judgment aggregation is that of *propositional unanimity* [21, 39, 16]. Propositional unanimity requires that, if every agent in profile P has the same value for some issue $\varphi \in \mathcal{A}$, then the same value for $\varphi \in \mathcal{A}$ shows up either in at least one of the judgments in $F^{d,\star}$ (weak unanimity) or in all of the judgments in $F^{d,\star}$ (strong unanimity). It is interesting to note that the most popular distance based judgment aggregation rule $F^{d_h,\Sigma}$ does not satisfy even the weak version of the property [38] and the same applies to $F^{d_m,\Sigma}$ and $F^{d_c,\Sigma}$ [22]. In this respect, iterative consensus building behaves better.

Proposition 14 *If Algorithm 1 terminates with a consensus on $G^h_{\mathcal{A}}$, then the consensus satisfies strong unanimity with respect to the initial profile P^0.*

Proof Note that, for $G^h_{\mathcal{A}}$, judgment J' is between judgments J and J'' iff vertex J' is on the shortest path between vertices J and J'' in the graph. Consequently, if all the agents in P^0 give the same truth-value on an issue, then $\mathrm{CH}(P^0)$ cannot contain judgments that assign different truth-value to this issue. ■

The same is not the case for $G^m_{\mathcal{A},\Gamma}$.

Proposition 15 *There is an initial profile P^0 such that Algorithm 1 terminates with a consensus on $G^m_{\mathcal{A},\Gamma}$, and the consensus does not satisfy weak unanimity with respect to P^0.*

Proof As a counter-example consider Example 1 and Graph *b* in Figure 2. The vertex $(1,1,1)$ is between vertices $(0,1,0)$ and $(1,0,0)$, but the judgment $(1,1,1)$ is not between judgments $(0,1,0)$ and $(1,0,0)$. Thus the agents can move from $(0,1,0)$ and $(1,0,0)$ to $(1,1,1)$ thus violating propositional unanimity on the last issue. ■

A big advantage of one-shot distance-based aggregation $F^{d,\Sigma}$ is that it produces output (a winner or winners) on any profile P^0, while our Algorithm 1 is more restricted in this respect. As we have seen, a necessary condition for successful termination of Algorithm 1 is that P^0 is 1-connected. Sufficient conditions are even more restrictive. Still, Proposition 14 demonstrates that, when Algorithm 1 reaches a consensus, it is structurally "better behaved" then a distance-based judgment aggregation rule for the most popular approach based on the sum of Hamming distances. In the next subsection we show that Algorithm 1 is also "better behaved" in the sense of computational complexity.

6.2 Complexity of Reaching Consensus

An important drawback of distance-based judgment aggregation is the computational complexity of producing the output, i.e., the winning judgment or judgments. The winner determination problem for $F^{d_h,\Sigma}$ is known to be Θ^p_2-complete [14], and the result extends to most other distances d and aggregation functions $\star$ [19]. The computational complexity of determining the collective judgment sets by $F^{d_m,\Sigma}$ is actually not known. How does it work for the iteration procedure formalized with Algorithm 1?

We have shown that the algorithm reaches consensus for an odd number $n > 3$ of agents on 1-connected, not equidistant profiles. How costly is it to reach the consensus? On $G^c_{\mathcal{A},\Gamma}$, it is evident that Algorithm 1 performs well, but the resulting consensus is not very exciting. For the $G^m_{\mathcal{A},\Gamma}$ graph, the consensus-friendly attitude may not earn much in terms of computational complexity, when compared to $F^{d_m,\Sigma}$. For each $P[i]$, we need to find every $J \in \mathcal{J}_{\mathcal{A},\Gamma}$ s.t. there is no rational judgment between $P[i]$ and J. It is not difficult to show, by a reduction to coSAT, that checking whether there is no rational judgment between two given rational judgments is in general coNP-complete. This has to be repeated for multiple candidate judgments to compute the set MOVES, and on top of that with every iteration of the algorithm. As a consequence, we get the following.

Theorem 16 *For $G^m_{\mathcal{A},\Gamma}$, determining* MOVES *of a single iteration of Algorithm 1 coNP-hard.*

Note that the hardness result is not really due to the iteration procedure, but rather due to the inherent complexity of computing d_m, which requires to determine nonexistence of particular rational judgments, i.e., to solve the Boolean co-satisfiability problem.

In contrast, the Hamming distance d_h can be always computed efficiently. Consequently, when Algorithm 1 reaches a consensus on $G^h_{\mathcal{A}}$, it is also "better behaved" computationally than the distance-based judgment aggregation rule $F^{d_h,\Sigma}$. We demonstrate it formally.

Proposition 17 *For $G^h_{\mathcal{A}}$, a single iteration of Algorithm 1 runs in deterministic polynomial time.*

Proof Follows from the fact that the set MOVES can be constructed by checking at most $|\mathcal{A}|$ candidate judgments. ■

By Corollary 10, if the $\mathrm{CH}(P^0)$ induced subgraph of $G^h_{\mathcal{A}}$ has no cycles, then the diameter of $\mathrm{CH}(P)$ is strictly shrinking with each non-terminating step t. In consequence, if Algorithm 1 reaches consensus for such P^0, then it does so in polynomially many steps. However, in case of cycles in the $\mathrm{CH}(P^0)$ induced graph in $G^h_{\mathcal{A}}$, the algorithm may run into such a cycle and take some time until the agents "stray" from the loop. When it happens, any judgment occurring on the loop can be the consensus. Using this observation, we propose the following modification of Algorithm 1.

Algorithm 2: Same as Algorithm 1, only it stops the iteration when $\{P_t\} = \{P_{t'}\}$ for some $t > t'$, and non-deterministically chooses one $J \in \{P\}$ as the consensus, producing in the next step P_{t+1} with $\{P_{t+1}\} = \{J\}$.

Unlike Algorithm 1, Algorithm 2 avoids looping and waiting until two or more agents "move" in the same direction. It also avoids infinite loops in the case of profiles with evenly many agents. On the other hand, Algorithm 2 is no longer decentralised, which is a clear disadvantage. We suggest that it can be treated as a technical variant of Algorithm 1 that potentially reduces its running time by employing a trusted third party which simulates probabilistic convergence of the profile in Algorithm 1 by one-step non-deterministic choice in Algorithm 2. The following formal results, which are straightforward consequences of our analysis above, justify the suggestion.

Theorem 18 *Consider $G^h_{\mathcal{A}}$ and N such that $|N|$ is odd and larger than 3. If Algorithm 1 can reach consensus with J then also Algorithm 2 can reach consensus with J.*

Theorem 19 *Consider $G^h_{\mathcal{A}}$ and N such that $|N|$ is odd and larger than 3. Moreover, let $P^0 \in \mathcal{J}^n_{\mathcal{A},\Gamma}$ be 1-connected and not equidistant. Algorithm 2 reaches consensus from P^0 on $G^h_{\mathcal{A}}$ in deterministic polynomial time.*

Lastly, let us observe that checking whether P^0 is equidistant can be done in linear time of the number of agents. For a graph G, determining if it has a simple cycle of size k, for k fixed, is a polynomial time problem over the size of G, see [1], however we do not generate the full $G_{\mathcal{A},\Gamma}$ when (or before) we run the iteration algorithm.

7 Related Work

List [25] considered judgment transformation functions $\tau : \mathcal{J}^n_{\mathcal{A},\Gamma} \to (\{0,1\}^{\mathcal{A}})^n$ as means to building iteration procedures for judgment aggregation problems. He showed that for a set of desirable properties no transformation functions exists. Such impossibility results exist for judgment aggregation functions, however, by relaxing some of the properties, specific judgment aggregation operators have been constructed: quota-based rules [6], distance-based rules [37, 28, 14, 11], generalisations of Condorcet-consistent voting rules [30, 29, 21], and rules based on the maximisation of some scoring function [21, 5, 42]. To the best of our knowledge, specific iteration procedures for judgment aggregation problems have not been proposed in the literature.

List [25] argues that the desirable conditions for judgment transformation functions should satisfy the following properties: universal domain, rational co-domain, consensus preservation, minimal relevance, and independence. Universal domain is satisfied when the transformation function accepts as admissible input any possible profile of rational judgments. Rational co-domain is satisfied when the function always outputs a profile of rational judgments. Consensus preservation is satisfied when τ always maps unanimous profiles into themselves. Minimal relevance is a weak property. It is satisfied when for each $P[i]$ there exists a profile P' to which P can be transformed such that $P[i] = P'[i]$. In other words, the transformation function should be such that it does not allow one agent to never change her judgment regardless of what the other profile judgments are. Lastly independence is satisfied when for each agenda issue, $J'_i(\varphi)$ depends only on $J(\varphi)$, and not on $J(\phi)$ for some other $\phi \in \mathcal{A}$; $J'_i = P'[i]$, $J_i = P[i]$, $P' = \tau(P)$.

Each step of Algorithm 1 can be seen as a (distributed) function that transforms an input profile into an output profile, namely as a List judgment transformation function. Given a profile $P \in \mathcal{J}_{\mathcal{A},\Gamma}$, let $T_d(P) = \{P' \mid P' \in \mathcal{J}^n_{\mathcal{A},\Gamma}, d(P[i], P'[i]) \leq 1, i \in [1,n]\}$. We can define the transformation function τ_d that maps a profile $P \in \mathcal{J}^n_{\mathcal{A},\Gamma}$ to a profile $P' \in T_d(P)$. Although Algorithm 1 does not terminate for each profile, τ_d does satisfy universal domain in the case of $G^c_{\mathcal{A},\Gamma}$ and $G^m_{\mathcal{A},\Gamma}$, because each step of the algorithm transforms the profile (possibly into itself). Universal domain is not satisfied on $G^h_{\mathcal{A}}$ because profiles on this graph do not always satisfy the necessary conditions for termination with a consensus. The rational co-domain and the consensus preservation properties are also trivially satisfied. It is not difficult to show that the minimal relevance property is also satisfied. Independence is the desirable property that is violated, and in fact List [25] argues that relaxing independence is the most plausible path towards avoiding the impossibility result.

In voting, deliberation and iterative consensus have been studied, although perhaps not axiomatically. As most similar with our work we distinguish [18] and [15]. Voting problems can be represented as judgment aggregation problems, see e.g., [6, 23], therefore it is possible to compare these works with ours. First we show how voting problems are represented in judgment aggregation.

A voting problem is specified with a set of agents N and a set of candidate options $O = \{x_1, x_2, \ldots, x_m\}$. Let $\mathcal{O}$ be the set of all total, transitive, and antisymmetric orders over the elements of O. A vote $\succ$ is an element of $\mathcal{O}$ and a voting profile is a collection of votes, one for each agent in N. The preference agenda $\mathcal{A}_o$ is constructed by representing each pair of options x_i and x_j, where $i < j$ with an issue $x_i P x_j$. The constraint Γ_{tr} is the transitivity constraint defined as $\Gamma_{\text{tr}} = \bigwedge_{x_iPx_j, x_jPx_k, x_iPx_k \in \mathcal{A}_o} \big((x_iPx_j) \wedge (x_jPx_k) \rightarrow (x_iPx_k)\big)$. For each vote $\varphi \in \mathcal{O}$ we obtain a rational judgment $J_\succ$ such that $J_\succ(x_iPx_j) = 1$ iff $x_i \succ x_j$ and $J_\succ(x_iPx_j) = 0$ iff $x_j \succ x_i$.

A Condorcet winner for a voting profile, when it exists, is the option that wins the majority of pairwise comparison for every other option in O, see e.g., [32]. The corresponding concept in the judgment aggregation representation of a voting problem is called *majority consistency*. A judgment profile is majority-consistent if the judgment obtained by taking the value for each issue assigned by a strict majority of agents in the profile is rational. The doctrinal paradox profile from Example 1 is not majority-consistent. It was shown [23, 31] that if a judgment profile on the preference agenda is majority-consistent, then the corresponding voting profile has a Condorcet winner.

Hassanzadeh *et al.* [18] consider an iterative consensus algorithm for voting profiles. In their algorithm, each agent is allowed to (simultaneously with other agents) move from vote $\succ_i$ to vote $\succ$ if she can flip the order of two adjacent options without violating transitivity. This corresponds to the agents moving to an adjacent judgment

in the agenda graph $G^h_{\mathcal{A}_o,\Gamma_{\text{tr}}}$. Hassanzadeh *et al* consider the majority graph for a voting profile (for an odd number of agents): the vertices in this directed graph are the elements of O and there is an edge from x_i to x_j if there are more agents in the profile who prefer x_i to x_j than agents who prefer x_j to x_i. The majority graph corresponds to a judgment $J \in \mathcal{J}_{\mathcal{A}_o,\Gamma_{\text{tr}}}$ for which $J(x_iPx_j) = v$, $v \in \{0,1\}$ if there is a strict majority of agents $r \in N$ for which $J_r(x_iPx_j) = v$, $J_r = P[r]$. Hassanzadeh *et al* show that their algorithm terminates with a consensus on the Condorcet winner when the majoritarian graph has no cycles. If the majority graph of a voting profile has no cycles, then the voting profile has a Condorcet winner.

Goel and Lee [15] consider an iteration procedure in which the agents "move" along adjacent vertices along (what corresponds to) the graph $G^m_{\mathcal{A},\Gamma}$. They do not commit to the nature of their vertices, so they are not exactly judgments or alternatives, just allowed options for iteration. In their algorithm not all agents move individually, but three agents at a time first reach a consensus and then all three move to the consensus option in the graph. Goel and Lee consider line graphs, graphs in which two vertices have degree 1 and all other vertices have degree 2, and show that the consensus produced by their algorithm is the generalised median. Namely, if the options in their algorithms were judgments from $\mathcal{J}_{\mathcal{A},\Gamma}$ the consensus their algorithm reaches for these graphs is an approximation of $F^{d_m,\Sigma}$.

Both [18] and [15] offer interesting directions for future study in context of our algorithm: to consider the profiles that have a Condorcet winner (see e.g., [23] for the concept of Condorcet winner in judgment aggregation) and to consider triadic iteration, allowing three agents to coordinate their moves with respect to each other and then see when a consensus emerges.

It is an open question of how our algorithm would perform on the special case of voting problems represented in judgment aggregation. The $G^h_{\mathcal{A}_o,\Gamma_{\text{tr}}}$ graph on the preference agenda has a more regular topology in comparison to general judgment aggregation problems, it is a permutahedron. For example, for an agenda of three options, the graph $G^h_{\mathcal{A}_o,\Gamma_{\text{tr}}}$ is a cycle of length 6. For every $J, J' \in \mathcal{J}_{\mathcal{A}_o,\Gamma_{\text{tr}}}$, $d_h(J,J') = d_m(J,J')$, thus the necessary conditions for reaching consensus for Algorithm 1 would be satisfied even on $G^h_{\mathcal{A}}$ because every profile on the preference agenda and judgments rational for the transitivity constraint is 1-connected in $G^h_{\mathcal{A}_o,\Gamma_{\text{tr}}}$. The graph $G^m_{\mathcal{A}_o,\Gamma_{\text{tr}}}$ always has $\frac{|O|\cdot(|O|-1)}{2}$ vertices and each of these vertices has a degree $|O|-1$. We leave for future work the study of whether our algorithm terminates for voting profiles. In particular, we conjecture that the Algorithm 1 for an odd number of agents will converge on the Condorcet winner in the case of voting profiles from the single crossing domain [4], also studied in judgment aggregation [8]. This is because profiles in this domain would have a hull whose induced graph is a line on $G^h_{\mathcal{A}_o,\Gamma_{\text{tr}}}$.

Lastly, we must mention [33]. Obraztsova *et al* [33] consider a graph similar to our Hamming agenda graph. They work with preferences, not judgments, but most importantly, the vertices of their graph are elements of (what would correspond to) $\mathcal{J}^n_{\mathcal{A},\Gamma}$, i.e., the vertices are profiles of votes. There exists a connection between two profiles if one profile can be obtained from the other by making exactly one swap between adjacent options in one vote. Obraztsova *et al* [33] study the properties of voting rules with respect to the "geometry" of the profiles in their graph.

8 Conclusions

In this paper, we propose a decentralised algorithm for iterative consensus building in judgment aggregation problems. We study the termination conditions for this algorithm, some of its structural properties, and its computational complexity.

In order to reach a consensus, our algorithm exploits the topology of a graph. All the available judgments that the agents can chose from are vertices in the graph. The algorithm models an agent's change of mind as a move between adjacent judgments in the graph. We define three natural graphs that can be constructed for a set of rational judgments $\mathcal{J}_{\mathcal{A},\Gamma}$: the complete graph $G^c_{\mathcal{A},\Gamma}$, the Hamming graph $G^h_{\mathcal{A}}$, and the model agenda graph $G^m_{\mathcal{A},\Gamma}$. We prove that our algorithm always terminates for an odd number of agents on the graph $G^c_{\mathcal{A},\Gamma}$, but it necessarily selects one of the judgments proposed in the first round of iterations. For the graphs $G^h_{\mathcal{A}}$ and $G^m_{\mathcal{A},\Gamma}$ we show a class of profiles for which the algorithm terminates with a consensus and a class of profiles for which it does not terminate with a consensus.

If the agents initially chose judgments such that the convex hull of the profile of these judgments induces a subgraph of $G^h_{\mathcal{A}}$, or $G^m_{\mathcal{A},\Gamma}$ in which each vertex has a degree of at most 2, then our algorithm probabilistically terminates with a consensus for an odd number of (more than 3) agents.

The list of profiles we give here, for which Algorithm 1 terminates with a consensus, is clearly not exhaustive. For example, it is easy to show that, for an odd number of agents, Algorithm 1 terminates with a consensus if the $\text{CH}(P^0)$ induced subgraph of $G^h_{\mathcal{A}}$, or $G^m_{\mathcal{A},\Gamma}$, is such that it contains only k-cycles, where $k = 2 \cdot mx_d(P^0) + 1$. This is because for such profiles there exists at least one pair of antipodal judgments with degree no more than 2 who will have a nonempty set MOVES. An immediate direction for future work is to strengthen our results with other classes of consensus terminating agenda graph topologies, particularly those corresponding to profiles on the preference agenda (and transitivity constraints).

A step of our algorithm implements a judgment profile transformation function of the type defined in [25]. List [25] gives an impossibility characterisation of such functions. Our function "escapes" this impossibility result by not satisfying the independence property on all agenda graphs and the universal domain on $G^h_{\mathcal{A}}$.

While $G^c_{\mathcal{A},\Gamma}$, and $G^m_{\mathcal{A},\Gamma}$ satisfy the necessary conditions for termination of Algorithm 1 for any $\mathcal{A}$ and Γ, this is not the case with $G^h_{\mathcal{A}}$, which is why the transformation function fails to satisfy universal domain on $G^h_{\mathcal{A}}$. On $G^h_{\mathcal{A}}$, sometimes all the adjacent judgments to a rational judgment J are not rational and thus not allowed to move to. In our future work we aim to explore modifications of the algorithm allowing the agents to make "longer" moves, i.e., to "jump over" a vertex that is not a rational judgment.

In Section 6 we gave two results with respect to the quality of the consensus reached by Algorithm 1 with respect to the widely used distance-based aggregation function $F^{d_h,\Sigma}$. This function $F^{d_h,\Sigma}$ is also known as the median aggregation rule and it is widely used in many domains, e.g., generalises the Kemeny voting rule, see [23], and for measuring dissimilarity between concepts in ontologies [10]. We merely scratched the surface of this consensus quality analysis and this line of research merits further attention.

Lastly, a more long-term goal for our future work is to explore versions of iteration on an agenda graph where the agents do not try to move to reduce the path distance to all of the other agents, but only to their neighbours in a given social network, or as in [15], to two randomly selected two agents.

Acknowledgements. We are grateful to Edith Elkind and the anonymous reviewers for their help in improving this work. Wojciech Jamroga acknowledges the support of the 7th Framework Programme of the EU under the Marie Curie IEF project ReVINK (PIEF-GA-2012-626398). We also acknowledge the support of ICT COST Action IC1205.

REFERENCES

[1] N. Alon, R. Yuster, and U. Zwick, 'Finding and counting given length cycles', *Algorithmica*, **17**(3), 209–223, (1997).

[2] G. Boella, G. Pigozzi, M. Slavkovik, and L. van der Torre, 'Group intention is social choice with commitment', in *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, eds., Marina De Vos, Nicoletta Fornara, Jeremy Pitt, and George Vouros, volume 6541 of

Lecture Notes in Computer Science, 152–171, Springer Berlin / Heidelberg, (2011).
[3] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A.D. Procaccia, *Handbook of Computational Social Choice*, Cambridge University Press, March 2016.
[4] R. Bredereck, J. Chen, and G. J. Woeginger, 'A characterization of the single-crossing domain', *Social Choice and Welfare*, **41**(4), 989–998, (2012).
[5] F. Dietrich, 'Scoring rules for judgment aggregation', *Social Choice and Welfare*, 1–39, (2013).
[6] F. Dietrich and C. List, 'Arrow's theorem in judgment aggregation', *Social Choice and Welfare*, **29**(1), 19–33, (July 2007).
[7] F. Dietrich and C. List, 'Judgment aggregation by quota rules: Majority voting generalized', *Journal of Theoretical Politics*, **19**(4), 391–424, (2007).
[8] F. Dietrich and C. List, 'Majority voting on restricted domains', *Journal of Economic Theory*, **145**(2), 512–543, (2010).
[9] F. Dietrich and P. Mongin, 'The premiss-based approach to judgment aggregation', *Journal of Economic Theory*, **145**(2), 562–582, (2010).
[10] F. Distel, J. Atif, and I. Bloch, 'Concept dissimilarity with triangle inequality', in *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*, pp. 614–617, (2014).
[11] C. Duddy and A. Piggins, 'A measure of distance between judgment sets', *Social Choice and Welfare*, **39**, 855–867, (2012).
[12] U. Endriss and R. de Haan, 'Complexity of the winner determination problem in judgment aggregation: Kemeny, Slater, Tideman, Young', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pp. 117–125, Richland, SC, (2015). International Foundation for Autonomous Agents and Multiagent Systems.
[13] U. Endriss and U. Grandi, 'Binary aggregation by selection of the most representative voters', in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 668–674, (2014).
[14] U. Endriss, U. Grandi, and D. Porello, 'Complexity of judgment aggregation', *Journal Artificial Intelligence Research (JAIR)*, **45**, 481–514, (2012).
[15] A. Goel and D. Lee, 'Triadic consensus', in *Internet and Network Economics: 8th International Workshop, WINE 2012, Liverpool, UK, December 10-12, 2012. Proceedings*, ed., P. W. Goldberg, volume 7695 of *Lecture Notes in Computer Science*, 434–447, Springer Berlin Heidelberg, Berlin, Heidelberg, (2012).
[16] U. Grandi and U. Endriss, 'Lifting integrity constraints in binary aggregation', *Artificial Intelligence*, **199–200**, 45–66, (2013).
[17] U. Grandi, A. Loreggia, F. Rossi, K. Brent Venable, and T. Walsh, 'Restricted manipulation in iterative voting: Condorcet efficiency and borda score', in *Proceeding of the 3rd International Conference on Algorithmic Decision Theory (ADT-2013)*, pp. 181–192, (2013).
[18] F.F. Hassanzadeh, E. Yaakobi, B. Touri, O. Milenkovic, and J. Bruck, 'Building consensus via iterative voting', in *2013 IEEE International Symposium on Information Theory Proceedings (ISIT)*, pp. 1082–1086, (July 2013).
[19] W. Jamroga and M. Slavkovik, 'Some complexity results for distance-based judgment aggregation', in *AI 2013: Advances in Artificial Intelligence - 26th Australasian Joint Conference, Dunedin, New Zealand, December 1-6, 2013. Proceedings*, pp. 313–325, (2013).
[20] E. M. Kok, J. J.Ch. Meyer, H. Prakken, and G. A. W. Vreeswijk, 'A formal argumentation framework for deliberation dialogues', in *Argumentation in Multi-Agent Systems: 7th International Workshop, ArgMAS 2010 Toronto, ON, Canada, May 10, 2010 Revised, Selected and Invited Papers*, eds., P. McBurney, I. Rahwan, and S. Parsons, volume 6614 of *Lecture Notes in Computer Science*, chapter Argumentation in Multi-Agent Systems, 31–48, Springer Berlin Heidelberg, Berlin, Heidelberg, (2011).
[21] J. Lang, G. Pigozzi, M. Slavkovik, and L. van der Torre, 'Judgment aggregation rules based on minimization', in *TARK*, pp. 238–246, (2011).
[22] J. Lang, G. Pigozzi, M. Slavkovik, L. van der Torre, and S. Vesic, 'Majority-preserving judgment aggregation rules', *CoRR*, **abs/1502.05888**, (2015).
[23] J. Lang and M. Slavkovik, 'Judgment aggregation rules and voting rules', in *Proceedings of the 3rd International Conference on Algorithmic Decision Theory*, volume 8176 of *Lecture Notes in Artificial Intelligence*, pp. 230–244. Springer-Verlag, (2013).
[24] O. Lev and J. Rosenschein, 'Convergence of iterative voting', in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '12, pp. 611–618, Richland, SC, (2012). International Foundation for Autonomous Agents and Multiagent Systems.
[25] C. List, 'Group communication and the transformation of judgments: An impossibility result', *Journal of Political Philosophy*, **19**(1), 1–27, (2011).
[26] L. List and B. Polak, 'Introduction to judgment aggregation', *Journal of Economic Theory*, **145**(2), 441 – 466, (2010).
[27] R. Meir, M. Polukarov, J. Rosenschein, and N. Jennings, 'Convergence to equilibria in plurality voting', in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, pp. 823–828, (2010).
[28] M.K. Miller and D. Osherson, 'Methods for distance-based judgment aggregation', *Social Choice and Welfare*, **32**(4), 575 – 601, (2009).
[29] K. Nehring and M. Pivato, 'Majority rule in the absence of a majority'. Presented at the New Developments in Judgement Aggregation and Voting Theory Workshop, September 2011.
[30] K. Nehring, M. Pivato, and C. Puppe, 'Condorcet admissibility: Indeterminacy and path-dependence under majority voting on interconnected decisions'. http://mpra.ub.uni-muenchen.de/32434/, July 2011.
[31] K. Nehring, M. Pivato, and C. Puppe, 'The condorcet set: Majority voting over interconnected propositions', *J. Economic Theory*, **151**, 268–303, (2014).
[32] H. Nurmi, 'Voting theory', in *e-Democracy*, eds., D. Rios Insua and S. French, volume 5 of *Advances in Group Decision and Negotiation*, 101–123, Springer Netherlands, (2010).
[33] S. Obraztsova, E. Elkind, P. Faliszewski, and A. Slinko, 'On swap-distance geometry of voting rules', in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '13, pp. 383–390, Richland, SC, (2013). International Foundation for Autonomous Agents and Multiagent Systems.
[34] S. Obraztsova, E. Markakis, M. Polukarov, Z. Rabinovich, and N. Jennings, 'On the convergence of iterative voting: How restrictive should restricted dynamics be?', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 993–999, (2015).
[35] S. Ontañon and E. Plaza, 'An argumentation-based framework for deliberation in multi-agent systems', in *Argumentation in Multi-Agent Systems: 4th International Workshop, ArgMAS 2007, Honolulu, HI, USA, May 15, 2007, Revised Selected and Invited Papers*, eds., I. Rahwan, S. Parsons, and C. Reed, volume 4946 of *Lecture Notes in Computer Science*, chapter Argumentation in Multi-Agent Systems, 178–196, Springer Berlin Heidelberg, Berlin, Heidelberg, (2008).
[36] I. M. Pelayo, *Geodesic Convexity in Graphs*, SpringerBriefs in Mathematics, 2013.
[37] G. Pigozzi, 'Belief merging and the discursive dilemma: an argument-based account to paradoxes of judgment aggregation', *Synthese*, **152(2)**, 285–298, (2006).
[38] G. Pigozzi, M. Slavkovik, and L. van der Torre, 'A complete conclusion-based procedure for judgment aggregation', in *1rst International Conference on Algorithmic Decision Theory, Venice, Italy, October 20-23, 2009. Proceedings*, eds., F. Rossi and A. Tsoukiàs, volume 5783 of *Lecture Notes in Computer Science*, pp. 1–13. Springer, (2009).
[39] M. Slavkovik, *Judgment aggregation for multiagent systems*, Ph.D. dissertation, University of Luxembourg, 2012.
[40] M. Slavkovik and T. Ågotnes, 'Measuring dissimilarity between judgment sets', in *Logics in Artificial Intelligence - 14th European Conference, JELIA*, pp. 609–617, (2014).
[41] M. Slavkovik and G. Boella, 'Recognition-primed group decisions via judgment aggregation', *Synthese*, (2012).
[42] W. Zwicker, 'Towards a Borda count for judgment aggregation'. Working Paper., 2011.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1537

Partial Order Temporal Plan Merging for Mobile Robot Tasks

Lenka Mudrova[1] and **Bruno Lacerda**[1] and **Nick Hawes**[1]

Abstract. For many mobile service robot applications, planning problems are based on deciding *how* and *when* to navigate to certain locations and execute certain tasks. Typically, many of these tasks are independent from one another, and the main objective is to obtain plans that efficiently take into account where these tasks can be executed and when execution is allowed. In this paper, we present an approach, based on merging of partial order plans with durative actions, that can quickly and effectively generate a plan for a set of independent goals. This plan exploits some of the synergies of the plans for each single task, such as common locations where certain actions should be executed. We evaluate our approach in benchmarking domains, comparing it with state-of-the-art planners and showing how it provides a good trade-off between the approach of sequencing the plans for each task (which is fast but produces poor results), and the approach of planning for a conjunction of all the goals (which is slow but produces good results).

1 INTRODUCTION

Consider a mobile service robot operating in an office building for a long period of time, where it autonomously performs tasks to assist the occupants in their everyday activities. One can imagine a wide array of tasks for such a robot to execute, for example:

- *"Bring me a cup of coffee."*
- *"Check if there are people in office 123."*
- *"Check if the emergency exits are clear."*

Note that these tasks have common properties that one can take advantage of:

1. They require the robot to navigate to certain locations to execute certain actions, i.e., they include *spatial constraints*;
2. The actions associated with them can be executed concurrently. For example, a sensing action is often fixed to a location, but processing and reasoning about the sensed data can be done in parallel with the robot's movement;
3. Their goals are independent, in the sense that executing a certain task ω_i is not a precondition to successfully execute task ω_j. This means that they can be straightforwardly split into as separate planning problems.

Furthermore, even though the goals are independent, the existence of spatial constraints means that there might be "synergies" between the independent plans, i.e. the locations visited while executing task ω_i might also be of use for executing task ω_j.

[1] School of Computer Science, University of Birmingham, UK, email: {lxm210, b.lacerda, n.a.hawes}@cs.bham.ac.uk

In this paper, we build on these insights to present an algorithm that, given a plan for each task, efficiently *merges* them into a single plan for all tasks, interleaving actions which work towards different goals.

Our merging algorithm is based on partial-order planning (POP), a least-commitment search in the space of (partial order) plans. POP presents clear benefits for our robot-oriented merging approach, including:

1. The least-commitment approach of POP yields more "merging points" between plans, when compared to a totally ordered plan;
2. POP presents a flexible approach to temporal planning with durative concurrent actions, allowing parallel action execution;
3. POP produces plans with more flexibility in execution as commitments can be determined at execution time when temporal information is more certain.

The main contributions of this paper are (i) the definition of a class of planning problems that are well-suited for the specification of *execution routines* for mobile service robots; and (ii) a partial order plan merging algorithm that is able to generate a plan for a large number of tasks, while taking advantage of possible synergies between such tasks, thus improving the overall robot's behaviour. For the class of problems we tackle in this paper, our approach is competitive with the performance of state-of-the-art forward chaining planners on benchmarking domains. Furthermore, the use of POP allows us to easily tackle concurrent actions, which allows us to outperform the state-of-the-art forward chaining planners in domains where reasoning about concurrent actions is required.

The structure of the paper is as follows. In Section 2, we provide an overview of state-of-the-art planners, and their limitations for our domain. In Sections 3 and 4 we introduce the background on partial order planning we rely on, formalise the problem we tackle, and describe possible solution approaches. Finally, Sections 5 and 6 present our novel plan merging algorithm, and its evaluation.

2 RELATED WORK

2.1 Temporal planners

In order to compare our proposed algorithm with state-of-the-art temporal planners, we focus on those that successfully participated in the temporal track of the latest International Planning Competition (IPC) in 2014[1]. The 6 following planners participated in the competition: YAHSP3 [30] pre-computes relaxed plans for estimated future states which are then exploited to speed-up the forward state-space search. YAHSP3-MT [31] is a multi-threaded extension of

[1] https://helios.hud.ac.uk/scommv/IPC-14/index.html

YAHSP. The YASHP3 planner is also exploited by another contestant, DAE$_{YAHSP}$ [4]. DAE uses a *Divide-and-Evolve* strategy in order to split the initial problem into a set of sub-problems, and to evolve the split based on the solutions found by its wrapped planner. In general, DAE$_X$ can be used with any planner, with the version we evaluate wrapping YAHSP. Two other participants extend well-known approaches to the temporal domain. First, the temporal fast downward (TFD) planner [10] expands the fast downward algorithm [13]. The search state is extended with a discrete timestamp component, and state expansion can be performed either by inserting an action or by incrementing the timestamp. Second, ITSAT [21] expands a satisfiability checking (SAT) approach to the temporal domain. ITSAT is the only planner from the aforementioned to handle concurrent actions properly.

2.2 Temporal Partial Order Planners

Another important class of temporal planners are those that provide a temporal partial order plan as a solution. Versatile Heuristic Partial Order Planner (VHPOP) [24] is one of the pioneers in this field. It builds on the classical backward plan-space search used by partial order planners, adding to it a set of different heuristics that allow for a more informed choice of which *flaw* to solve, or which plan to explore. The use of these heuristics yields large improvements in terms of speed, when comparing to previous partial order planners. In contrast, the more recent OPTIC [2] planer combines advantages of partial order planning with forward chaining techniques, which are very popular in current planners, due to their speed and scalability.

2.3 Planning & Execution for Service Robots

The CoBot service robots [29] operate in an office building performing several predefined tasks. A server-based architecture [9] manages incoming tasks from a web-based user interface, schedules tasks across several robots [8], and keeps track of task execution. A similar centralised system architecture is used by the mobile service robot Tangy [16] which performs a sequence of predefined tasks, taking the schedules of users into account. This problem is modelled by mixed-integer programming and constraint programming techniques [5]. Mixed-integer programming is also used for scheduling in the integrated control framework presented in [18]. In this work, a stochastic high-level navigation planner provides expectations on travel times between different locations to the scheduling algorithm. In contrast to the previous architectures, robots Rin and Rout use a constraint network [22]. This network is continuously modified by an executor, a monitor and a planner in order to create configuration plans which specify causal, temporal, resource and information dependencies between individual actions. All the above works are based on scheduling approaches, which rely on a coarse-grained model of the environment, where tasks are seen as black-boxes, being pre-specified instead of planned for, and with the scheduler trying to order them such that a set of timing constraints is satisfied. This does not allow for direct reasoning over the possible synergies between different tasks, and the possibility to interleave actions from different tasks in order to minimise execution time.

In recent years, there has also been work aiming at closing the loop between task planning and real world execution on a robot platform. The ROSPlan framework [6] is a general framework that allows for a task planner to be embedded into the Robot Operating System (ROS), a middleware widely used in the robotics community. As a proof of concept, ROSPlan has integrated the POPF planner [7], an ancestor of OPTIC. This integration with execution is also part of our future work, and we plan to explore how our techniques can be integrated in such an execution framework.

Additionally, some modelling languages have been developed with the goal of having a closer integration between planning and execution. Of particular interest are the NDDL [3] and the ANML [25] modelling languages. These are based on the notion of *timelines*, i.e., sequences of *tokens*. A token is a timed predicate that holds within a start and end time. The timeline representation was developed by NASA and used in open-source project EUROPA [1] in order to model and plan for real world problems and to allow a close integration with the execution of such plans. This representation was also exploited in T-REX [17], a model-based architecture for robot control which used a tight integration of planning and execution. Another system closely integrating planning and execution is FAPE [20], built on the ANML language. Unfortunately, these system based on timelines do not have the scalability of the state-of-the-art planning approaches presented above.

2.4 Merging Algorithms

The first planning system to use merging was probably NOAH [23], as described in [11]. In NOAH, three criteria were introduced to handle possible interactions between plans: eliminate redundant preconditions, use existing operators, and optimise disjuncts. NONLIN [26] is also able to recognise if any goal is achievable by an operator already in a plan. If such operator is detected, then ordering constraints and variable bindings are used to change the plan such that the found operator is used to fulfil the goal.

Temporal and conditional plan merging is done in [27] which extends the work of Yang [33]. For two input plans, the algorithm provides a new order of actions while detecting and removing redundant actions, by checking if their effects are already fulfilled by some preceding action.

Related techniques to plan merging are *plan repair* and *plan refinement*. Refinement planning focuses on how to introduce a new action to an existing plan, and was introduced in [15]. Work in [14] uses a *partial plan* to save current refinements. The opposite case, i.e., removing an action from the plan, is handled in *unrefinement planning* [28]. This addresses the *plan repair* problem of changing a plan when it cannot be executed. Despite the fact that it was proved that modifying an existing plan is no more efficient than a full (re)planning in the general case [19], plan repair might still be efficient in certain domains. An example of recent plan refinement is planning for highly uncertain domains, such as underwater robotics [12]. In this case, one plan achieving a subset of tasks is produced. While it is executed, the current state is observed in order to limit uncertainties. If the robot has unexpected available resources, allowing it to perform more tasks, a pre-computed plan achieving another task is inserted into the global plan. Our proposed algorithm combines ideas from aforementioned merging approaches in order to allow flexible execution on a mobile robot.

3 PARTIAL ORDER PLANNING

We start by introducing the definitions for POP that will be used throughout the paper. For a thorough overview of POP see [32]. Our merging algorithm assumes that plans have already been generated, so all actions we deal with are grounded. Thus, we omit details about lifted actions and bindings when describing the planning problems.

We start by defining a task in our framework. A task domain is a set $D = \{f_1, .., f_n\}$ of atomic formulas (atoms); literals – formulas and their negations $L_D = \{f_1, \neg f_1, \dots\}$ – are used to describe a given state of the world. A state in this domain is represented by $L \subseteq L_D$, such that either f or $\neg f$ are in L. A literal l is satisfied in L if $l \in L$.

A durative action $a \in A$ consists of its start point $a_\vdash$ and end point $a_\dashv$, and we define the sets $A_\vdash$ and $A_\dashv$ of action start points and end points, respectively. Preconditions $pre(a)$ of a durative action a are a set of *timed* literals, which are literals which must hold at a specific action point. We recognise the *at start* literal which must hold at the action start point $a_\vdash$, the *at end* literal which holds at the action end point $a_\dashv$ and the *over all* literal which must hold during whole action. Hence, if we refer to $pre(a_\vdash)$ we mean only those literals which are meant to hold *at start* or *over all*, etc. A set of action effects $\mathit{eff}(a)$ is a set of timed literals as well but only *at start* or *at end* extensions are assumed, as an *over all* effect is same as *at start* effect. The duration of an action is $d(a) \in \mathbb{R}$.

Finally, a planning problem is defined as $P = (I, G)$, where $I \subseteq L_D$ is the initial state, and $G \subseteq L_D$ is the goal. We say that a state L achieves the goal if $G \subseteq L$. A task is then defined as $\omega = \langle D, A, P\rangle$.

A partial order plan (POP) is a tuple $\pi = \langle \mathcal{A}, \mathcal{L}, \mathcal{O}\rangle$, where:

- $\mathcal{A}$ is a set of actions;
- $\mathcal{L}$ is a set of causal links. A causal link $a_j \xrightarrow{l} a_k$ represents that literal $l \in pre(a_k)$ is fulfilled as an effect of action a_j;
- $\mathcal{O}$ is a set of ordering constraints defining a partial order on the set $\mathcal{A}$. An ordering constraint $a_j \prec a_k$ represents that action a_j must finish before action a_k can start.

Given a causal link $a_j \xrightarrow{l} a_k$, we refer to a_j as the *producer* of literal l and to a_k as the *consumer* of literal l. Given a POP π, an open condition $\xrightarrow{l} a_j$ means that literal $l \in pre(a_j)$ has not yet been linked to the effect of any action. An unsafe link (or a threat) is a causal link $a_j \xrightarrow{l} a_k$ such that there is an action $a_m \in \mathcal{A}$ that could possibly be ordered between a_j and a_k and threatens $a_j \xrightarrow{l} a_k$ by having $\neg l \in \mathit{eff}(a_m)$. The set of flaws of a POP π is given by the union of its open conditions and unsafe links. A POP planner searches through the space of POPs trying to resolve all flaws of π. To do that, the planner tries to close open conditions by adding new actions to $\mathcal{A}$ and new causal links to $\mathcal{L}$, and solve threats by adding new orderings to $\mathcal{O}$ to make sure that the threatening action a_m does not occur between the unsafe link $a_j \xrightarrow{l} a_k$. This can be done by, for example, *promotion*, i.e., adding the constraint $a_k \prec a_m$ to $\mathcal{O}$, or *demotion*, i.e., adding $a_m \prec a_j$ to $\mathcal{O}$. In this work, we use the VHPOP planner as described in Section 2.2.

4 PROBLEM DEFINITION AND SOLUTION APPROACHES

In this paper, we are interested in solving the problem of finding a POP for a given set of input tasks. More specifically, given a set of tasks $\Omega = \{\omega_1, \dots, \omega_n\}$, where $\omega_i = \langle D_i, A_i, (I_i, G_i)\rangle$ and the initial states of each problem are mutually consistent (I_j is consistent with I_k if for all $l \in L_{D_j} \cap L_{D_k}$, $l \in I_j$ if and only if $l \in I_k$), we want to find a plan that achieves $G_1, ..., G_n$.

There are several ways of solving such a problem. In this section, we present three different approaches: *unifying*, *sequencing* and *plan merging*. We argue that the merging approach provides a good trade-off between the plan quality of the unifying approach and the efficiency of the sequencing approach, hence, in the next section we present a plan merging algorithm to solve our problem.

4.1 Unifying planning algorithm

This approach relies of *unifying* the set of tasks into a single task, i.e., $\omega = \left\langle \bigcup_{i\in\{1\dots n\}} D_i, \bigcup_{i\in\{1\dots n\}} A_i, (\bigcup_{i\in\{1\dots n\}} I_i, \bigcup_{i\in\{1\dots n\}} G_i)\right\rangle$ Then, one can use an appropriate planning algorithm to find a solution for the single unified task. While this approach can more easily take advantage of relations between goals in different tasks (e.g., two tasks that should be executed in the same location), it can suffer from scalability issues as finding a plan for the unified task can be much harder than finding plans for each individual task by itself. This approach is used by most planners, such as VHPOP, OPTIC and all presented planners from IPC 2014. Generally, the unified tasks are modelled a priori and passed directly as an input to such systems.

4.2 Sequencing planning algorithm

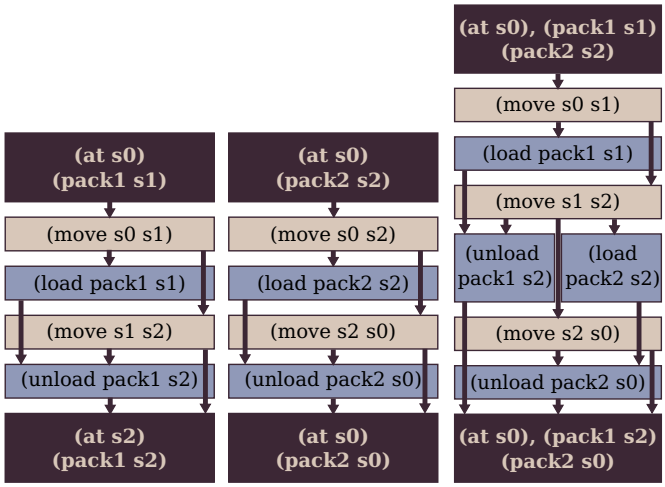
This approach generates a set of independent plans $\Pi_\omega = \{\pi_1, \dots, \pi_n\}$, one for each task ω_i, and then *sequences* them to create a single final plan. For the resulting plan to be valid, one needs to decide on an ordering of the tasks and then modify the initial states of each task according to the final state of the plan for the preceding task. The ordering of the tasks can be done using a *scheduling algorithm* that can take into account extra timing constraints on the execution of tasks. This approach is common for mobile service robots, e.g., [29, 18], due to its simplicity and efficiency. This is because the planning problems to be solved when planning for the tasks independently will in general be much smaller than the single unified one problem. However, simple sequencing comes at the price of plan quality: this approach does not allow for the interleaving of actions from plans for different tasks, taking advantage of synergies between them.

4.3 Merging planning algorithm

This approach combines both aforementioned methods. It also plans for tasks separately, obtaining $\Pi_\omega = \{\pi_1, \dots, \pi_n\}$ but then it reasons over each plan, *merging* them together into a better plan than the one obtained by simple sequencing. The final plan π_f consists of parts of the task plans in Π_ω and newly created plans Π_{join} which are used in order to connect these parts such that the final plan is free of flaws. Furthermore, while the merging procedure adds an overhead at plan generation time when compared to the sequencing approach, it allows us to find synergies between plans for different tasks, interleaving execution for different goals. A typical benefit of this approach in the mobile robot domain is the possibility to execute actions from different tasks when these actions share a common location. The algorithm we present in the next section follows this approach.

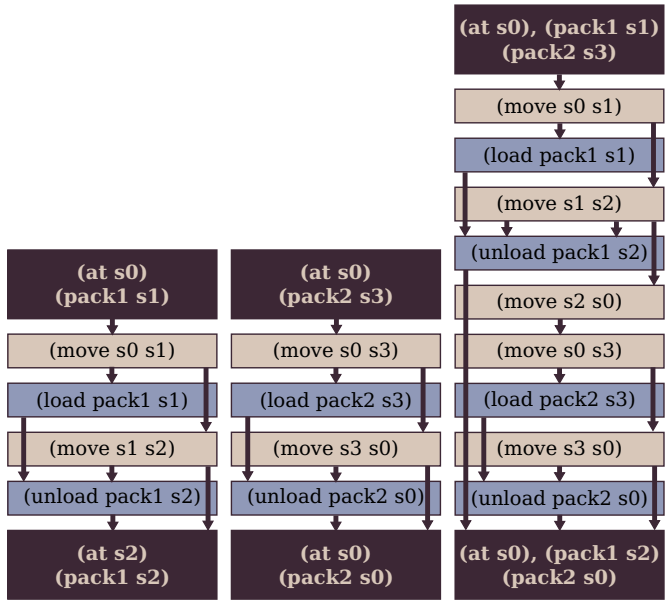
5 PROPOSED ALGORITHM

In this section, we present our merging algorithm. Before we describe it, we need to address an issue that can hinder the performance of the merging algorithm, and present a solution for it.



(a) Plan π_{1a} (b) Plan π_{2a} (c) Plan π_{fa}

Figure 1. Single plans π_{1a}, π_{2a} for tasks ω_1, ω_2 respectively and the merged plan π_{fa}. The arrows represent the causal links between actions. The order of actions in the plan corresponds to the orderings, for example two concurrent actions are parallel in the final plan.



(a) Plan π_{1b} (b) Plan π_{2b} (c) Plan π_{fb}

Figure 2. Single plans π_{1b}, π_{2b} for tasks ω_1, ω_2 respectively and the merged plan π_{fb}. The arrows represent the causal links between actions. The order of actions in the plan corresponds to the orderings.

5.1 Dependency Caused by External Constraints

5.1.1 Problem Illustration

As stated in the introduction, we assume independent tasks, i.e., tasks where the goals can be partitioned and for which the execution of a plan for task ω_i is not a precondition to successfully execute task ω_j. However, these tasks can become dependent due to external constraints. In mobile robots domains, these are typically spatial constraints. We will illustrate this problem on the following example.

DeliveryBot A mobile robot delivers packages around a building. A robot can *move between locations* (with duration 10), *load packages* (with duration 2), and *unload packages* (with duration 2). The robot receives two tasks:

1. ω_1: *"Deliver package1 to location s2."*
2. ω_2: *"Deliver package2 to location s0."*

Assume the initial state is $I = \{$*(at s0), (pack1 s1), (pack2 s2)*$\}$, and a partial order planner produces optimal plans π_{1a} and π_{2a}, as depicted in Fig. 1a and Fig. 1b. An example of a final merged plan, π_{fa}, with makespan 38 is also given in Fig. 1c. Notice that action *(move s0 s2)* from plan π_{2a} is not used as its effects are satisfied by action *(move s1 s2)* from plan π_{1a}. However, if the initial state would be $I = \{$*(at s0), (pack1 s1), (pack2 s3)*$\}$, a partial order planner would produce the plan π_{2b} which is again optimal, see Fig. 2b. In this case, action *move(s0 s3)* will need to be merged as well as its effects are not satisfied. Hence, the output plan has makespan 58, see Fig. 2c. However, an optimal plan will not contain *(move s2 s0), (move s0 s3)* and will instead directly use action *(move s2 s3)*. Thus, the optimal plan for the two goals has makespan 48.

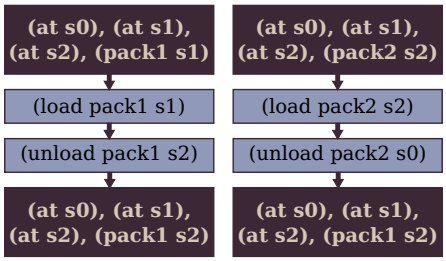
The problem during merging is that an action in an input plan is linked to a preceding state which contains all its preconditions. However, during merging, the action might be chosen to be merged into the final plan in a state which does not achieve all of the action's preconditions. In such cases, *filling actions* (i.e., actions whose effects change the current state to contain all needed preconditions) need to be added to the final plan before the candidate action is merged. This extends the makespan of the final plan, hence we would like to minimise the occurrence of such filling actions. Furthermore, we note that actions related to the external constraints cause unnecessary joining actions. For example, action *(move s2 s0)* in plan π_{fb} is a filling action, required for action *(move s0 s3)* from the original plan to be merged.

As discussed in the introduction, actions related to robot movement (in this case, action *move*) are typically a significant contributor to the makespan of plans generated on a mobile service robot domain. Hence, minimising their occurrence in the plan generally also allows for reducing on the makespan. Therefore, addressing these dependencies caused by external constraints before merging can lead to significant improvements to the quality of the merged plans.

5.1.2 Preprocessing External Constraints

In order to address the problem described above, we create relaxed planning problems that have their initial states extended by literals which satisfy the external constraints. Therefore the resulting plans for these relaxed problems will not contain any external constraints. By removing the external constraints from the input problems, we allow more freedom for the merging algorithm o merge them together, resulting in a final plan with better makespan.

In mobile service robot domains, the external constraints are related to the position of the robot. Therefore in our in the DeliveryBot example we add *(at s0), (at s1), (at s2)* to the initial states for the individual task plans, see Fig. 3a and Fig. 3b. We will address the automated detection of external constraints in future work.



(a) $\pi_1^\times$ (b) $\pi_2^\times$

Figure 3. Relaxed plans created by adding all locations to the initial states.

5.2 Plan Merging Algorithm

Given a set of relaxed plans, the proposed algorithm for plan merging $POMer_X$ incrementally extracts actions from the input plans and greedily adds an action start point $a_\vdash$ or an action end point $a_\dashv$ per iteration to the merged plan. Action end points are added depending on their temporal constraints, but action start points are added based on whether they require a *joining* subplan. A joining subplan is required by an action start if its preconditions are not satisfied by the effects of actions already merged into the final plan. This is caused by two factors. First, the relaxation of the input plans removes external constraints that need to be addressed during plan merging. Second, the greedy merging of actions might cause for an interleaving of actions from different plans that yield certain action inapplicable. We refer to an action requiring a *joining* plan as *inapplicable* in the current state.

If there exists action starts which do not require a *joining* plan, the algorithm picks one of them to be merged. Otherwise, $POMer_X$ calls its wrapped POP planner X in order to find the joining subplan that satisfies the preconditions and connects the current state of the merged plan to the preconditions of the new action start point. Then, the greedy decision is made over all *joining* subplans, with a single joining subplan with the minimal duration being chosen; its actions are then extracted and picked to be merged the same way as actions from the input plans.

Furthermore, $POMer_X$ allows for backtracking when the greedy choices result in an intermediary plan that does not satisfy the temporal constraints or causal links of the input plans. Thus, while not optimal, $POMer_X$ is guaranteed to find a solution for the merging, if one exists.

We now describe the main steps of the algorithm. Let $\Omega = \{\omega_1, ..., \omega_n\}$ be a set of tasks, with $\omega_i = \langle D_i, A_i, (I_i, G_i)\rangle$, and $\Pi^\times = \{\pi_1^\times, ..., \pi_n^\times\}$ be a set of relaxed plans, where $\pi_i^\times$ is the plan obtained for some relaxation $\langle D_i, A_i, (I_i^\times, G_i)\rangle$ of ω_i. The input of Algorithm 1 is the initial state of the (unrelaxed) global problem $\mathcal{I} = \cup_{i\in\{1,...,n\}} I_i$, the set $\Pi^\times$ of relaxed plans, and the set $\mathcal{G} = \cup_{i\in\{1,...,n\}} G_i$, of goal states for each task. The algorithm then calculates a merged plan π_s such that all goals $G_1, ..., G_n$ are satisfied, when applying plan π_s to the initial state $\mathcal{I}$. To construct the final plan π_s, the algorithm searches over *merging states* of the form

$$S = \langle t, L, \mathcal{A}^+, \mathcal{A}^-, Q_\vdash, Q_\dashv, L_{block}, \Pi_{join}, \Pi_\omega^\times, \pi_s\rangle \qquad (1)$$

where:

- t is duration of the current plan π_s;
- $L = \{F^+, \neg F^-\}$ is the set of literals that hold in S, where sets F^+, F^- contains atoms that hold and do not hold in the current state, respectively.
- A^+ is the set of *achievers* of all atoms that hold in the current state. The achiever of f, denoted $A^+(f)$, is the last action a added to π_s such that $f \in \mathit{eff}(a)$.
- A^- is the set of *deleters* of all atoms that do not hold in the current state. The deleter of f, denoted $A^-(f)$, is the last action a added to π_s such that $\neg f \in \mathit{eff}(a)$.
- $Q_\vdash \subseteq A_\vdash$ is a queue of action start points that are extracted from the input plans to be merged into plan π_s;
- $Q_\dashv \subseteq A_\dashv \times \mathbb{R}$ is a queue of action end points for actions for which the start point has already been merged into π_s. Each end point is of the form $(a_\dashv, t_{a_\vdash} + d(a))$, where $t_{a_\vdash}$ is the time point when the start of a was merged into π_s;
- $L_{block} \subseteq L \times A_\vdash$ is a set of *blocked* literals. Each blocked literal $(l_{block}, a_\vdash) \in L_{block}$ is such that the starting point $a_\vdash$ of action a has been already merged to π_s in some previous state, the end point $a_\dashv$ of a is not yet merged, $l_{block} \in pre(a)$, and it must hold *over all* duration of the action. Therefore, l_{block} must hold until $a_\dashv$ is merged, thus removing the blocked literal; the validity of blocked literals cannot be changed by other actions in the merging process.
- Π_{join} is a set of *joining* POPs satisfying preconditions of actions in $Q_\vdash$.
- $\Pi_\omega^\times$ is the set of all input plans for all tasks.
- $\pi_s = \langle \mathcal{A}_{\pi_s}, \mathcal{L}_{\pi_s}, \mathcal{O}_{\pi_s}\rangle$ is the POP that reaches the current state.

At each step on its main loop, Algorithm 1 starts by analysing the queue of action end points $Q_\dashv$ to check if there are action end points that must be merged into the plan at the current state, given their temporal constraints. If so, the action end point is merged into π_s. Otherwise, the algorithm proceeds by adding to $Q_\vdash$ the starts points of actions a in each relaxed input plan for which the following three conditions hold. First, a has not been merged before; second, the *producing* actions in the input causal links where a is a *consumer* have already been merged; and third, the actions that must be ordered before a are already merged (lines 8–14). The merged actions, links and orderings $\langle \mathcal{A}_m, \mathcal{L}_m, \mathcal{O}_m\rangle$ are obtained by a method *merged-subplan*($\pi_i^\times$) which for a given plan returns a subplan that is already *merged* in the current state S, i.e., $\mathcal{A}_m \subseteq \mathcal{A}_{\pi_s}, \mathcal{L}_m \subseteq \mathcal{L}_{\pi_s}, \mathcal{O}_m \subseteq \mathcal{O}_{\pi_s}$. If all merged subplans are equal to the input plans, the algorithm successfully merged all the plans and it returns the final plan. (line 17). Otherwise, if $Q_\vdash$ is not empty, method *pickActionStart*($Q_\vdash$) (see Algorithm 2) will choose an action start point that is not affecting any of the blocked literals in L_{block}. All action starts affecting blocked literals are removed from $Q_\vdash$.

In the cases where $Q_\vdash$ is empty, i.e., no action start points can be extracted from the input plans, or all action start points were removed due to affecting blocked literals, there is no action start point that can be merged at the current state. Hence, the next action end point to be merged (i.e., the one with the earliest deadline) is merged instead, if one exists. If no action end point exists either, then we have reached a state for which it is not possible to merge actions while satisfying the input plans constraints. Therefore, the algorithm backtracks to the parent state, removing the last merged action from the plan and propagates it as an invalid choice. If the initial state is reached by backtracking, it means that the plans cannot be merged while maintaining their individual constraints, and the algorithm outputs that there is no solution.

Algorithm 2 chooses one action start point $a_\vdash \in Q_\vdash$ to be merged into the final plan. It starts by checking which actions in $Q_\vdash$ are *in-*

Algorithm 1 POMer$_X(\Pi^\times, \mathcal{I}, \mathcal{G})$

1: $S = \langle 0, \mathcal{I}, \mathcal{A}_0^+, \mathcal{A}_0^-, \emptyset, \emptyset, \emptyset, \emptyset, \check{\Pi}_\omega^\times, \langle \{start\}, \emptyset, \emptyset \rangle \rangle$
2: **while true do**
3: **if** $\exists (a_\dashv, t_\dashv) \in Q_\dashv$ such that $t = t_\dashv$ **then**
4: remove $(a_\dashv, t_\dashv)$ from $Q_\dashv$
5: merge($a_\dashv$)
6: **else**
7: $allMerged$ = **true**
8: **for** $\pi_i^\times = \langle \mathcal{A}^\times, \mathcal{L}^\times, \mathcal{O}^\times \rangle \in \Pi^\times$ **do**
9: $\langle \mathcal{A}_m, \mathcal{L}_m, \mathcal{O}_m \rangle$ =merged-subplan($\pi_i^\times$)
10: **if** $\langle \mathcal{A}_m, \mathcal{L}_m, \mathcal{O}_m \rangle \neq \langle \mathcal{A}^\times, \mathcal{L}^\times, \mathcal{O}^\times \rangle$ **then**
11: $allMerged$ = **false**
12: **end if**
13: $\{a \in \mathcal{A}^\times \setminus \mathcal{A}_m \mid \forall (a' \xrightarrow{l} a \in \mathcal{L}^\times), \mid \forall (a' \prec a \in \mathcal{O}^\times),\ a' \in \mathcal{A}_m^\times\}$
14: $Q_\vdash = Q_\vdash \cup a_\vdash$
15: **end for**
16: **if** $allMerged$= **true then**
17: **return** π_s
18: **end if**
19: **if** $Q_\vdash \neq \emptyset$ **then**
20: $\{a, Q_\vdash\}$=pickActionStart($Q_\vdash$)
21: **if** $a \neq \emptyset$ **then**
22: $merge(a)$
23: **continue**
24: **end if**
25: **end if**
26: **if** $Q_\vdash = \emptyset$ **then**
27: **if** $Q_\dashv \neq \emptyset$ **then**
28: pick $a_\dashv \in Q_\dashv$ with the soonest deadline
29: merge($a_\dashv$)
30: **else**
31: S = backtrack()
32: **end if**
33: **else**
34: go to line 19
35: **end if**
36: **end if**
37: **end while**

applicable. An action start $a_\vdash$ is *inapplicable* in the current state if at least one of its preconditions do not hold in the current state. This can happen if another action deleting literal l was merged after the achiever of l; or the literal was relaxed in the input plan. Thus, in line 1, we build the set of *applicable* of action starts, i.e., the set of action start points that are not *inapplicable* at the current search state. If there are no applicable actions at the current state, we generate, for each $a_{vdash} \in Q_\vdash$, *joining* plans. To do so, we use a call to the wrapped POP planner, defining the current set of literal L as the initial state, and the preconditions of the action as the goal state (line 3). From the set of joining plans, we then choose the one that has the shortest duration, and insert its applicable actions to the *applicable* set (lines 4 and 5).

Then, we remove from *applicable* and $Q_\vdash$ all actions that affect the *blocked* literals in L_{block}. Note that the actions removed from $Q_\vdash$ will, in some future state, be again extracted from the input plans and will be potentially merged into the final plan. If after removing the actions that affect L_{block}, *applicable* is empty, then the algorithm returns an empty action, else it non-deterministically picks and returns an action start point $a_\vdash \in applicable$.

Algorithm 2 pickActionStart($Q_\vdash$)

1: $applicable = \{a_\vdash \in Q_\vdash \mid pre(a_\vdash) \subseteq L\}$
2: **if** $applicable = \emptyset$ **then**
3: $\Pi_{join} = \{X(D, A, \langle L, pre(a_\vdash) \rangle) \mid a_\vdash \in Q_\vdash\}$
4: pick $\pi_{join} = \langle \mathcal{A}_{join}, \mathcal{L}_{join}, \mathcal{O}_{join} \rangle$ from Π_{join} that has the shortest duration
5: $applicable = \{a_\vdash \in \mathcal{A}_{\vdash join} \mid pre(a_\vdash) \subseteq L\}$
6: **end if**
7: remove from $Q_\vdash$ and $applicable$ all actions a that do not affect L_{block}, i.e., $\forall l \in \mathit{eff}(a) : (\neg l, .) \notin L_{block}$
8: **if** $applicable = \emptyset$ **then**
9: **return** $\{\emptyset, Q_\vdash\}$
10: **else**
11: pick action a from $applicable$
12: **return** $\{a, Q_\vdash\}$
13: **end if**

Finally, Algorithm 3 updates the current state S depending on the chosen action point, i.e., an action start or end, and merges it into the plan π_s. It proceeds as follows. First, if an action end is being merged, the time is updated by the corresponding time deadline of the action. An action start cannot move the time forward. Then, the set of active literals, achievers and deleters are updated by the effects of the action point; additionally the set of blocked literals is updated by preconditions (line 11 – 16). The action point is removed from the particular queue and added to the set of actions A_{π_s}. For each precondition of the action point, new causal link and corresponding ordering is added between achiever or deleter (for negative literals) and the merged action point. Finally, the added action point might be ordered in parallel to another action and it might threaten an existing causal link (line 22). In such a case, the action point is *promoted*.

Algorithm 3 merge(a_{inst})

1: $t = (a_{inst} \in Q_\dashv)\ ?\ t = Q_\dashv[a_{inst}]$
2: $L = L \cup \mathit{eff}(a_{inst})$
3: **for all** $f \in \mathit{eff}^+(a_{inst})$ **do**
4: $A^+(f) = a_{inst}$
5: **end for**
6: **for all** $f \in \mathit{eff}^-(a_{inst})$ **do**
7: $A^-(f) = a_{inst}$
8: **end for**
9: $(a_{inst} \in Q_\vdash)\ ?\ Q_\vdash = Q_\vdash \setminus a_{inst}$
10: $(a_{inst} \in Q_\dashv)\ ?\ Q_\dashv = Q_\dashv \setminus a_{inst}$
11: **for all** $l \in pre(a_{inst})$ **do**
12: **if** $a_{inst} \in Q_\vdash$ **and** l is *over all* **then**
13: $L_{block} = L_{block} \cup l$
14: **end if**
15: **if** $a_{inst} \in Q_\dashv$ **and** l is *over all* **then**
16: $L_{block} = L_{block} \setminus l$
17: **end if**
18: **end for**
19: $\mathcal{A}_{\pi_s} = \mathcal{A}_{\pi_s} \cup a_{inst}$
20: $\forall l \in pre^+(a_{achi}) : \mathcal{L}_{\pi_s} \cup \langle A^+(l) \xrightarrow{l} a_{inst} \rangle$, $\mathcal{O} \cup \langle A^+(l) \prec a_{inst} \rangle$
21: $\forall l \in pre^-(a_{inst}) : \mathcal{L}_{\pi_s} \cup \langle A^-(l) \xrightarrow{l} a_{inst} \rangle$, $\mathcal{O} \cup \langle A^-(l) \prec a_{inst} \rangle$
22: A_{threat} = threats(a_{inst})
23: $\forall a_{threat} \in A_{threat} : \mathcal{O}_{\pi_s} \cup \langle a_{threat} \prec a_{inst} \rangle$

5.3 Example

The flow of the algorithm is illustrated on the DeliveryBot example. The input to the algorithm is the conjunction of the original initial states, i.e., $\mathcal{I} = \{$*(at s0)*$\}$, the conjunction of the goals $\mathcal{G} = \{$*(pack1 s2), (pack2 s0)*$\}$ and the extracted plans $\pi_1^\times, \pi_2^\times$, see Fig. 3a and Fig. 3b. At the first iteration, the *merged-subplans* are empty, hence Algorithm 1 on line 13 chooses from all actions in the input plans. First, actions from the input plan on Fig. 3a are being extracted; action *(load pack1 s1)* has two preconditions: *(at s1), (pack1 s1)*. The achiever of both preconditions is the *start* action in the relaxed plan. Hence, this action is extracted in order to be merged. Then, *(unload pack1 s2)* action is tested. This action has two preconditions as well: *(pack1 truck)* and *(at s2)*. The achiever of the first precondition is the action *(load pack1 s1)*, however that action is not yet merged in plan π_s. Hence, action *(unload pack1 s2)* cannot be extracted to be merged. The same reasoning is done for the second plan $\pi_2^\times$.

After that, Algorithm 2 proceeds by extracting *applicable* actions from $Q_\vdash = \{$*(load pack1 s1), (load pack2 s2)*$\}$. Even though both actions have satisfied preconditions in the input relaxed plan, they are not satisfied in the current set of literals $L =$*(at s0)* due to relaxation. Therefore, the *applicable* set is empty and the wrapped planner X is called in order to obtain *joining* plans to achieve the preconditions of actions in $Q_\vdash$. The plan $\pi_{join-1} = \langle\{a_{1-1} =$*(move s0 s1)*$\}, \{start \xrightarrow{(at\ s0)} a_{1-1}\}, \{start \prec a_{1-1}\}\rangle$ is obtained, as is the similar π_{join-2} containing only action *(move s0 s2)*. In this particular case, the plans contain only a single action but in general, they can contain more. The temporary plan with the minimal duration is chosen; in this example, both plans have same duration, hence one plan is chosen randomly and *applicable* actions are extracted from the plan, i.e. *applicable* =*(move s0 s1)*. Finally, one action start point, in this case *(move s0 s1)*$\vdash$ is merged into the plan π_s.

Now we illustrate the meaning of *blocked* literals. In the third iteration, the action *(load pack1 s1)* is chosen; it has precondition *(at s1)* which must be valid *over all* duration of the action. Hence, $L_{block} =$ *(at s1)*. In the next iteration, actions' start *(move s1 s2)*$\vdash$ from the *applicable* set has $\neg$*(at s1)* as an effect which will change the blocked literal. Therefore, it cannot be chosen and set $Q_\vdash$ will become empty. Thus, Algorithm 1 proceeds to line 28 and chooses *(load pack1 s1)*$\dashv$ from the queue $Q_\dashv$ and merges it.

6 EVALUATION

We have developed a version of our algorithm POMer$_{VHPOP}$ which embeds the VHPOP planner [24] for plan generation for individual tasks[2]. In this section, we evaluate the POMer$_{VHPOP}$ algorithm and compare it with other temporal planners based on plan quality, measured using the makespan of the found solution, and scalability. For each found plan, we run VAL, the validator of PDDL plans [3] in order to ensure that the plan is valid for domain and problem. The evaluated planners maximum memory usage was limited to 8 GB. All evaluation was run on Lenovo ThinkPad E-540 with Intel i74702MQ Processor (6MB Cache, 800 MHz).

6.1 Domains and problems

We evaluate using domains taken from IPC 2014. However, we generate our own planning problems in these domains, as our algorithm is based on the assumptions that tasks, i.e., goals in problems, are independent. Moreover, we assume only a single agent performing the actions.

6.1.1 Drivelog domain

The Drivelog domain is the IPC 2014 domain closest to our main focus of mobile service robots as it has spatial constraints. In order to generate problems, we take the hardest problem from IPC 2014, i.e., problem P_{23}, and modify it so that it has a single agent, i.e., one truck with the driver already boarded. Then, we split the 23 goals for package placing into 23 single tasks. For the merging algorithm the initial state of these single tasks is extended by adding all atoms related to the spatial constraints, i.e., all *(at ?loc)* where *?loc* stands for any location in the problem.

6.1.2 TMS domain

This domain is another benchmarking domain for IPC 2014 which requires *concurrent* actions, a type of problem for which POP problems are especially suited. Even though this domain is about producing ceramic pieces, we choose it in order to demonstrate the capability of POMer$_{VHPOP}$ to handle concurrent actions. In this domain, a kiln represents the agent. Hence, our problems contain initial state that a kiln is always ready. We take the hardest problem from IPC 2014 and from it create 17 problems. The smallest problem contains 2 goals and the largest 50 goals.

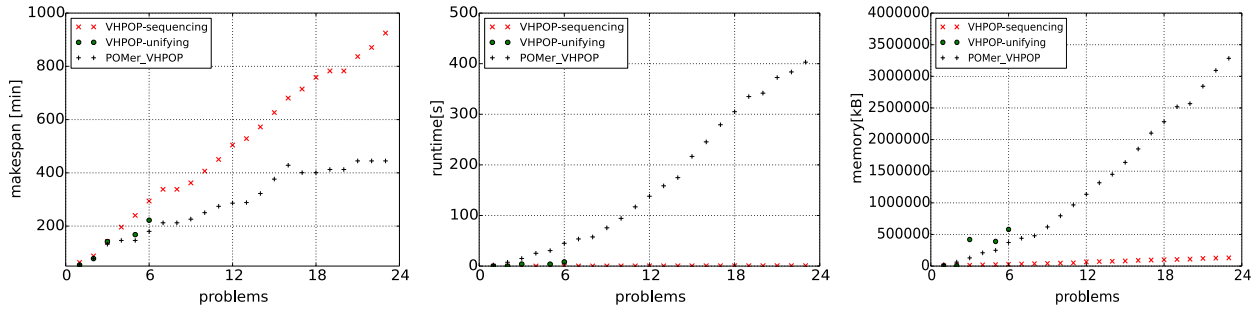
6.2 POMer$_{VHPOP}$ in comparison to VHPOP

First, we analyse how our proposed algorithm improves over its wrapped planner, in this case VHPOP. Thus we compare three algorithms: POMer$_{VHPOP}$, VHPOP used to solve the unified problem, and VHPOP used to solve the sequencing problem. All algorithms were run on problems for Drivelog domain for 30 min and could use 8 GB of memory. The makespans are depicted in Fig. 4a. VHPOP-unifying is able to find a solution for only five problems before it reaches the memory limit. We also report on time and memory consumed, see Fig. 4b and Fig. 4c, respectively. As expected, VHPOP-unifying consumes the most memory for most of the cases and in problem 4, and problems 6-23, it does not find a solution before the limit of 8 GB is reached. In contrast, the sequencing approach is the fastest and the most memory efficient however it always finds the worst makespan. For the largest problem, the makespan found by the sequencing approach is double the one found by POMer$_{VHPOP}$. This means that if the makespan is expressed in duration POMer$_{VHPOP}$ saves about 460 min in the biggest problem comparing to the fast sequencing approach even though it takes up to 7 min to provide a solution. To summarise, we can state that our merging algorithm wrapped around VHPOP significantly improved scalability of standalone VHPOP. As POMer$_{VHPOP}$ is not yet optimised, it is the slowest approach.

6.3 POMer$_{VHPOP}$ in comparison to IPC planners

This evaluation is focused on comparing properties of our proposed algorithm POMer$_{VHPOP}$ with the state of the art planners from IPC 2014, such as DAE$_{YAHSP}$, YAHSP3-MT, TFD and ITSAT, as described in Section 2. Additionally, we also compare to the POP-based OPTIC planner [2] and to VHPOP using the sequencing solution.

[2] Available at https://github.com/mudrole1/POMer
[3] http://www.inf.kcl.ac.uk/research/groups/PLANNING/



(a) Makespan (b) Time (c) Memory

Figure 4. Comparision of POMer$_{VHPOP}$, VHPOP-unifying and VHPOP-sequencing.

6.4 Drivelog domain

For each problem from the Drivelog domain, the aforementioned planners were run for the duration POMer$_{VHPOP}$ required for the same problem, see Fig. 4b. In order to express a quality of the found makespan, we introduce estimates of the best and the worst makespan. As the worst estimate, we use makespan found by VHPOP in sequencing approach and as the best estimate, we run DAE$_{YAHSP}$ for 30 minutes. Fig. 5 shows the recorded makespan for each problem. Even though the makespan is not a continuous function, we visualise the worst and the best estimates as a line in order to highlight these limits.

Note that DAE$_{YAHSP}$ struggles in problems 7 and 11 to provide a good solution. In both cases, the found solution is even slightly worse than the worst estimate. Hence, we exclude these problems from the following analysis. Our algorithm is better than the best estimate in three problems by total difference 46.9. This means that an average difference per plan is 15.63 units in which makespan is recorded. As all packages must be loaded and unloaded in both plans, the POMer$_{VHPOP}$ has only two options how to find better plan - place loading and unloading actions concurrently or find better path between locations. In 20 cases, POMer$_{VHPOP}$ found worse solution than the best estimate by a total difference of 327.88, or 16.39 average difference. Most of these cases were problems with more goals, which is expected as the greedy heuristics are driven to local optima more often in bigger problems.

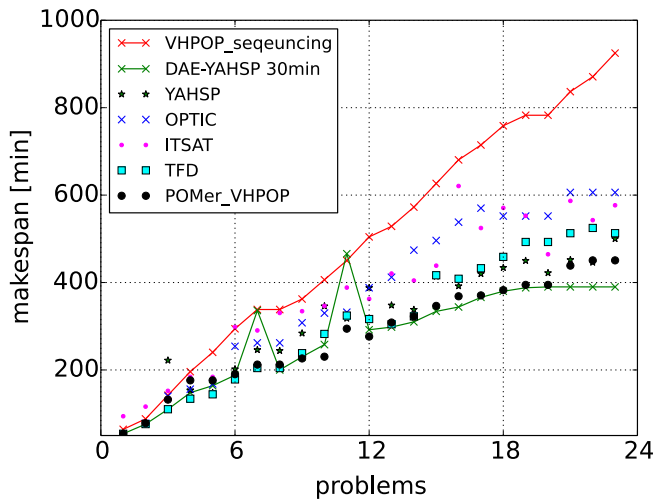


Figure 5. Makespans for problems in Drivelog domain.

6.5 TMS domain

Even though, planners DAE$_{YAHSP}$, YAHSP and TFD perform very well in Drivelog domain, they are unable to find valid solutions in TMS domain as they do not handle concurrent actions correctly. As result, we comparing POMer$_{VHPOP}$ with only OPTIC, ITSAT and the original VHPOP. However an interesting phenomena occurs for our problems: all planners find almost the same makespans. This phenomena occurs due to a fact that a kiln used for baking ceramic has no resource limits. Thus all the ceramic pieces can be baked in parallel.

7 CONCLUSION

We presented an approach for merging of partial order plans especially suited for mobile service robots that need to execute tasks at different locations in an environment. The approach is based on first solving relaxed problems for each individual task, and then perform search over the solutions for these relaxed problems, stitching them together in a way that takes advantage of the synergies between the different tasks. We provided an evaluation of our approach on two benchmarking domains, showing that, for the class of problems we are interested in, it is competitive with state-of-the-art temporal planners. Furthermore, it illustrated our approaches flexibility, as it can perform well in the two domains we analysed, while the other approaches have issues in at least one of the domains.

Future work includes developing an automatic relaxation of the individual problems, and tackling issues related to the execution of the plans we are generating in a mobile robot. This includes closing the loop between plan generation and execution, for which we feel partial order plans are better suited than totally ordered ones, and tackle other common issues for service robotics, such as timing constraints on task execution, the uncertainty inherent to execution in the real world, or merging of plans for new tasks arriving during execution.

REFERENCES

[1] J. Barreiro, M. Boyce, M. Do, J. Frank, M. Iatauro, T. Kichkaylo, E. Remolina P. Morris, J. Ong, T. Smith, and D. Smith, 'Europa: A platform for ai planning, scheduling, constraint programming, and optimization',

in *Proc. of 4th International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS)*.

[2] J. Benton, Amanda Jane Coles, and Andrew Coles, 'Temporal planning with preferences and time-dependent continuous costs', in *International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI, (2012).

[3] Sara Bernardini and David E. Smith, 'Developing domain-independent search control for europa2', in *ICAPS-07 Workshop on Heuristics for Domain-independent Planning*, (2007).

[4] Jacques Bibaï, Pierre Savéant, Marc Schoenauer, and Vincent Vidal, 'An evolutionary metaheuristic based on state decomposition for domain-independent satisficing planning', in *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS-2010)*, pp. 18–25, Toronto, ON, Canada, (May 2010). AAAI Press.

[5] K. E. C. Booth, T. T. Tran, G. Nejat, and J. C. Beck, 'Mixed-integer and constraint programming techniques for mobile robot task planning', *IEEE Robotics and Automation Letters*, **1**(1), 500–507, (Jan 2016).

[6] Michael Cashmore, Maria Fox, Derek Long, Daniele Magazzeni, Bram Ridder, Arnau Carrera, Narcís Palomeras, Natàlia Hurtós, and Marc Carreras, 'Rosplan: Planning in the robot operating system', in *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015.*, pp. 333–341, (2015).

[7] A. J. Coles, A. I. Coles, M. Fox, and D. Long, 'Forward-chaining partial-order planning', in *The International Conference on Automated Planning and Scheduling (ICAPS-10)*, (May 2010).

[8] Brian Coltin and Manuela M. Veloso, 'Online pickup and delivery planning with transfers for mobile robots', in *Proc. of 2014 IEEE Int. Conf. on Robotics and Automation (ICRA)*, (2014).

[9] Brian Coltin, Manuela M. Veloso, and Rodrigo Ventura, 'Dynamic user task scheduling for mobile robots.', in *Proc. of 2011 AAAI Workshop on Automated Action Planning for Autonomous Mobile Robots*, (2011).

[10] 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).

[11] David E. Foulser, Ming Li, and Qiang Yang, 'Theory and algorithms for plan merging', *Artificial Intelligence*, **57**(23), 143 – 181, (1992).

[12] C. Harris and R. Dearden, 'Contingency planning for long-duration auv missions', in *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pp. 1–6, (Sept 2012).

[13] Malte Helmert, 'The fast downward planning system', *Journal of Artificial Intelligence Research*, **26**, 191–246, (2006).

[14] Subbarao Kambhampati, 'Refinement planning as a unifying framework for plan synthesis', *Artificial Intelligence Magazine*, **18**(2), 67–97, (1997).

[15] Subbarao Kambhampati, Craig A. Knoblock, and Qiang Yang, 'Planning as refinement search: A unified framework for evaluating design tradeoffs in partial-order planning', *Artificial Intelligence*, **76**, 167–238, (1995).

[16] Wing-Yue Geoffrey Louie, Tiago Stegun Vaquero, Goldie Nejat, and J. Christopher Beck, 'An autonomous assistive robot for planning, scheduling and facilitating multi-user activities', in *Proc. of 2014 IEEE Int. Conf. on Robotics and Automation (ICRA)*, (2014).

[17] Conor McGann, Frederic Py, Kanna Rajan, Hans Thomas, Richard Henthorn, and Rob McEwen, 'T-rex: A model-based architecture for auv control', *3rd Workshop on Planning and Plan Execution for Real-World Systems*, **2007**, (2007).

[18] Lenka Mudrova, Bruno Lacerda, and Nick Hawes, 'An integrated control framework for long-term autonomy in mobile service robots', in *2015 European Conference on Mobile Robots (ECMR)*. IEEE, (2015).

[19] Bernhard Nebel and Jana Koehler, 'Plan reuse versus plan generation: A theoretical and empirical analysis', *Artificial Intelligence*, **76**, 427–454, (1995).

[20] Filip Dvořák, Arthur Bit-Monnot, Félix Ingrand, and Malik Ghallab, 'A flexible anml actor and planner in robotics', in *ICAPS PlanRob WorkShop*, Portsmouth, USA, (June 2014).

[21] Masood Feyzbakhsh Rankooh and Gholamreza Ghassem-Sani, 'New encoding methods for sat-based temporal planning', 110–117, (2013).

[22] Maurizio Di Rocco, Federico Pecora, and Alessandro Saffiotti, 'When robots are late: Configuration planning for multiple robots with dynamic goals.', in *Proc. of 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, (2013).

[23] Earl D. Sacerdoti, 'The nonlinear nature of plans', in *Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'75, pp. 206–214, (1975).

[24] Reid G. Simmons and Håkan L. S. Younes, 'VHPOP: versatile heuristic partial order planner', *CoRR*, **abs/1106.4868**, (2011).

[25] D. E. Smith, J. Frank, and W. Cushing, 'The anml language', in *ICAPS-08 Workshop on Knowledge Engineering for Planning and Scheduling (KEPS)*, (2008).

[26] A. Tate, 'Generating project networks', in *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, IJCAI'77, pp. 888–893, (1977).

[27] Ioannis Tsamardinos, Martha E. Pollack, and John F. Horty, 'Merging plans with quantitative temporal constraints, temporally extended actions, and conditional branches', in *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems, Breckenridge, CO, USA, April 14-17, 2000*, pp. 264–272, (2000).

[28] Roman Van Der Krogt and Mathijs De Weerdt, 'Plan repair as an extension of planning', in *In Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS-05*, pp. 161–170, (2005).

[29] Manuela M. Veloso, Joydeep Biswas, Brian Coltin, Stephanie Rosenthal, Thomas Kollar, Cetin Mericli, Mehdi Samadi, Susana Brandao, and Rodrigo Ventura, 'Cobots: Collaborative robots servicing multi-floor buildings.', in *Proc. of 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, (2012).

[30] Vincent Vidal, 'YAHSP2: Keep it simple, stupid', in *Proceedings of the 7th International Planning Competition (IPC-2011)*, pp. 83–90, Freiburg, Germany, (June 2011).

[31] Vincent Vidal, Lucas Bordeaux, and Youssef Hamadi, 'Adaptive K-parallel best-first search: A simple but efficient algorithm for multi-core domain-independent planning', in *Proceedings of the 3rd Symposium on Combinatorial Search (SOCS-2010)*, pp. 100–107, Stone Mountain, GA, USA, (July 2010). AAAI Press.

[32] Daniel S. Weld, 'An introduction to least commitment planning', *AI magazine*, **15**(4), 27, (1994).

[33] Qiang Yang, *Intelligent Planning: A Decomposition and Abstraction Based Approach*, Springer-Verlag, London, UK, 1997.

This page intentionally left blank

ECAI Short Papers

This page intentionally left blank

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1549

Towards Online Concept Drift Detection with Feature Selection for Data Stream Classification

Mahmood Hammoodi [1] and **Frederic Stahl** [2] and **Mark Tennant** [3]

Abstract. Data Streams are unbounded, sequential data instances that are generated very rapidly. The storage, querying and mining of such rapid flows of data is computationally very challenging. Data Stream Mining (DSM) is concerned with the mining of such data streams in real-time using techniques that require only one pass through the data. DSM techniques need to be adaptive to reflect changes of the pattern encoded in the stream (concept drift). The relevance of features for a DSM classification task may change due to concept drifts and this paper describes the first step towards a concept drift detection method with online feature tracking capabilities.

1 INTRODUCTION

Velocity in Big Data Analytics [6] refers to data that is generated at a high speed in real-time and challenges our computational capabilities in terms of storing and processing the data [2]. DSM requires techniques that are incremental, computationally efficient and can adapt to *concept drift* for applications such as real-time analytics of chemical plant data in the chemical process industry [10], intrusion detection in telecommunications [9], etc. A concept drift occurs if the pattern encoded in the data stream changes. DSM has developed various real-time versions of established predictive data mining algorithms that adapt to concept drift and keep the model accurate over time, such as CVFDT [8] and G-eRules [11]. The benefit of classifier independent concept drift detection methods is that it allows providing information about the dynamics of the data generation. Common drift detection methods are for example ADaptive sliding WINdow (ADWIN) [4], Drift Detection Method (DDM) by [7] and the Early Drift Detection Method (EDDM) by [3]. However, to the best of our knowledge, no drift detection method provides insights into which features are involved in the concept drift, which is potentially valuable information. For example, if a feature is contributing to a concept drift it can be assumed that the feature may have become either more or less relevant for the concept encoded in the stream after the drift. This knowledge about a feature's contribution to concept drift could be used to develop an efficient real-time feature selection method that does not require examining the entire feature space for online feature selection. This paper proposes a concept drift detection method for data stream classification that also feeds forward information about the involvement of individual features in the drift for feature selection purposes. The proposed method could be used with any learning algorithms either as a real-time wrapper for a batch classifier or realised inside a real-time adaptive classifier. This paper is organised as follows: Section 2 introduces the proposed concept drift and feature selection method, Section 3 evaluates the methodology briefly as a proof of concept and Section 4 provides concluding remarks.

[1] University of Reading, United Kingdom, email: m.s.h.hammoodi@pgr.reading.ac.uk
[2] University of Reading, United Kingdom, email: F.T.Stahl@reading.ac.uk
[3] University of Reading, United Kingdom, email: m.tennant@pgr.reading.ac.uk

2 REAL-TIME FEATURE SELECTION USING USING ADAPTIVE MICRO-CLUSTERS

The work presented in this paper is based on the Micro-Cluster structure of the MC-NN classifier [12] developed by one of the authors of this paper. MC-NN Micro-Clusters are an extension of the Micro-Clusters used in the CluStream data stream clustering algorithm [1]. The notation used for a Micro-Cluster has been taken from [1]. Essentially Micro-Clusters in MC-NN aim to keep a recent accurate summary of the data stream. The structure of MC-NN Micro-Clusters is: $< CF2^x, CF1^x, CF1^t, n, CL, \epsilon, \Theta, \alpha, \Omega >$
$CF2^x$ is a vector with the sum of squares of the features; $CF1^x$ a vector with the sum of feature values; $CF1^t$ a vector with the sum of time stamps; n is the number of data instances in the cluster; CL is the cluster's majority class label; ϵ the error count; Θ the error threshold (default 5000) for splitting the Micro-Cluster; α is the initial time stamp and Ω a threshold for the Micro-Cluster's performance (default 50). The centroid of a Micro-Cluster is calculated by $\frac{CF1^x}{n}$.

Loosely speaking MC-NN updates Micro-Clusters by adding a new instance to its nearest Micro-Cluster if it matches the CL it decrements the error ϵ by 1. Otherwise it adds the data instances to its nearest Micro-Cluster that matches the CL but increases the error count ϵ of both involved Micro-Clusters by 1. If a Micro-Cluster's error count reaches Θ it splits into to new Micro-Clusters placed about the original Micro-Cluster's feature of greatest variance and the original Micro-Cluster is removed in order to fit the data stream better. The variance for a feature x can be calculated by $Variance[x] = \sqrt{\left(\frac{CF2^x}{n}\right) - \left(\frac{CF1^x}{n}\right)^2}$, the assumption is that the larger the variance, the greater the range of values that have been seen for this feature and thus is may contribute to misclassification. New Micro-Clusters are generated with the old Micro-Clusters' centroid values. The centroids values of the 2 new Micro-Clusters, for the attribute that has the largest variance is 'altered' by either adding or subtracting the variance amount (adding in one Micro-Cluster, subtracting in the other). The participation of the cluster on absorbing instances is monitored over time and if a cluster has not participated recently in classifications it is removed. The clusters' participation is measured with the *Triangle Number* $\Delta(T) = ((T^2 + T)/2)$ which can be calculated from $CF1^t$. The lower $CF1^t$ the lower the participation of the Micro-Cluster, but the triangle number gives more importance to recent instances than older ones. In order to detect a concept drift, we track the total number of Micro-Cluster splits and

removals. The assumption is that the larger either or both numbers are, the more likely it is that a concept drift happened.

Using a windowing approach upon the data stream, a running average of the split and death rates are calculated concurrently. If the percentage of the split and death rates differs from the mean (of the statistical windows) by 50% (default value), this is consider as a concept drift. Then a closer look can be taken into the individual features through examining their change in velocity. This is tracked through an extension of MC-NN's Micro-Cluster structure by: $< CF1^{hx}, n_h >$. Where the components of the structure above are equivalent to $CF1^x$ and n. However, the h denotes that these components are *historical* summaries (taken from the statistical windows); the value of h is a fixed user defined parameter and denotes how many time stamps the historical summaries are behind the recent ones (default 10,000). The *velocity* of a feature x can then be calculated by $V_x = \frac{CF1^x}{n} - \frac{CF1^{hx}}{n_h}$. A high velocity during a concept drift indicates that the feature changed. The assumption here is that this particular feature may have changed its contribution towards the classification technique, whereas the remaining ones have not. Thus feature selection can be limited to examining only features that have changed their velocity where there is a concept drift detected. Section 3 evaluates this approach as a principal proof of concept.

3 EXPERIMENTAL EVALUATION

This section aims to show that the proposed methodology can identify concept drift in a data stream and at the same time detect which features were involved. Random Tree data stream generator, which was introduced in [5] and generates a stream based on a randomly generated tree, was used for a proof of concept. New examples are generated by assigning uniformly distributed random values to features, which then determine the class label using the randomly generated tree.

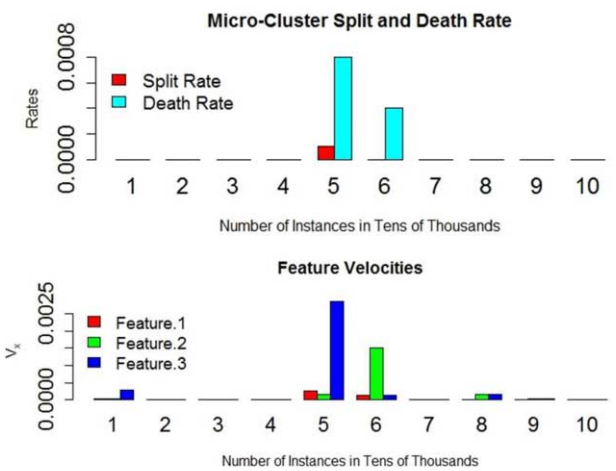


Figure 1. The top of the figure shows the Micro-Cluster split and death rate and the bottom of the figure shows the feature velocities.

We generated a stream of 100,000 instances, 3 features and 2 classification labels. Features 1 and 2 are relevant to determine the class label and feature 3 is random. After 50,000 instances features 2 and 3 were swapped making feature 2 irrelevant and feature 3 relevant for classification tasks. We noticed that the method accurately detected a drift at 50,000 instances as Micro-Clusters were reset. For the experiment the default parameters stated in Section 2 of the method were used unless stated otherwise. In the top half of Figure 1 it can be seen that the split and death rates at the time of concept drift increase, indicating that the current set of Micro-Clusters does not fit the concept encoded in the data anymore. The bottom of Figure 1 shows the velocities of the features in the Micro-Clusters. In this case we know that the concept appeared due to swapping features 2 and 3, hence we would expect a higher velocity of these two features. Figure 1 shows this change in velocity. Thus the method is capable to detect a concept drift but also delivers an indication which features are involved, which can be used to perform online real-time feature selection.

4 CONCLUSIONS

This paper introduced a novel Micro-Cluster based methodology for drift detection in data streams. Different compared with existing drift detection techniques, the proposed method is also capable to detect which features have been involved in the drift through the velocity of Micro-Clusters in different dimensions; and thus can be used to implement real-time feature selection techniques. The experimental proof of concept shows that the methods can successfully detected concept drifts and identify drifting features. Ongoing and future work comprises an in depth evaluation of the method and the development of a real-time feature selection technique.

REFERENCES

[1] C. Aggarwal, J. Han, J. Wang, and P.Yu, 'A framework for clustering evolving data streams', in *Proceedings of the 29th VLDB Conference*, Berlin Germany, (2003).

[2] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom, 'Models and issues in data stream systems', in *In PODS*, pp. 1–16, (2002).

[3] Manuel Baena-Garcıa, José del Campo-Ávila, Raúl Fidalgo, Albert Bifet, R Gavalda, and R Morales-Bueno, 'Early drift detection method', in *Fourth international workshop on knowledge discovery from data streams*, volume 6, pp. 77–86, (2006).

[4] Albert Bifet and Ricard Gavald, 'Learning from time-changing data with adaptive windowing', in *In SIAM International Conference on Data Mining*, pp. 443–448.

[5] Pedro Domingos and Geoff Hulten, 'Mining high-speed data streams', *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '00*, 71–80, (2000).

[6] M Ebbers, A Abdel-Gayed, V Budhi, and F Dolot, *Addressing Data Volume, Velocity, and Variety with IBM InfoSphere Streams V3.0*, IBM Redbooks, 2013.

[7] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues, 'Learning with drift detection', in *Advances in artificial intelligence–SBIA 2004*, 286–295, Springer, (2004).

[8] Geoff Hulten, Laurie Spencer, and Pedro Domingos, 'Mining Time-Changing Data Streams', in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, volume 18, pp. 97–106. ACM, (2001).

[9] A. Jadhav, A. Jadhav, P. Jadhav, and P. Kulkarni, 'A novel approach for the design of network intrusion detection system(NIDS)', in *Sensor Network Security Technology and Privacy Communication System (SNS PCS), 2013 International Conference on*, pp. 22–27, (May 2013).

[10] Petr Kadlec, Bogdan Gabrys, and Sibylle Strandt, 'Data-driven Soft Sensors in the process industry', *Computers and Chemical Engineering*, **33**(4), 795–814, (2009).

[11] Thien Le, Frederic Stahl, João Bártolo Gomes, Mohamed Medhat Gaber, and Giuseppe Di Fatta, 'Computationally Efficient Rule-Based Classification for Continuous Streaming Data', in *Research and Development in Intelligent Systems XXIV*, p. 2014, (2008).

[12] Mark Tennant, Frederic Stahl, and João Bártolo Gomes, 'Fast adaptive real-time classification for data streams with concept drift', in *Internet and Distributed Computing Systems*, 265–272, Springer, (2015).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1551

Towards SPARQL-Based Induction for Large-Scale RDF Data Sets

Simon Bin[1] and **Lorenz Bühmann**[1] and **Jens Lehmann**[2] and **Axel-Cyrille Ngonga Ngomo**[1]

Abstract. We show how to convert OWL Class Expressions to SPARQL queries where the instances of that concept are with a specific ABox equal to the SPARQL query result. Furthermore, we implement and integrate our converter into the CELOE algorithm (Class Expression Learning for Ontology Engineering), where it replaces the position of a traditional OWL reasoner. This will foster the application of structured machine learning to the Semantic Web, since most data is readily available in triple stores. We provide experimental evidence for the usefulness of the bridge. In particular, we show that we can improve the run time of machine learning approaches by several orders of magnitude.

1 INTRODUCTION AND MOTIVATION

A growing amount of data from diverse domains is being converted into RDF[3] as demonstrated by the growth of the Linking Open Data Cloud.[4] With this conversion come a significant number of complex applications which rely on large amounts of data in RDF and OWL[5] to perform demanding tasks, such as detecting patients with particular diseases [11]. While OWL reasoners can provide the required information for structured machine learning, they do not scale to large data sets. On the other hand, the SPARQL query language was developed specifically to query large amounts of data. By creating a bridge between SPARQL and OWL, we are able to answer OWL queries on large amounts of data.

Description Logics is the name of a family of knowledge representation (KR) formalisms. They emerged from earlier KR formalisms like semantic networks and frames. Their origin lies in the work of Brachman on structured inheritance networks [3]. $\mathcal{SROIQ}$ is a well-known description language, as it is the basis for OWL 2. $\mathcal{SROQ}$ is the subset of that language lacking inverse properties. We refer to [6] for details. For a complete definition of the SPARQL syntax and semantics, we refer to [1, 10] and the official W3C recommendation.[6]

2 OWL CLASS EXPRESSION REWRITING ALGORITHM

Proposition. Given an ABox $\mathcal{A}$, which contains class assertions to named classes and role assertions, we define $\mathcal{I}$ as the canonical interpretation [2]. Then we can show that executing the query converted from a concept C using τ as given in Tables 1 and 2 over $\mathcal{A}$ is the same as the canonical interpretation of C. Due to space constraints, please refer to our technical report for the proof.[7]

Table 1. Conversion of class expressions into a SPARQL graph pattern.

Class Expression C_i	Graph Pattern $\mathfrak{p} = \tau(C_i, ?\text{var})$
A	`{?var rdf:type A.}`
$\neg C$	`{?var ?p ?o .` `FILTER NOT EXISTS {` $\tau(C, ?\text{var})$`}}`
$\{a_1, \ldots, a_n\}$	`{?var ?p ?o .` `FILTER (?var IN (`$a_1, \ldots, a_n$`))}`
$C_1 \sqcap \ldots \sqcap C_n$	$\{\tau(C_1, ?\text{var}) \cup \ldots \cup \tau(C_n, ?\text{var})\}$
$C_1 \sqcup \ldots \sqcup C_n$	$\{\tau(C_1, ?\text{var})\}$ `UNION ... UNION` $\{\tau(C_n, ?\text{var})\}$
$\exists\, r.C$	`{?var r ?s.}` $\cup\, \tau(C, ?s)$
$\exists\, r.\{a\}$	`{?var r a.}`
$\exists\, r.SELF$	`{?var r ?var.}`
$\forall\, r.C$	`{ ?var r ?s0.` `{ SELECT ?var` `(count(?s1) AS ?cnt1)` `WHERE { ?var r ?s1 .` $\tau(C, ?s1)$ `}` `GROUP BY ?var }` `{ SELECT ?var` `(count(?s2) AS ?cnt2)` `WHERE { ?var r ?s2 }` `GROUP BY ?var }` `FILTER ( ?cnt1 = ?cnt2 ) }`
$\Theta n\, r.C$ $\Theta \in \{\leq, \geq, =\}$	`{ ?var r ?s0.` `{ SELECT ?var` `WHERE { ?var r ?s .` $\tau(C, ?s)$ `}` `GROUP BY ?var` `HAVING ( count(?s)` $\Theta\, n$ `) } }`

Table 2. Conversion of property expressions into a SPARQL g.p.

Property Expression p_i	Graph Pattern $\mathfrak{p} = \tau(p_i, ?\text{var})$
p	`{?var p ?o.}`
p^{-1}	`{?s p ?var .}`
$p_1 \circ \cdots \circ p_n$	`{?var` p_1 `?o1.` ⋮ ⋮ ⋮ `?o_n-1` p_n `?o_n.}`

3 LEARNING PROBLEM AND ALGORITHM

We consider supervised machine learning from positive and negative examples. All our experiments are binary classification tasks. The *CELOE* algorithm (Class Expression Learning for Ontology Engineering) iteratively generates class expressions and evaluates their performance on the positive and negative examples [8]. CELOE relies

[1] Universität Leipzig, Germany, email: {sbin,buehmann,ngonga}@informatik.uni-leipzig.de
[2] University of Bonn and Fraunhofer IAIS, Germany, email: jens.lehmann@cs.uni-bonn.de and jens.lehmann@iais.fraunhofer.de

[3] http://www.w3.org/TR/rdf11-concepts/
[4] http://lod-cloud.net/
[5] http://www.w3.org/TR/owl2-overview/
[6] http://www.w3.org/TR/sparql11-query/
[7] http://svn.aksw.org/papers/2016/ECAI_SPARQL_Learner/tr_public.pdf

Table 3. Data set characteristics.

number of	triples	classes	data/object properties		expressivity
carcinogenesis	74,567	142	15	4	$\mathcal{ALC}$(D)
mutagenesis	62,067	86	6	5	$\mathcal{AL}$(D)
mammograms	6,809	19	2	3	$\mathcal{AL}$(D)
TCGA-A	35,329,868	24	113	48	$\mathcal{AL}$(D)

on a top-down algorithm based on refinement operators. We use the refinement operator defined in [9].

4 EVALUATION

We implemented the SPARQL querying method according to Section 2 as an extension to the DL-Learner [7] framework in the place of an OWL API reasoner.[8] We use four data sets to compare the SPARQL induction to OWL reasoners. The *Carcinogenesis* and *Mutagenesis* data sets are moderately-sized data sets converted from data provided by the Oxford University Machine Learning group.[9] The mutagenesis data set is based on the results of [4]. The *Mammographic Mass data set* (Mammograms) was published in [5].[10] Additionally, we evaluated our approach on an excerpt from the cancer patient data in LinkedTCGA[11] (35 million triples). This data set has not previously been possible to use with DL-Learner due to its size.

In Table 3, the main characteristics of the data sets are described. The number of RDF triples describes the total size of the data set. Furthermore, the total number of classes and data as well as object properties (across all classes) is indicated. The expressivity refers to the description logic language features that are used in the data set as customary in description logics.

The approaches are evaluated using three different access options inside the DL-Learner framework. We tested the popular HermiT,[12] Pellet[12] and FaCT[13] OWL reasoners. For SPARQL, the data was loaded into an in-memory Jena[13] model with OWL/Lite inference rules[14] as well as a pre-computed model, in which case the SPARQL back-end acts as a pure graph database. In our evaluation setup, 22 seconds (mutagenesis), 36 seconds (carcinogenesis), 7 seconds (mammograms) were spent on pre-computing all inferences externally beforehand. We loaded the LinkedTCGA data set[15] into a SPARQL endpoint running OpenLink Virtuoso[16] 7.1. All experiments were run on an AMD Opteron 6376 @ 2.3GHz with 256 GB system memory, of which the DL-Learner framework used 32 GB. The algorithm itself is single-threaded.

The configuration files to reproduce our experiment can be found in the DL-Learner repository.[17]

5 CONCLUSION

The SPARQL approaches were nearly two orders of magnitude faster as can be seen in Tables 4 and 5. However, for the moment we have excluded the pressing lack of inference support for larger data sets.

[8] http://owlapi.sourceforge.net/reasoners.html
[9] http://www.cs.ox.ac.uk/activities/machlearn/applications.html
[10] http://archive.ics.uci.edu/ml/datasets/Mammographic+Mass
[11] http://aksw.org/Projects/LinkedTCGA
[12] http://www.hermit-reasoner.com/
[13] https://jena.apache.org/
[14] https://jena.apache.org/documentation/inference/
[15] https://code.google.com/p/bigrdfbench/
[16] http://virtuoso.openlinksw.com/
[17] https://github.com/AKSW/DL-Learner/tree/sparql-comparison/test/sparql-comparison

Table 4. Number of concept tests within 4000 seconds (mean average in ten runs).

	SPARQL Precomp.	SPARQL Micro-rule	HermiT	Pellet
carcinogenesis	162,430	60,527	87	93
mutagenesis	11,713	4,552	timeout	176
mammograms	26,036	12,260	28	173

Table 5. Run time (seconds) until top accuracy.

	Precomp.	Micro-rule	HermiT	Pellet	FaCT
carcinogenesis	95	1,380	timeout	timeout	713
mutagenesis	1	3	timeout	159	-
mammograms	286	647	494	365	-
TCGA-A	1,243	timeout	timeout	timeout	-

ACKNOWLEDGEMENTS

This work was supported in part by a research grant from the German Ministry for Finances and Energy under the SAKE project (Grant No. 01MD15006E) and by a grants from the EU's 7th Framework Programme provided for the projects GeoKnow (GA no. 318159), Big Data Europe (GA no. 644564) and HOBBIT (GA no. 688227).

REFERENCES

[1] M. Arenas, C. Gutierrez, and J. Pérez, 'On the semantics of SPARQL', in *Semantic Web Information Management*, 281–307, Springer, (2010).
[2] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, eds. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
[3] R.J. Brachman, 'A structural paradigm for representing knowledge', Technical Report BBN Report 3605, Bolt, Beraneck and Newman, Inc., Cambridge, MA, (1978).
[4] A.K. Debnath, R.L. Lopez de Compadre, G. Debnath, A.J. Shusterman, and C. Hansch, 'Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity', *Journal of medicinal chemistry*, **34**(2), 786–797, (1991).
[5] M. Elter, R. Schulz-Wendtland, and T. Wittenberg, 'The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process', *Medical Physics*, **34**(11), 4164–4172, (2007).
[6] I. Horrocks, O. Kutz, and U. Sattler, 'The even more irresistible SROIQ', in *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, eds., P. Doherty, J. Mylopoulos, and C.A. Welty, pp. 57–67. AAAI Press, (2006).
[7] J. Lehmann, 'DL-Learner: learning concepts in description logics', *Journal of Machine Learning Research (JMLR)*, **10**, 2639–2642, (2009).
[8] J. Lehmann, S. Auer, L. Bühmann, and S. Tramp, 'Class expression learning for ontology engineering', *Journal of Web Semantics*, **9**, 71 – 81, (2011).
[9] J. Lehmann and P. Hitzler, 'Concept learning in description logics using refinement operators', *Machine Learning journal*, **78**(1-2), 203–250, (2010).
[10] J. Pérez, M. Arenas, and C. Gutierrez, 'Semantics and complexity of SPARQL', *ACM Transactions on Database Systems (TODS)*, **34**(3), 16, (2009).
[11] M. Saleem, S.S. Padmanabhuni, A. Ngonga Ngomo, J.S. Almeida, S. Decker, and H.F. Deus, 'Linked cancer genome atlas database', in *Proceedings of I-Semantics2013*, (2013).
[12] E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz. Abstract Pellet: A practical OWL-DL reasoner, 2008.
[13] D. Tsarkov and I. Horrocks, 'FaCT++ description logic reasoner: System description', in *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pp. 292–297. Springer, (2006).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1553

Burg Matrix Divergence Based Multi-Metric Learning

Yan Wang and **Han-Xiong Li** [1]

Abstract. The basic idea of most distance metric learning methods is to find a space that can optimally classify data points belong to different categories. However, current methods only learn one Mahalanobis distance for each data set, which actually fails to perfectly classify different categories in most real world applications. To improve the classification accuracy of k-nearest-neighbour algorithm, a multi-metric learning method is proposed in this paper to completely classify different categories by sequentially learning sub-metrics. The proposed algorithm is based on minimizing the Burg matrix divergence between metrics. The experiments on five UCI data sets demonstrate the improved performance of Multi-Metric learning when comparing with the state-of-the-art methods.

1 Introduction

Learning a good distance metric in feature space is crucial in many learning algorithms, such as nearest neighbors classifier and K-means clustering [2]. Over the past decade, a large number of distance metric learning (DML) algorithms have been proposed to learn a Mahalanobis distance in feature space, and some of them have been successfully applied to real world applications. In order to learn a distance metric that can well classify the dis-similar data pairs, an earlier work [2] uses a semi-definite programming formulation under similarity and dissimilarity constraints. In [5], the Large Margin Nearest Neighbor (LMNN) is suggested to learn a Mahalanobis distance metric for kNN Classification.

Current DML methods are actually learning one metric space that properly classify different categories. Unfortunately, due to the complexity and uncertainty, a linear space that can perfectly classify different categories may not exist. In order to remedy the disadvantages, we propose a method to learn multi-metric spaces so that all the training data can be correctly classified in at least one metric space, as shown in figure 1. In section 2, the base-metric learning

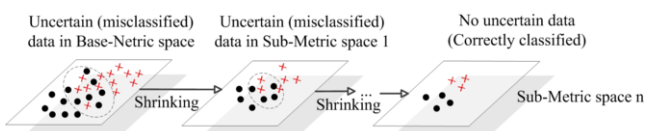


Figure 1. The Sequentially shrinking of Inseparable Set

problem and sub-metric learning problem will be defined. In section 3, the optimization of these two problems are introduced. In case study, we conduct experiments on five public data sets to demonstrate the effectiveness of the proposed method.

[1] Department of Systems Engineering and Engineering Management, City University of Hong Kong

2 Problems and Definitions

Given a data set x_i, where $x_i \in R^d, i = 1, 2, \ldots, n$, the Mahalanobis distance parameterized by positive semi-definite (PSD) matrix A is expressed as:

$$d_A(x_i, x_j) = \sqrt{(x_i - x_j)^T A^{-1}(x_i - x_j)} \tag{1}$$

The uncertain data in this metric space can be defined as follows.

Definition 1 *Uncertain Data. For data x_i, if*

$$d_A(x_i, \mu_{i,k}) - d_A(x_i, \mu_{i,j}) < \rho \tag{2}$$

then x_i is uncertain in metric space A. In this formula, $\mu_{i,j}$ is the class center with the same label to x_i, and $\mu_{i,k}$ is the nearest class center with different class label to x_i, ρ is a hyper-parameter, which represents the desired margin between different classes

Definition 2 *Uncertain Set. If x_i is uncertain in all the existing metric spaces, then it will belong to uncertain set U. As shown in figure 1, the goal of proposed method is gradually shrinking the uncertain set to empty: $U \to \emptyset$.*

Definition 3 *Base-Metric Learning. The base-metric will be a Mahalanobis distance parameterized by a PSD matrix A^0. It is a global optimal metric learnt with the following problem:*

$$\begin{aligned} &\min_{A^0} D_\phi(A^0, M) \\ &s.t.1) \sum_i^n (d_{A^0}(x_i, \mu_{i,j}) - u) \le 0 \\ &\quad 2) \sum_i^n (d_{A^0}(x_i, \mu_{i,k}) - l) \ge 0; \end{aligned} \tag{3}$$

where $D_\phi()$ denotes the distance between matrix A^0 and M, M is a baseline matrix (we choose M equal to the covariance matrix of training set in this paper), u and l are upper limit and lower limit for distance, respectively.

Definition 4 *Sub-Metric Learning. The Sub-metrics is a group of Mahalanobis distances parameterized by PSD matrix $A^1, \cdots, A^k$. Under different distance constraints that force the data in uncertain set to be correctly classified, these distance metrics will be learnt with following problem:*

$$\begin{aligned} &\min_{A^{new}} D_\phi(A^{new}, A^0) \\ &s.t. \sum_i^{x^i \in U} (d_{A^{new}}(x_i, \mu_{i,k}) - d_{A^{new}}(x_i, \mu_{i,j})) \ge \rho * N_U \end{aligned} \tag{4}$$

where U denotes the uncertain set defined in definition 2, N_U is the size of U. The details of this problem will be analyzed in section 3.

3 Optimization

3.1 Measure of Similarity Between Metrics

In problem (3) and (4), the objective is minimizing the difference $D_\phi()$ between target matrix A^0 and original matrix M. In this paper, The Burg matrix divergence is adopted to quantify this difference [4], which defines the $D_\phi()$ as:

$$\begin{aligned} D_\phi(A, M) &= KL(p(x, A)||p(x, M)) \\ &= tr(AM^{-1}) - \log|AM^{-1}| - d \end{aligned} \tag{5}$$

3.2 Base-Metric Learning

With formula (5), we can rewrite the base-metric learning problem in (3) as a Burg matrix optimization process:

$$\begin{aligned} &\min_{A^0} tr(A^0 M^{-1}) - \log|A^0 M^{-1}| - d \\ &s.t.1) tr((A^0)^{-1} \sum_i^n (x_i - \mu_{i,j})(x_i - \mu_{i,j})^T) \leq n * u \\ &2) tr((A^0)^{-1} \sum_i^n (x_i - \mu_{i,k})(x_i - \mu_{i,k})^T) \geq n * l; \end{aligned} \tag{6}$$

In above formulation, since the constraints here only have demands on the averaged distance to class centers, they are weaker than the commonly used pairwise constraints [4] or triplet constraints [3]. A colesd-form solution for this problem is derived in [1].

3.3 Sub-metric learning

The weaker constraints adopted in problem (6) may not be able to ensure a prefect metric that can found linear boundaries between different categories. To remedy the disadvantage of base-metric learning, we propose sub-metric learning problem shown in (4). The task of sub-metric learning is shrinking the uncertain set U so that each data instance can be correctly classified in at least one metric. Following formula (5), the sub-metric learning can be rewritten as:

$$\begin{aligned} &\min_{A^{new}} tr(A^{new} A^{0^{-1}}) - \log|A^{new} A^{0^{-1}}| - d \\ &s.t. \quad tr((A^{new})^{-1} \sum_i^{x_i \in U} ((x_i - \mu_{i,k})(x_i - \mu_{i,k})^T) \\ &\quad - (x_i - \mu_{i,j})(x_i - \mu_{i,j})^T)) \leq -\rho * N_U; \end{aligned} \tag{7}$$

where U denotes the uncertain set (defined in section 2), N_U is the size of U. A closed-form solution for this problem is proposed in [3], which is much more efficient than other common DML methods.

3.4 Classification

For input x_i, its confidence weight in metric space A^m is defined as:

$$\omega_i = -\log \frac{d_{A^m}(x_i, \mu_{1st})}{d_{A^m}(x_i, \mu_{2nd})} \tag{8}$$

where μ_{1st} denotes the nearest class center to x_i, and μ_{2nd} denotes the second nearest class center to x_i. Then, the classification result will be the class label with the maximum weight. For example, in $0-1$ classification, the probability of $y = 1$ will be:

$$p(y = 1|x, A^0, \ldots, A^m) = \frac{\sum_{i=0}^m \omega_i f_{A_i}(x*)}{\sum_{i=0}^m \omega_i} \tag{9}$$

4 Experiments

In this section we compare the proposed Multi-ML, method with a few methods: Euclidean distance, Mahalanobis distance, lda, ITML [4] and LMNN [5]. Experiments were run on 5 UCI data sets, that are: 1)Pima Indian Diabetes, 2)Breast Cancer Wisconsin Diagnostic, 3)Heart, 4)Liver Disorders and 5)Robot execution failures . All experimental results are obtained by averaging 50 runs. For each run, we randomly split the data sets 70% for training and 30% for testing.

Table 1. KNN (k=1) average classification accuracy of 50 random experiments via different metrics

	Diabetes	WDBC	Heart	Liver	Failures
Multi-ML	**0.693**	**0. 936**	**0.758**	0.607	**0.879**
LMNN	0.680	0.916	0.660	0.588	0.853
LDA	0.678	0.927	0.755	0.567	0.882
ITML	0.681	0.912	0.728	**0.608**	0.850
Euclidean	0.680	0.915	0.590	0.607	0.798
Mahalanobis	0.671	0.895	0.581	0.551	0.785

As we can see from Table 1, the proposed Multi-ML method outperforms other state-of-the-art methods on 4 of the 5 date sets. To understand the complexity of proposed method, the average computation time of different algorithms are listed in Table 2. We can find that Multi-ML is much faster than LMNN and ITML.

Table 2. Computation time (s)

	Diabetes	WDBC	Heart	Liver	Failures
LMNN	3.65	1.55	0.33	0.70	7.15
ITML	5.34	6.16	6.37	4.79	27.82
Multi-ML	0.10	0.17	0.062	0.071	1.01

5 Conclusion

Instead of current single-metric learning method, a multi-ML is proposed in this paper to improve the accuracy classification. By proposing base-ML and sub-ML problem as Burg matrix optimization problem, the proposed model enables us to derive an efficient close-form algorithm. The experiments on five UCI data sets prove the effectiveness of the proposed method.

REFERENCES

[1] M. Sustik B. Kulis and I. Dhillon, 'Learning low-rank kernel matrices', in *Proceedings of the 23rd international conference on Machine learning,(ICML 2007)*, pp. 505–512, (2006).

[2] S. Russell E. P. Xing, M. I. Jordan and A. Y. Ng, 'Distance metric learning with application to clustering with side-information', in *Advances in neural information processing systems (NIPS 2002)*, pp. 505–512, (2002).

[3] H. R. Karimi J. Mei, M. Liu and H. Gao, 'Logdet divergence based metric learning using triplet labels', in *Proceedings of the Workshop on Divergences and Divergence Learning (ICML 2013)*, ed., Springer, (2013).

[4] P. Jain S. Sra J. V. Davis, B. Kulis and I. S. Dhillon, 'Information-theoretic metric learning', in *Proceedings of the 24th international conference on Machine learning, (ICML 2007)*, ed., Springer, pp. 209–216, (2007).

[5] J. Blitzer K. Q. Weinberger and L. K. Saul, 'Distance metric learning for large margin nearest neighbor classification', in *Advances in neural information processing systems (NIPS 2005)*, pp. 1473–1480, (2005).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1555

A Novel Approach of Applying the Differential Evolution to Spatial Discrete Data

Vojtěch Uher and **Petr Gajdoš** and **Michal Radecký** and **Václav Snášel**[1]

Abstract. The Differential Evolution (DE) is a powerful bio-inspired algorithm searching optimal solutions. The actual DE modifications can handle the real, integer and discrete valued problems. The values of the discrete-valued variables represent the integer indices addressing the discrete samples in the ordered array. The optimization in unordered samples leads to a random search. This paper proposes a novel modification dealing with d-dimensional discrete vertices. A vertex hashing is used to strengthen the local properties of a dataset and to improve the spatial convergence of the evolution.

1 Introduction

Several DE modifications dealing with the real, integer and discrete valued problems have been published [6][4][3][1]. The discrete DE variants [1][4] iteratively optimize integer indices addressing the enumerative samples in the memory during g generations. The presented method is based on the DDE by Onwubolu and Davendra [1]. The DDE uses the integer/real value transformation and profits from the robustness of the real-coded DE [6]. The efficiency of the algorithm depends on the order of the data, because it significantly affects the convergence of the evolution [4]. This paper aims at spatial combinational problems, where a set of discrete vertices represents the solution. An analysis of the spacial data is a very common task in computer vision, robotics or pattern recognition. Such a problem can be classified as a discrete-valued problem, so that integer indices addressing the proposed vertices are iteratively optimized. However, the d-dimensional vertices cannot be simply ordered in memory. The vertices are often non-uniformly distributed, thus the linear combination of indices addressing unordered vertices leads to the random selection. This paper proposes a solution using a linearization of the spatial data with the space-filling curves (SFCs) [2][5].

2 Spatial optimization with DDE

The parameters of our modification are similar to the DDE [1]. P is the number of individuals of a population, F is the mutational factor, C is the crossover probability, g is the maximum number of generations, n is the number of individual variables, d is the dimension of the vertices, l is the total number of vertices and $f(X)$ is the objective function $f(X) : R_n \rightarrow R$, where $X = (x_1, \ldots, x_n)$ represents the vertex indices. The DDE defines so-called Forward Backward Transformation of the variable values. The values are transformed from integer to real values (Forward) before the DE strategy starts. The real values are transformed to integers (Backward) for the objective function evaluation. Our algorithm is explained on the DE/best/l/bin variant [1] and it is modified in the initialization, evaluation and mutation phases. The crossover and selection phases are traditional.

2.1 Initialization and evaluation

First, the l vertices are hashed and sorted in the memory according to the selected SFC (see e.g. [2][5]). The SFC makes the d-dimensional data partly sequenced, so that the spatially close vertices are stored in a row in the memory. Next, the DDE schematic is followed [1]. The input parameters are set and the initial population with random values within the bounds $\langle 0, l)$ is generated.

The objective function $f(X)$ evaluates the quality of the found individual. An individual consists of n variables storing the vertex indices. The variable values are regularly transformed from real to integer ones (Backward Transformation) to address the corresponding vertices for the evaluation. In our test case, the point-to-point distance function is tested, thus the distances of n vertices are computed and the total sum of the distances is used as the objective value.

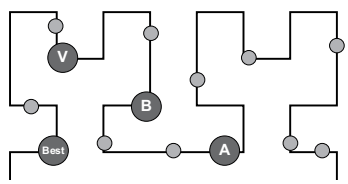


Figure 1. The random vertices are ordered by the Hilbert curve. A new curve index V is computed by the mutation operator from the three individual indices ($best$, A, B).

2.2 Mutation operator

The mutation operator computes a mutant vector as a linear combination of three different individuals: two from the current population and the best-known one (Figure 1). In this case, an individual is a set of vertex indices, which represents the potential solution of a discrete optimization. The integer values are transformed to the real ones before the mutation phase (Forward Transformation). For each index $x^G_{i,j}$ a mutant index $v^G_{i,j} = x^G_{best,j} + F \cdot (x^G_{A,j} - x^G_{B,j})$ is computed, where $i = 1, \ldots, P$, $j = 1, \ldots, n$, G is a generation counter and $i \neq A \neq B$. If $v^G_{i,j}$ is placed out of the interval $\langle 0, l)$, a random index from the interval is selected. An application of this operator to the indices addressing the unordered vertices is wasteful, because they do not provide any information about the course of the dataset. However, the SFCs systematically connect the nearby vertices, so that the vertex order represents the spatial character of the data. The SFCs better ensure that a mutant index $v^G_{i,j}$ addresses

[1] Department of Computer Science and National Supercomputing Center, VŠB-Technical University of Ostrava - Ostrava, Czech Republic, email: {vojtech.uher, petr.gajdos, michal.radecky, vaclav.snasel}@vsb.cz

a vertex that lies closer to the reference vertices with indices $x^G_{best,j}$, $x^G_{A,j}$ and $x^G_{B,j}$ (see Figure 1). The evolution thus better converges to the extremes. The Figure 2 shows a comparison of the heatmaps representing the distribution of mutation indices depending on F for the three different SFCs. It shows that the greater F leads to greater number of mutant indices placed out of the interval $\langle 0, l)$. The parameter F affects the coverage of the area. The smaller F leads to gentler sampling of the area. Thus, the specific F can improve the accuracy of the local search in the dataset. The dark areas represent the places with worse coverage. The wrong mutation indices that do not meet the constraints are replaced by random ones, thus they can additionally cover the dark areas.

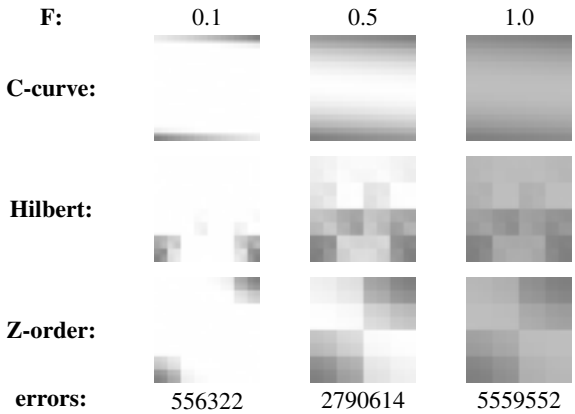


Figure 2. The heatmaps showing the distribution of the mutation indices for different SFCs (C-curve, Hilbert, Z-order) and factor F. The heatmaps were obtained by computation of the all 256^3 existing combinations of the complete level-4 SFC indices. The errors (dependent on F) represent the number of combinations leading to mutant indices placed out of bounds.

3 Experiments

The proposed method was tested on the nearest neighbors (NN) problem and five datasets (see Figure 3) to show the functioning and the improved spatial convergence of the DDE combined with the three selected SFCs (Z-odrer, Hilbert, C-curve). The C-curve was selected as a naive approach for comparison with the more sophisticated SFCs. A vertex $\vec{p}$ is randomly selected from the dataset for each measurement. The DDE algorithm searches the NN of the $\vec{p}$, so that the distance between the proposed vertices and the $\vec{p}$ is minimized. The DDE was tested with the following parameters: $n = 5$, $C = 0.95$, $F = 0.15$, $P = 30$. Stochastic search of the precise n-nearest neighbors is practically impossible. A sufficient result is searched, which is defined as a vertex with lower distance than the best analytically computed solution multiplied by the fitness rate $f_R = 2.0$. This pays for all the n vertices addressed by the individual indices. Figure 3 shows the comparison of the SFCs on the three standard Stanford datasets (buddha, bunny, dragon) and two artificial datasets with 10^6 vertices (uniform and standard normal distribution).

4 Conclusion

This paper introduced a novel variant of the Discrete Differential Evolution (DDE) which searches the optimal solutions in the spatial data. The problem with d-dimensional vertex ordering was solved by the space filling curves (SFCs) to strengthen the local properties of discrete datasets. Figure 3 proves that the DDE combined with the more sophisticated SFCs (Z-order, Hilbert) converges faster to the sufficient result than with the naive C-curve order. Thus, our modified DDE seems to be an efficient and functional method.

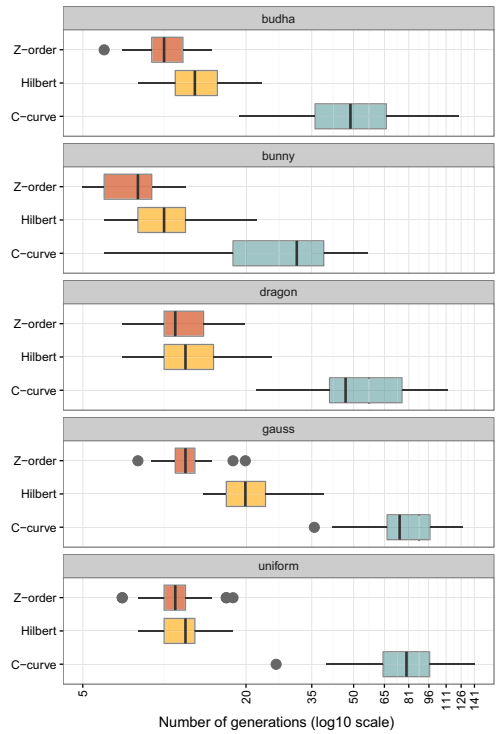


Figure 3. The box plot comparing the SFCs on the point-to-point distance minimization problems in the 3D space. The horizontal axis shows the number of generations needed to reach a sufficient result. Each measurement was done 50 times for the same parameters and datasets.

ACKNOWLEDGEMENTS

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602". This work was supported by SGS project, VSB-Technical University of Ostrava, under the grants no. SP2016/97 and no. SP2016/68.

REFERENCES

[1] Donald Davendra and Godfrey Onwubolu, *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization*, chapter Forward Backward Transformation, 35–80, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[2] Jonathan K. Lawder and Peter J. H. King, 'Using space-filling curves for multi-dimensional indexing', in *Proceedings of the 17th British National Conferenc on Databases: Advances in Databases*, BNCOD 17, pp. 20–35, London, UK, UK, (2000). Springer-Verlag.

[3] Efrñn Mezura-Montes, Jesús Velázquez-Reyes, and Carlos A. Coello Coello, 'A comparative study of differential evolution variants for global optimization', in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pp. 485–492, New York, NY, USA, (2006). ACM.

[4] Godfrey C. Onwubolu and Donald Davendra, *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization*, Springer Publishing Company, Incorporated, 1st edn., 2009.

[5] John Skilling, 'Programming the hilbert curve', *AIP Conference Proceedings*, **707**(1), 381–387, (2004).

[6] Rainer Storn and Kenneth Price, 'Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces', *J. of Global Optimization*, **11**(4), 341–359, (December 1997).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1557

A Stochastic Belief Change Framework with a Stream of Expiring Observations (Short Paper)[1]

Gavin Rens[2]

1 MOTIVATION

I propose a belief maintenance framework which draws from partially observable Markov decision processes (POMDPs) [1] and my hybrid stochastic belief change (HSBC) construction [2] to accommodate a stream of observations. Agent actions and environment events are distinguishable and form part of the agent model. It is left up to the agent designer to provide an environment model; a submodel of the agent model. Observations in the stream which are no longer relevant, become default assumptions until overridden by newer, more prevalent observations. A distinction is made between background and foreground beliefs. Voorbraak's [3] partial probability theory (PPT) is used as guidance for the 'dual-belief-base' approach. In PPT, the agent may be somewhat ignorant, or might not have all (probabilistic) knowledge desired. Consider the following scenario. Your neighbour tells you he needs to visit the dentist urgently. You know that he uses the dentist at the Wonder-mall. A month later, you see your neighbour at the Wonder-mall. Is he there to see the dentist? The answer to the question makes use of the persistence of truth of certain pieces of information. After a period has elapsed, the veracity of some kinds of information dissipates. One would expect a person who says they must visit the dentist urgently to visit the dentist within approximately seven days. So your neighbour is probably not at Wonder-mall to see the dentist. Hence 'Neighbour must visit dentist' should be true for no longer than seven days, after which, the statement becomes *defeasibly* true.

2 FRAMEWORK DEFINITION

Let L be a finite classical propositional language. A *world* is a logical model which evaluates every propositional variable to true or false, and by extension, evaluates every propositional sentence in L to true or false. Let W be a set of possible worlds – a subset of the conceivable worlds. The fact that $w \in W$ satisfied $\phi \in L$ is denoted by $w \Vdash \phi$. Let L^{prob} be a probabilistic language over L defined as $L^{prob} := \{\phi[\ell, u] \mid \phi \in L,\ \ell, u \in [0,1], \ell \leq u\}$. A sentence of the form $\phi[\ell, u]$ means the likelihood of proposition ϕ is greater than or equal to ℓ and less than or equal to u. Let b be a probability distribution over all the worlds in W. For all $\phi \in L$, $b(\phi) := \sum_{w \in W, w \Vdash \phi} b(w)$. b *satisfies* formula $\phi[\ell, u]$ (denoted $b \Vdash \phi[\ell, u]$) iff $\ell \leq b(\phi) \leq u$.

In POMDP theory, to update an agent's beliefs about the world, a *state estimation function* $SE(b, a, z) = b^{SE}_{a,z}$ is defined as

$$b^{SE}_{a,z}(s') = \frac{O(a, z, s') \sum_{s \in S} T(s, a, s') b(s)}{Pr(z \mid a, b)},$$

where a is an action performed in 'current' belief-state b, z is the resultant observation, $T(s, a, s')$ is the probability of being in s' after performing action a in state s, $O(a, z, s')$ is the probability of observing z in state s' resulting from performing action a in some other state, and $b^{SE}_{a,z}(s')$ denotes the probability of the agent being in state s' in 'new' belief-state $b^{SE}_{a,z}$.

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$. The proposed framework is built around the structure $\langle W, B, F, A, Evt, Z, Prs, Eng, \mathcal{M} \rangle$, where

- W is a set of possible worlds;
- $B \subset L^{prob}$ is a *background belief base* of fixed assumptions;
- $F \subset L^{prob}$ is a *foreground belief base* of default assumptions;
- A is a set of (agent) *actions*, including a special action *null*;
- Evt is a set of (environment) *events*;
- Z is the *observation stream*, a set of observation triples: $Z := \{(a_1, t_1, z_1), (a_2, t_2, z_2), \ldots, (a_k, t_k, z_k)\}$, where $a_i \in A$, $t_i \in \mathbb{N}$, $z_i \in L$, and such that $\forall t_i, t_j \in \mathbb{N}$, $i = j$ iff $t_i = t_j$ (i.e., no more than one action and observation occur at a time-point);
- $Prs : L \times W \to \mathbb{N}$ is a *persistence function*, where $Prs(z, w)$ indicates how long z is expected to be true from the time it is received, given the 'context' of w; it is a total function over $L \times W$;
- $Eng : L \times W \times A \to [0, 1]$, where $Eng(z, w, a)$ is the agent's confidence that z perceived in w was caused by action a (i.e., that z has an endogenous source);
- $\mathcal{M}$ is a model of the environment, and any auxiliary information required by the definition of the particular belief change operation ($\lhd$).

The expected persistence of z perceived in belief state b is $ExpPrs(z, b) := \sum_{w \in W} Prs(z, w) \times b(w)$. An observation triple $(a, i, z) \in Z$, has *expired* at point s if $ExpPrs(z, b) < s - i$.

Let $b^{\lhd}_{a,z}$ be the *change* of belief state b by a and z. My intention is that "change" is a neutral term, not necessarily indicating revision or update. Here, belief *update* is taken to mean belief change due to an observation received, where the observation is the signal generated by some event which changed the world. And belief *revision* is taken to mean belief change due to an observation received in a static world, in other words, revision occurs when information does not have its source in the changing world, but in an 'announcement'. Update focuses on the effects of physical agent-actions and revision focuses on new evidence received (implicitly or explicitly postulating an event as source).

[1] An unabridged version of this paper was presented at the Third International Workshop on Defeasible and Ampliative Reasoning, 29 August 2016, The Hague, Netherlands.

[2] Centre for Artificial Intelligence Research, University of KwaZulu-Natal, School of Mathematics, Statistics and Computer Science and CSIR Meraka, South Africa, email: gavinrens@gmail.com

Next, I propose one instantiation of $b^{\lhd}_{a,z}$. Let the environment model $\mathcal{M} = \langle E, T, O\rangle$, where

- $E : Evt \times W \to [0,1]$ is the *event function*. $E(e,w) = P(e \mid w)$, the probability of the occurrence of event e in w;
- $T : W \times (A \cup Evt) \times W \to [0,1]$ is a *transition function* such that for every æ $\in A \cup Evt$ and $w \in W$, $\sum_{w' \in W} T(w, æ, w') = 1$, where $T(w, æ, w')$ models the probability of a transition to world w', given the execution of action / occurrence of event æ in world w;
- $O : W \times W \times A \to [0,1]$ is an *observation function* such that for every $w \in W$ and $a \in A$, $\sum_{w^z \in W} O(w^z, w, a) = 1$, where $O(w^z, w, a)$ models the probability of observing ϕ^z (a complete theory for w^z) in w and where $O(z, w, a) := \sum_{\substack{w^z \in W \\ w^z \Vdash z}} O(w^z, w, a)$, for all $z \in L$.

Observation z may be due to an *exogenous* event (originating and produced outside the agent) or an *endogenous* action (originating and produced within the agent). It is up to the agent designer to decide, for each observation, whether it is exogenous or endogenous, given the action and world.

The hybrid stochastic belief change (HSBC) formalism of Rens [2] defines the (exogenous) update of b with z as

$$b^{\diamond}_z := \{(w,p) \mid w \in W, p = \frac{1}{\gamma} O(z,w) \sum_{w' \in W} \sum_{e \in Evt} E(e,w') T(w', e, w) b(w')\},$$

where γ is a normalizing factor and $O(z,w)$ can be interpreted as $O(z, w, null)$. But HSBC does not involve agent actions; HSBC assumes that agents passively receive information. Hence, when an agent is assumed to act and the actions are known, the POMDP state estimation function can be employed for belief state update.

Only for the propose of illustrating how the framework can be used, the following belief change procedure is defined.

$$b^{\lhd}_{a,z} := \{(w,p) \mid w \in W, p = Exg(z,w,a) b^{\diamond}_z(w) + Eng(z,w,a) b^{SE}_{a,z}(w)\}, \quad (1)$$

where $Exg(z,w,a) := 1 - Eng(z,w,a)$ is the confidence that z is exogenous in w, given a was executed.

3 OPERATIONAL SEMANTICS

Expired observations are continually aggregated into the agent's set of default assumptions F (foreground beliefs). Prevalent (unexpired) observations remain in the agent's 'current memory stream' Z. Whenever the agent wants to perform some reasoning task, (i) it combines the prevalent observations with its fixed beliefs B, then modifies its changed beliefs with respect to its default assumptions, or (ii) modifies its fixed beliefs B with respect to its default assumptions, then combines the prevalent observations with its changed beliefs – and then reasons with respect to this final set of beliefs. However, the agent always reverts back to the original fixed beliefs (hence, "fixed"). And the default assumptions keep changing as memories fade, that is, as observations expire. The agent's initial beliefs (i.e., the system conditions at point $\mathcal{N} = 0$) must be specified in F.

Let $C \subset L^{prob}$ be a belief base. A practical approach is to reduce Π^C to a representative belief state (denoted $\lhd_{ME}$), employing the principle of maximum entropy.

$$\Pi^C \ \lhd_{ME} \ a, z := \{b^{\lhd}_{a,z} \mid b = ME(\Pi^C)\},$$

where $ME(\Pi^C)$ is the belief state in Π^C with maximum entropy.

Another approach (denoted $\lhd_{FS}$) is to find a finite (preferably, relatively small) subset of $\Pi^{FS} \subset \Pi^C$ which is somehow representative of Π^C, and then applies $\lhd$ to the individual belief states in Π^{FS}:

$$\Pi^C \ \lhd_{FS} \ a, z := \{b^{\lhd}_{a,z} \mid b \in \Pi^{FS} \subset \Pi^C\}.$$

Let $\lhd_{set}$ denote one of these two operators. Then $\Pi^C \lhd_{set} \sigma$ can be defined, where σ is any stream of observation triples, where $\sigma = \{(a_1,t_1,z_1),(a_2,t_2,z_2),\ldots,(a_k,t_k,z_k)\}$ and $t_1 < t_2 < \cdots < t_k$. That is, $\Pi^C \leftarrow \Pi^C \ \lhd_{set} \ a_1, z_1$ and then $\Pi^C \leftarrow \Pi^C \ \lhd_{set} \ a_2, z_2$ ...and then $\Pi^C \leftarrow \Pi^C \ \lhd_{set} \ a_k, z_k$. Let *Expired* be derived from the triples in Z which have just expired (at point $\mathcal{N}$). That is,

$$Expired := \{(a,i,z) \in Z \mid ExpPrs(z, \Pi_{\mathcal{N}-1}) < \mathcal{N} - i\},$$

where $\Pi_{\mathcal{N}}$ is defined below. At each time-point, F is refreshed with all the expired observation triples in the order they appear in Z. In other words, at each point in time, $F \leftarrow Th(\Pi^F \lhd_{set} Expired)$, where $Th(\Pi^C) := \{\psi \in L^{prob} \mid b \in \Pi^C, b \Vdash \psi\}$. As soon as the foreground has been refreshed, the expired triples are removed from the observation stream: $Z \leftarrow Z \setminus Expired$. I shall use the notation $\Pi^F_{\lhd_{set}}$ to clarify which operation was used to arrive at the current Π^F.

Reasoning is done with respect to the set of belief states $\Pi_{\mathcal{N}}$. I look at two patterns of cognition to determine $\Pi_{\mathcal{N}}$.

$$(\Pi_{\mathcal{N}})_1 := \begin{cases} (\Pi^B \lhd_{set} Z) \cap \Pi^F_{\lhd_{set}} \text{ if } (\Pi^B \lhd_{set} Z) \cap \Pi^F_{\lhd_{set}} \neq \emptyset \\ \{Closest(\Pi^B \lhd_{set} Z, \Pi^F_{\lhd_{set}})\} \text{ otherwise,} \end{cases}$$

or

$$(\Pi_{\mathcal{N}})_2 := \begin{cases} (\Pi^B \cap \Pi^F_{\lhd_{set}}) \lhd_{set} Z \text{ if } \Pi^B \cap \Pi^F_{\lhd_{set}} \neq \emptyset \\ \{Closest(\Pi^B, \Pi^F_{\lhd_{set}})\} \lhd_{set} Z \text{ otherwise,} \end{cases}$$

where $Closest(X,Y)$ refers to an abstract function which selects a belief state in X which is in some sense closest to set Y of belief states.

4 CONCLUDING REMARKS

I believe that it is important to have the facility to reason about both (exogenous) events and (endogenous) actions; I am definitely not the first to propose a framework with both notions. Inherent to the framework is that the agent's knowledge may be incomplete. Much work dealing with ignorance or missing information has been done.

One line of research that should be followed is to generalize the dual-belief-base approach: What would happen if three, four, more belief bases are employed, each accommodating a different degree of entrenchment of given and received information?

Keeping the belief base B fixed is quite a strong stance. In reality, only the most stubborn people will never change their core views even a little bit. In future versions, I would like to make B more amenable to learning, while minding sound principles of belief change in logic and cognitive psychology.

REFERENCES

[1] W. Lovejoy, 'A survey of algorithmic methods for partially observed Markov decision processes', *Annals of Operations Research*, **28**, 47–66, (1991).

[2] G. Rens, 'On stochastic belief revision and update and their combination', in *Proceedings of the Sixteenth Intl. Workshop on Non-Monotonic Reasoning (NMR)*, eds., G. Kern-Isberner and R. Wassermann, pp. 123–132. Technical University of Dortmund, (2016).

[3] F. Voorbraak, 'Partial Probability: Theory and Applications', in *Proceedings of the First Intl. Symposium on Imprecise Probabilities and Their Applications*, pp. 360–368, (1999). url: decsai.ugr.es/ smc/isipta99/proc/073.html.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1559

Topic-Level Influencers Identification in the Microblog Sphere

Yakun Wang and **Zhongbao Zhang** and **Sen Su** and **Cheng Chang** and **Muhammad Azam Zia**[1]

Abstract. This paper studies the problem of identifying influencers on specific topics in the microblog sphere. Prior works usually use the cumulative number of social links to measure users' topic-level influence, which ignores the dynamics of influence. As a result, they usually find faded influencers. To address the limitations of prior methods, we propose a novel probabilistic generative model to capture the variation of influence over time. Then a influence decay method is proposed to measure users' current topic-level influence.

1 Introduction

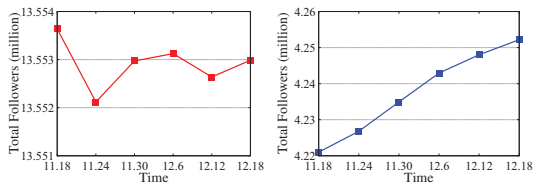
Researchers have focused on topic-level influence analysis [1, 6]. In general, people want to find recent influencers rather than outdated ones. In these prior studies, it is a common way to utilize the cumulative number of social links (e.g., followship, reposts and mentions) to identify the topic-level influencers. However, we observe that the links are created over time. For measuring users' influence, it is critical to incorporate the variation trend of influence. A real example about two famous basketball players Jianlian Yi and Jeremy Lin in Sina Weibo is illustrated in Fig. 1. We can see that although Yi has more followers than Lin, the number of Yi's followers no longer increases, while Lin gets more and more followers along with time. Accordingly, we can not simply assume Yi has more influence than Lin just because Yi owns more followers. However, if assuming both Yi and Lin are followed for basketball, all prior methods will select Yi as the key influencer rather than Lin, which leads to inaccurate models. This example conveys that the learned influence by the cumulative number of links is far from adequate, since users' influence is dynamic and rises or falls over time [3].

In this paper, we intend to identify recent popular and influential users on specific topics rather than faded ones. To address this problem, we firstly propose a novel probabilistic generative model, which we refer to as Topic-level Influence over Time (TIT), for capturing the temporal aspect of influence on specific topics. Then we design an exponential decay method that works on the learned temporal influence to compute the influence of each user on specific topics, which takes both quantity and trend of influence into consideration. Through extensive experiments on real-world dataset, we demonstrate the effectiveness of our approach.

2 Topic-Level Influence Analysis

2.1 TIT Model

Firstly, we intend to model the topic-level influence over time, which can better help us capture the dynamics of influence. We propose a Topic-level Influence over Time (TIT) model jointly over text, links and time based on the LDA model [2]. It uncovers the latent topics and users' topic-level temporal influence in a unified way. The plate notation is given in Fig. 2. Specifically, there are two components in this model: the user-word component in the right part and the user-(link, time) component in the left part of Fig. 2.



(a) Jianlian Yi (b) Jeremy Lin

Figure 1. The Number of Total Followers over Time (Year 2015)

The user-word component is to model user u's words. We aggregate the words w posted by u into an integrated document from which we use LDA model to discover the latent topics. As a result, each user has a Multinomial distribution θ over topics and each topic has a Multinomial distribution φ over words. The user-(link, time) component is to model the u's links (e.g., followship) and the corresponding generation time in the microblog network. We discretize the time by dividing the entire time span of all links into T time slices. We consider the network as a document corpus and each user u is represented by a document where user f that u communicated with and the corresponding time t pairs form the words in this document. Note that this component consists of two levels of mixtures: an upper-level Bernoulli mixture μ and two underneath-level multinomial mixture parts σ and π. μ is for deciding whether the link creation is based on u's topics or not. If topic based, we model the topic x (generated by θ) over (f, t) by a multinomial distribution σ. Otherwise, we use a global multinomial distribution π to model (f, t). Benefiting from the learning results of σ, we can generate the influence trend line over time of each user like Fig. 1, and this can greatly help us to identify the key topic-level influencers on microblogs.

2.2 Parameter Estimation

We use Gibbs sampling [5] to obtain samples of the hidden variable assignment and to estimate the model parameters from these samples. Let $x_{\neg i}$ denote the set of all hidden variables of topics except x_i and $n^{(\cdot)}_{\cdot,\neg i}$ denote the count that the element i is excluded from the corresponding topic or user. Here, we only give the sampling formula of links. For a link f_i and the corresponding time t_i with index $i = (u, l)$, we jointly sample y_i and x_i from the conditional as the following two equations:

$$p(x_i, y_i = 1 | f, t, x_{\neg i}, y_{\neg i}, z, \alpha, \gamma, \rho) \quad (1)$$

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China, email: {wangyakun, zhongbaozb, susen, changwow, zia}@bupt.edu.cn. Corresponding author of this paper is Prof. Sen Su.

$$\propto \frac{n^{(f,*)}_{k,\neg i}+\gamma}{\sum_{f=1}^{U} n^{(f,*)}_{k,\neg i}+U\gamma}(n^{(y=1)}_{u,\neg i}+\rho_1)(n^{(k)}_{u(w)}+n^{(k)}_{u(f),\neg i}+\alpha)$$

$$p(x_i, y_i = 0|f, t, x_{\neg i}, y_{\neg i}, z, \alpha, \epsilon, \rho) \quad (2)$$

$$\propto \frac{n_{(f,*),\neg i}+\epsilon}{\sum_{f=1}^{U} n_{(f,*),\neg i}+U\epsilon}(n^{(y=0)}_{u,\neg i}+\rho_0)(n^{(k)}_{u(w)}+n^{(k)}_{u(f),\neg i}+\alpha),$$

where $n^{(f,*)}_{k,\neg i}$ denotes the number of times that user f occurs in topic k, $n_{(f,*),\neg i}$ denotes the number of times that user f occurs without any topic, $*$ represents an aggregation on time dimension, $n^{(k)}_{u(w)}$ denotes the number of times that topic k has been observed with a word w of user u, and $n^{(k)}_{u(f),\neg i}$ denotes the number of times that topic k has been observed with a link f of u, $n^{(y=1)}_{u,\neg i}$ and $n^{(y=0)}_{u,\neg i}$ denote the number of times the links created by u is related to topics or regardless of topics, respectively.

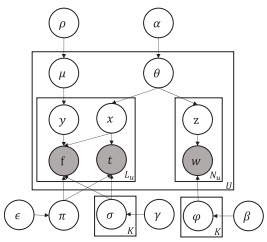


Figure 2. Plate diagram of TIT

2.3 Measuring Users' Influence

Given the topic-level influence trend lines over time derived from σ, users who get lots of attention from others and have a upward trend of influence can be easily found as the key influencers on the corresponding topics. However, for some cases like the example Yi and Lin in Fig. 1, we can not easily identify who exhibits more influence, since Yi has more followers than Lin, while Lin has a better growing trend of influence than Yi. Intuitively, links generated long time ago have little contribution to users' influence. It means the more closer of the links generated in time, the more important they are to users' influence. Hence, we utilize the exponential decay function to model the influence decay. Specifically, σ is a distribution of topics over a set of 2-tuples $\{(f,t)\}$. That is, σ is a $U \times T \times K$ matrix in the procedure of sampling recording the number of times (f,t) has been assigned to topic k, denoted as $n^{(f,t)}_k$, plus prior parameter γ, i.e., $\sigma_{u,t,k} = n^{(f,t)}_k + \gamma$. Thus, we can use the following equation to measure the influence of user f on topic k till time T:

$$Influence(f)@(k,T) = \gamma + \sum_{t=1}^{T} n^{(f,t)}_k \times e^{-\frac{T-t}{\lambda}} \quad \lambda{>}0. \quad (3)$$

Here, λ is a parameter controlling the decay rate of influence.

3 Experiment

Dataset: We crawl the followship network from Sina Weibo[2]. Since Sina Weibo does not release the information about when a user follows another, we periodically crawl the follow list of all users in our seed set, monitor their changes and then label the new generated links with timestamps. We also crawl their recent 100 messages. Finally, after preprocessing, there are 0.4M users, 207M words, 46M links with 7M time-tagged and 24 time slices with each nearly 1.5 days in our dataset. We empirically set the values of hyperparameters of TIT in line with other topic modelling work [1]. We set $\lambda = 11$ through minimizing held-out perplexity on a validation set.

[2] http://weibo.com

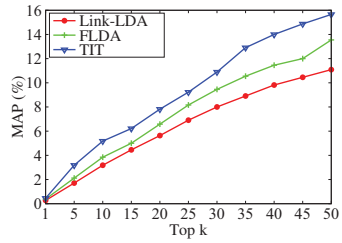


Figure 3. All Categories

Precision: We evaluate our approach by comparing it with Link-LDA [4] and Followship-LDA (FLDA) [1]. For the ground truth, Sina Weibo gives the lists of popular users or organizations about 36 categories such as sports and music. Each category list contains 100 ranked users. It is clear that these popular users or organizations are some kind of the key influencers on the corresponding topics. Sina Weibo states that these lists are updated by month. Intuitively, TIT considering the temporal dynamics of influence should produce more precise results. Although these rankings do not necessarily have 100% precision, they give us enough information to facilitate relative comparisons across different approaches. Fig. 3 shows the results of Mean Average Precision (MAP) across all categories. It is clear that TIT significantly outperforms the competitors.

4 Conclusion

This paper studies the problem of analyzing the topic-level temporal influence of users for the finding recent influencers on specific topics in microblog sphere. To achieve this, we first propose the TIT (Topic-level Influence over Time) model, a novel probabilistic generative model jointly over text, links and time. Then, we design an influence decay based approach to measure users' topic-level influence from the learned temporal influence. We compare our approach with Link-LDA and FLDA on a real dataset crawled from Sina Weibo. Experimental results demonstrate the effectiveness of our approach.

5 Acknowledgements

This work is supported in part by the following funding agencies of China: National Natural Science Foundation under Grant 61170274 and U1534201, and the Fundamental Research Funds for the Central Universities (2015RC21).

REFERENCES

[1] Bin Bi, Yuanyuan Tian, Yannis Sismanis, Andrey Balmin, and Junghoo Cho, 'Scalable topic-specific influence analysis on microblogs', in *Proceedings of the 7th ACM international conference on Web search and data mining*, pp. 513–522. ACM, (2014).

[2] David M Blei, Andrew Y Ng, and Michael I Jordan, 'Latent dirichlet allocation', *the Journal of machine Learning research*, **3**, 993–1022, (2003).

[3] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and P Krishna Gummadi, 'Measuring user influence in twitter: The million follower fallacy.', *ICWSM*, **10**(10-17), 30, (2010).

[4] Elena Erosheva, Stephen Fienberg, and John Lafferty, 'Mixed-membership models of scientific publications', *Proceedings of the National Academy of Sciences*, **101**(suppl 1), 5220–5227, (2004).

[5] Thomas L Griffiths and Mark Steyvers, 'Finding scientific topics', *Proceedings of the National Academy of Sciences*, **101**(suppl 1), 5228–5235, (2004).

[6] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He, 'Twitterrank: finding topic-sensitive influential twitterers', in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 261–270. ACM, (2010).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1561

Multilevel Agent-Based Modelling for Assignment or Matching Problems

Antoine Nongaillard[1] and **Sébastien Picault** [1]

Abstract. Assignment or matching problems have been addressed by various multi-agent methods, focused on enhancing privacy and distribution. Nevertheless, they little rely on the organisational structure provided by Multi-Agent Systems (MAS). We rather start from the intrinsic ability of multilevel MAS to represent intermediate points of view between the individual and the collective levels, to express matching or assignment problems in a homogeneous formalism. This model allows to define relevant metrics to assess the satisfaction of agent groups and allow them to build solutions that improve the overall well-being without disclosing all their individual information.

1 INTRODUCTION AND LITERATURE

A problem of resource allocation aims at distributing a resource set among a population of individuals, optimising a goal defined as the aggregation of individual measures. A matching problem aims at grouping individuals, by optimising a goal defined as the aggregation of assessments made by each individual on other members of their group. In both cases, objectives are often expressed as an aggregation of individual evaluations using the social choice theory [2]. These two families, despite their similarities, are addressed as different problems, because of the possibility or not for "resources" to express preferences towards other members of their group. They are thus resolved by algorithms dedicated to a specific families, or even specific sub-problems.

The most common resolution methods are centralised and based on complete information. Indeed, all preferences or constraints are public and manipulated by an overall solver. The most known algorithms in this field (e.g., the Hungarian algorithm [5] or the Gale-Shapley algorithm [4])focus in identifying optimal solutions in absolute terms, by making the strong assumption of a full and public information. Mechanisms for achieving such a solution from a given starting point is not a concern of these methods.

Recently, these problems have also been modelled using the multi-agent paradigm. On the one hand, one can consider the problem of resources/tasks allocation within a population of agents: MAS are then just an application area [1, 9], for which it seems natural to seek distributed solving methods [6]. On the other hand, MAS can be used as a tool to solve allocation or matching problems in a distributed way. Some of these approaches are a mere distribution of computations [7, 3] and do not consider the *behaviours* required for agents to achieve a solution, but only protocol issues and solution characteristics.

[1] Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL (SMAC), F-59000 Lille, France, email: name.surname@univ-lille.fr
This project is granting on the PartEns project, within the research program "*chercheur citoyen*" from the region Hauts de France.

Instead of addressing the very issue of distributing either the preferences or the resolution among a population of agents for a particular problem, we rather try to build a generic multilevel structure which allows the modelling of assignment or matching problems, and the expression of multiple concerns at the same time.

Thus, we propose a multilevel modelling of these problemswhich increases the variety of constraints and preferences that can be expressed at every organisation level, using the ability to build composite metrics.

2 MULTILEVEL MODEL

The multilevel formalism we are proposing assume that all relevant entities of the model (e.g. "individuals", "resources", "groups"...) are represented by agents and linked by a membership relation. Those relations are not necessarily hierarchical, since for example some problems may imply resource sharing or allow individuals to be members of several groups at the same time. In order to implement these features, the formalism we chose is a multilevel agent-based simulation meta-model called "PADAWAN" [8].

The set of all agents is denoted by $\mathbb{A}$; $a_1 \sqsubset a_2$ means that agent a_1 is **contained** by agent a_2 (or a_2 is **host** to a_1).The $\sqsubset$ relationship induces an oriented, cycle-free, *hosting graph* between the agents.

For any agent a we also define its *hosts* and conversely the set of all agents *contained* in a respectively as: $hosts(a) = \{a_i \in \mathbb{A} | a \sqsubset a_i\}$ and $content(a) = \{a_j \in \mathbb{A} | a_j \sqsubset a\}$

Our approach consists in grounding the computation of welfare values in the very membership relations in the MAS. We decompose the individual welfare of any agent into three contributions representing respectively the agent as *an individual*, as *the neighbour of other agents*, and as the *host to other agents*. Thus we have: $w(a) = f_a(\sigma(a), \mu(a), \gamma(a))$ where f_a function can be chosen arbitrarily, depending on the situation to be modelled.

The $\sigma(a)$ contribution represents *the satisfaction of agent a as an individual, which is also situated in a given structure*. Thus, this value can be computed from the state of agent a, but also according to the *perceived* properties of its hosts and of their own "ancestors": $\mathcal{H}(a) = \{h \in \mathbb{A} | a \sqsubset h \vee \exists h' \in \mathcal{H}(a), h' \sqsubset h\}$

For instance, in a problem where individuals have to be assigned to groups within organisations, the satisfaction of a person depends on its own state, on the features of its *role* within its group, on the properties of the group itself, but also on the organisation this group is part of, etc.

The corresponding value can then be computed by using an operator $\bigoplus^a$, *specific to agent a* (and to be defined for each concrete situation), in order to aggregate the perceptions by agent a of its *situation*. This situation is composed of the properties $\chi_a(h)$, perceived

by agent a, of itself and of all agents that contain a (either directly or transitively):

$$\sigma(a) = \bigoplus_{h \in \{a\} \cup \mathcal{H}(a)}^{a} \chi_a(h)$$

The $\mu(a)$ contribution represents the *satisfaction of agent a as member of a group*, in other words the contribution of externalities due to the presence of other agents within the same hosts.

$$\nu(a) = \bigcup_{h \in hosts(a)} content(h) \backslash \{a\}$$

The perceived properties of those neighbours are then aggregated using another operator which we denote by $\circledcirc^a$:

$$\mu(a) = \bigcircledcirc_{n \in \nu(a)}^{a} \chi_a(n)$$

The $\gamma(a)$ contribution represents the satisfaction of agent a *as the representative of a group of agents*. This satisfaction as a group is intended to measure the *collective welfare* of the agents contained in a, which can be done by aggregating the perceived properties of the agents contained in a, using a third domain-dependent operator, denoted by $\bigoplus^a$:

$$\gamma(a) = \bigoplus_{m \in content(a)}^{a} \chi_a(m)$$

3 EXAMPLE

In order to illustrate the specific contribution of our multilevel welfare assessment meta-model, we present below a complex problem involving multiple membership, demonstrating the capabilities of our meta-model to represent various points of view (hence, various objectives), specific to each agents family or to each level.

Here we consider individuals ($\mathcal{I}$) allowed to enrol in several associations ($\mathcal{A}$). These associations can gather into federations ($\mathcal{F}$) and can be funded either by municipalities ($\mathcal{M}$) or by regions ($\mathcal{R}$) (Figure 1). The individuals seek to maximise their participation in the associations offering their preferred activities. The associations and federations intend to assert their size (enrolment) so as to defend their grant requests. The municipalities aim at allocating their available budget as fairly as possible between their local associations. The regions try to do the same between municipalities and federations.

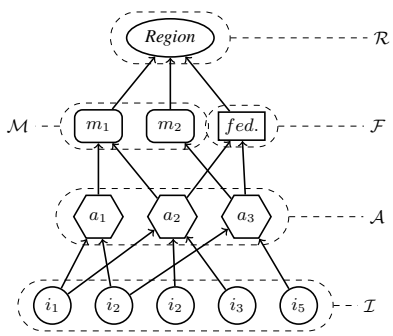


Figure 1. Hosting graph representing a common associative network, with individuals ($\mathcal{I}$) enrolled in several associations ($\mathcal{A}$), possibly grouped into federations ($\mathcal{F}$) and funded by municipalities ($\mathcal{M}$) or regions ($\mathcal{R}$).

In this example we assume that the grant allocation is based on a Nash welfare, in order to prevent the largest structures from monopolising the budget, and yet taking their enrolment into account, hence: $\bigoplus^{\mathcal{R}} = \bigoplus^{\mathcal{M}} = \prod$. On the contrary, an association a which only intends to assert the number of registered members (without consideration for their satisfaction), can simply use $\chi_a(i) = 1$ for every member i and $\bigoplus^{\mathcal{A}} = \sum$. The weight of an association obviously depends on its membership of a federation, thus for each of its hosts h, we assume $\chi_a(h) = 1$ if $h \in \mathcal{F}$, and $\chi_a(h) = 0$ otherwise, with $\bigoplus^{\mathcal{A}} = \sum$. The individuals are rather in search of a trade-off between the participation to their preferred activities and the cumulative cost of these activities (which requires a cost matrix $(c_{i,a})$). Again, $\bigoplus^{\mathcal{I}} = \sum$ can be used. But, in addition, people are highly sensitive to their neighbours, i.e. the other individuals enrolled in the same associations. This can be handled for instance through an affinity-based approach, and an aggregation operator $\circledcirc^{\mathcal{I}}$ either optimistic ($\max$), or pessimistic ($\min$).

To summarise, the diversity of objectives within the MAS is made explicit by using a large diversity of metrics, within a structure where all agents are otherwise homogeneous. We argue that this systematisation is the basis for defining generic resolution principles, which can deal with quite different situations, in opposition of classical approaches (either centralised or distributed) where each specific problem is to be solved by its specific method. Indeed, we assume that our meta-model allows the construction of intrinsically multi-agent resolution methods, i.e. relying on local perceptions and interactions between agents, and on generic behaviours subject to context-dependent settings.

4 CONCLUSION, PERSPECTIVES

We proposed a uniform approach to model assignment or matching problems through a multilevel multi-agent system. We have shown the capability of this formalism to express a broad diversity of objectives, representing the viewpoints of the actors of the system. We are currently working on the construction of several protocols, not dedicated to a specific problem, but rather based on the nature of the hosting graph between agents. These algorithmic aspects mix with methodological considerations, so as to identify couplings between the nature and structure of the addressed problems, and the behaviours to give the agents to enable them build a solution in an incremental way.

REFERENCES

[1] S. Airiau and U. Endriss, 'Multiagent resource allocation with sharable items', *JAAMAS*, **28**(6), 956–985, (2013).
[2] K.J Arrow, *Social Choice and Individual Values*, J Wiley, 1963.
[3] I. Brito and P. Meseguer, 'Distributed stable matching problems', *LNCS – Principles and Practice of Constraint Programming*, **3709**, 152–166, (2005).
[4] D. Gale and L.S. Shapley, 'College admissions and the stability of marriage', *American Mathematical Monthly*, **69**, 9–14, (1962).
[5] H.W. Kuhn, 'The Hungarian method for the assignment problem', *Naval Research Logistics Quarterly*, **2**, 83–97, (1955).
[6] K.S. Macarthur, R. Stranders, S.D. Ramchurn, and N.R. Jennings, 'A distributed anytime algorithm for dynamic task allocation in multi-agent systems', in *AAAI Conf. on Artificial Intelligence*, (2011).
[7] A. Netzer, A. Meisels, and R. Zivan, 'Distributed envy minimization for resource allocation', *JAAMAS*, **30**(2), 364–402, (2015).
[8] S. Picault and P. Mathieu, 'An interaction-oriented model for multi-scale simulation', in *22nd Int. Joint Conf. on Artificial Intelligence (IJCAI)*, ed., T. Walsh, pp. 332–337, (2011).
[9] M.M. Weerdt, Y. Zhang, and T. Klos, 'Multiagent task allocation in social networks', *JAAMAS*, **25**(1), 46–86, (2011).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1563

Simple Epistemic Planning: Generalised Gossiping

Martin C. Cooper Andreas Herzig Faustine Maffre Frédéric Maris Pierre Régnier[1]

Abstract. The gossip problem, in which information (secrets) must be shared among a certain number of agents using the minimum number of calls, is of interest in the conception of communication networks and protocols. We extend the gossip problem to arbitrary epistemic depths. For example, we may require not only that all agents know all secrets but also that all agents know that all agents know all secrets. We give optimal protocols for the generalised gossip problem, in the case of two-way communications, one-way communications and parallel communication. In the presence of negative goals testing the existence of a successful protocol is NP-complete.

1 Introduction

We consider communication problems concerning n agents. We consider that initially, for $i = 1, \ldots, n$, agent i has some information s_i, also known as this agent's secret since, initially, the other agents do not know this information. In many applications, this corresponds to information that agent i wishes to share with all other agents. On the other hand, it may be confidential information which is only to be shared with a subset of the other agents. The simplest version of the problem in which all agents want to communicate their secrets to all other agents (using the minimum number of communications) is traditionally known as the *gossip problem*. Several variants have been studied, and a survey has been published [5].

The gossip problem is a particular case of a multiagent epistemic planning problem. Our main contribution is to study the gossip problem at different epistemic depths. In the classic gossip problem, the goal is for all agents to know all secrets (which corresponds to epistemic depth 1). The equivalent goal at epistemic depth 2 is that all agents know that all agents know all the secrets; at depth 3, all agents must know that all agents know that all agents know all the secrets.

All proofs can be found in the full-length version of this article [3].

2 Epistemic planning and the gossip problem

Dynamic Epistemic Logic DEL [9] provides a formal and very expressive framework for the representation and update of knowledge, and several recent approaches to multi-agent planning are based on it. While DEL provides a very expressive framework, even simple fragments of it have unfortunately been proven to be undecidable [1]. We here consider a simple fragment of the language of DEL where the knowledge operator can only be applied to literals [2].

We use the notation $K_i s_j$ to represent the fact that agent i knows the secret of j, the notation $K_i K_j s_k$ to represent the fact that agent i knows that agent j knows the secret of k, etc. We use the term *positive fluent* for any epistemic proposition of the form $K_{i_1} \ldots K_{i_r} s_j$. If we consider the secrets s_i as constants and that agents never forget, then positive fluents, once true, can never become false. A negative fluent $\neg(K_{i_1} \ldots K_{i_r} s_j)$ can, of course, become false.

A *planning problem* consists of an initial state (a set of fluents I), a set of actions and a set of goals (another set of fluents $Goal$). Each action has a (possibly empty) set of preconditions (fluents that must be true before the action can be executed) and a set of effects (positive or negative fluents that will be true after the execution of the action). A *solution plan* (or protocol) is a sequence of actions which when applied in this order to the initial state I produces a state in which all goals in $Goal$ are true. An example of a goal is $\forall i, j, k \in \{1, \ldots, n\}$, $K_i K_j s_k$, i.e. that all agents know that all agents know all the secrets.

The *gossip problem* on n agents and a graph $G = \langle \{1, \ldots, n\}, E_G \rangle$ is the planning problem in which the actions are $\text{CALL}_{i,j}$ for $\{i, j\} \in E_G$ (i.e. agents i and j can call each other iff there is an edge between i and j in G) and the initial state contains $K_i s_i$ for $i = 1, \ldots, n$ (and implicitly all fluents of the form $K_{i_1} \ldots K_{i_r} s_j$ with $i_r = j$) The action $\text{CALL}_{i,j}$ has no preconditions and its effect is that agents i and j share all their knowledge. We go further and assume that the two agents know that they have shared all their knowledge, so that, if we had $K_i f$ or $K_j f$ before the execution of $\text{CALL}_{i,j}$, for any fluent f, then we have $K_{i_1} \ldots K_{i_r} f$ just afterwards, for any r and for any sequence $i_1, \ldots, i_r \in \{i, j\}$.

Let Gossip-pos$_G(d)$ be the gossip problem on a graph G in which the goal is a conjunction of positive fluents of the form $(K_{i_1} \ldots K_{i_r} s_j)$ $(1 \leq r \leq d)$. Thus, the parameter d specifies the maximum epistemic depth of goals. Gossip$_G(d)$ denotes the specific problem in which *all* such goals must be attained. We drop the subscript G to denote the corresponding problem in which the graph G is not fixed but part of the input.

3 Minimising the number of calls for positive goals

In this section we consider the gossip problem at epistemic depth d. For $d = 1$, the minimal number of calls to solve Gossip$_G(1)$ is either $2n - 4$ if the graph G contains a quadrilateral (a cycle of length 4) as a subgraph, or $2n - 3$ in the general case [4].

Proposition 1 *If the graph G is connected, then for $n \geq 2$ and $d \geq 1$, any instance of Gossip-pos$_G(d)$ has a solution of length no greater than $d(2n - 3)$ calls.*

For $d \geq 2$, we require considerably less than $d(2n - 3)$ calls for certain graphs since we can often achieve $(d + 1)(n - 2)$. The complete bipartite graph with parts $\{1, 2\}$, $\{3, \ldots, n\}$ is denoted in graph theory by $K_{2,n-2}$. There is a protocol which achieves $(d + 1)(n - 2)$ calls provided G contains $K_{2,n-2}$ as a subgraph. This subsumes a previous result which was given only for the case of a complete graph G [6]. Detecting whether an arbitrary graph G has $K_{2,n-2}$ as a subgraph can be achieved in polynomial time.

[1] IRIT, University of Toulouse III, 31062 Toulouse, France, email: {cooper|herzig|maffre|maris|regnier}@irit.fr

Proposition 2 *For $n \geq 4$, if the n-vertex graph G has $K_{2,n-2}$ as a subgraph, then any instance of Gossip-pos$_G(d)$ has a solution of length no greater than $(d+1)(n-2)$.*

Recall that Gossip$_G(d)$ denotes the version of Gossip-pos$_G(d)$ in which the goal consists of all depth-d positive epistemic fluents. We can, in fact, show that the solution plan given in the proof of Proposition 2 [3] is optimal for Gossip$_G(d)$.

Theorem 3 *The number of calls required to solve Gossip$_G(d)$ (for any graph G) is at least $(d+1)(n-2)$.*

4 One-way communications

We now study a different version of the gossip problem, denote by Directional-gossip, in which communications are one-way. Whereas a telephone call is essentially a two-way communication, e-mails and letters are essentially one-way. The result of CALL$_{i,j}$ is now that agent i shares all his knowledge with agent j but agent i receives no information from agent j. Indeed, to be consistent with communication by e-mail, we assume that after CALL$_{i,j}$, agent i does not even gain the knowledge that agent j knows the information that agent i has just sent in this call (e.g. the e-mail was not read).

Directional-gossip-pos$_G(d)$ can be solved in polynomial time: for example, on an undirected graph G, any solution plan for Gossip-pos$_G(d)$ can be converted into a solution plan for Directional-gossip-pos$_G(d)$ by replacing each two-way call by two one-way calls. What is surprising is that the exact minimum number of calls to solve Directional-gossip-pos$_G(d)$ is often much smaller than this and indeed often very close to the minimum number of calls required to solve Gossip-pos$_G(d)$. We consider, in particular, the hardest version of Directional-gossip-pos$_G(d)$, in which the aim is to establish all epistemic goals of depth d. Let Directional-gossip$_G(d)$ denote the directional gossip problem whose goal is to establish the conjunction of $K_{i_1} \dots K_{i_d} s_{i_{d+1}}$ for all $i_1, \dots, i_{d+1} \in \{1, \dots, n\}$.

In the directional version, the graph of possible communications is now a directed graph G. Let $\overline{G}$ be the graph with the same n vertices as the directed graph G but with an edge between i and j if and only if G contains the two directed edges (i,j) and (j,i). It is known that if the directed graph G is strongly connected, the minimal number of calls for Directional-gossip-pos$_G(1)$ is $2n-2$ [4]. We now generalise this to arbitrary d under an assumption about the graph $\overline{G}$.

Proposition 4 *For all $d \geq 1$, if $\overline{G}$ contains a Hamiltonian path, then any instance of Directional-gossip-pos$_G(d)$ has a solution of length no greater than $(d+1)(n-1)$.*

However, it should be pointed out that determining the existence of a Hamiltonian path in a graph is NP-complete.

The following theorem shows that the protocol given in the proof of Proposition 4 [3] is optimal even for a complete digraph G.

Theorem 5 *The number of calls required to solve Directional-gossip$_G(d)$ (for any digraph G) is at least $(d+1)(n-1)$.*

It is worth pointing out that, by Theorem 3, the optimal number of 2-way calls is only $d+1$ less than the optimal number of one-way calls and is hence independent of n, the number of agents.

5 Parallel communications

An the variant Parallel-gossip-pos$_G(d)$, we consider time steps instead of calls: in each time step each agent can only make one call but several calls can be made in parallel. Parallel-gossip$_G(d)$ is the problem of establishing all depth-d positive epistemic fluents. For Parallel-gossip$_G(1)$ on a complete graph G, if the number of agents n is even, the time taken (in number of steps) is $\lceil \log_2 n \rceil$, and if n is odd, it is $\lceil \log_2 n \rceil + 1$ [7]. We now generalise this.

Proposition 6 *For $n \geq 2$, if the n-vertex graph G has the complete bipartite graph $K_{\lceil n/2 \rceil, \lfloor n/2 \rfloor}$ as a subgraph, then any instance of Parallel-gossip-pos$_G(d)$ has a solution with $d(\lceil \log_2 n \rceil - 1) + 1$ time steps if n is even, or $d\lceil \log_2 n \rceil + 1$ time steps if n is odd.*

Determining whether a n-vertex graph G has the complete bipartite graph $K_{\lceil n/2 \rceil, \lfloor n/2 \rfloor}$ as a subgraph can be achieved in polynomial time [3]. On the other hand, it is known that deciding whether Directional-gossip(1) (the problem in which the digraph G is part of the input) can be solved in a given number of steps is NP-complete [8].

In fact, the following theorem shows that the protocol given in the proof of Proposition 6 [3] is optimal in the number of steps.

Theorem 7 *The number of steps required to solve Parallel-gossip$_G(d)$ (for any graph G) is at least $d(\lceil \log_2 n \rceil - 1) + 1$ if n is even, or $d\lceil \log_2 n \rceil + 1$ if n is odd.*

It can happen that increasing the number of secrets (and hence the number of agents) leads to less steps. Consider the concrete example of 7 or 8 agents. The number of steps decreases from $3d+1$ to $2d+1$ when the number of agents increases from 7 to 8. By adding an extra agent, we actually achieve more calls in less steps.

6 Discussion and conclusion

When we allow negative goals, the gossip problem becomes NP-complete [3]. Nonetheless, we avoid the PSPACE complexity of classical planning. The general conclusion that can be drawn is that many interesting epistemic planning problems are either solvable in polynomial time or are NP-complete, thus avoiding the PSPACE-complete complexity of planning. We consider the gossip problem to be a foundation on which to base the study of richer epistemic planning problems.

References

[1] Guillaume Aucher and Thomas Bolander, 'Undecidability in epistemic planning', in *IJCAI 2013, Proc. 23rd International Joint Conference on Artificial Intelligence*, (2013).

[2] Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frederic Maris, and Pierre Régnier, 'A simple account of multi-agent epistemic planning', in *Proc. ECAI 2016, The Hague, Netherlands*, (2016).

[3] Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frederic Maris, and Pierre Régnier, 'Simple epistemic planning: generalised gossiping', *CoRR*, **abs/1606.03244**, (2016).

[4] Frank Harary and Allen J. Schwenk, 'The communication problem on graphs and digraphs', *Journal of the Franklin Institute*, **297**(6), 491–495, (1974).

[5] Sandra M. Hedetniemi, Stephen T. Hedetniemi, and Arthur L. Liestman, 'A survey of gossiping and broadcasting in communication networks', *Networks*, **18**(4), 319–349, (1988).

[6] Andreas Herzig and Faustine Maffre, 'How to share knowledge by gossiping', in *Multi-Agent Systems and Agreement Technologies*, volume 9571 of *LNCS*, pp. 249–263. Springer, (2015).

[7] Walter Knödel, 'New gossips and telephones', *Discrete Mathematics*, **13**(1), 95, (1975).

[8] David W. Krumme, George Cybenko, and K. N. Venkataraman, 'Gossiping in minimal time', *SIAM J. Comput.*, **21**(1), 111–139, (1992).

[9] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi, *Dynamic Epistemic Logic*, Springer Publishing Company, 1st edn., 2007.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1565

A General Characterization of Model-Based Diagnosis

Gregory Provan[1]

Abstract. The Model-Based Diagnosis (MBD) framework developed by Reiter has been a strong theoretical foundation for MBD, yet is limited to models that are described in terms of logical sentences. We propose a more general framework that covers a wide range of modelling languages, ranging from AI-based languages (e.g., logic and Bayesian networks) to FDI-based languages (e.g., linear Gaussian models). We show that a graph-theoretic basis for decomposable system models can be augmented with several languages and corresponding inference algorithms based on valuation algebras.

1 A GENERAL MBD FRAMEWORK

We propose a framework for extending the Reiter MBD approach [7] by integrating several MBD and FDI approaches within a decomposable graphical framework in which the modeling language and inference are specified by a valuation algebra [6].

More formally, we define an MBD framework using a tuple $(\mathcal{G}, \mathcal{T}, \Gamma)$, where $\mathcal{G}$ is a factor graph [5], $\mathcal{T}$ is the diagnosis task, and Γ is a valuation algebra [6]. The *factor graph* $\mathcal{G}$ specifies a system topology in terms of a decomposable relation Ψ, defined over a set $\boldsymbol{V}$ of variables, such that $\Psi = \otimes_i \psi_i(\boldsymbol{V}_i)$, where ψ_i is a sub-relation, $\otimes$ is the composition operator and $\boldsymbol{V}_i \subseteq \boldsymbol{V}$. This decomposition can be mapped to a graph, e.g., a DAG for a Bayesian network (BN) or an undirected graph for a Markov network. The *diagnosis task* is given by the tuple $\mathcal{T} = (\mathcal{D}, \boldsymbol{y}, \mathcal{R})$, where $\mathcal{D}$ is the task specification; $\boldsymbol{y}$ is the required system measurement; and residual $\mathcal{R}(\Psi, \boldsymbol{y})$ indicates a discrepancy between observation $\boldsymbol{y}$ and model-derived prediction $\hat{\boldsymbol{y}}$ using some distance measure. The valuation algebra Γ specifies (1) the underlying language for the diagnosis system, and (2) the inference necessary to compute the diagnosis for the task $\mathcal{T}$, e.g., multiple-fault subset-minimal diagnosis or Most-Probable Explanation.

This decomposable representation can encode a wide range of diagnosis models, including propositional logic models, FDI dynamical systems models, as well as stochastic models (Bayesian networks, HMMs, and linear Gaussian models). Previous work, e.g., [4] has shown AI-based approaches to diagnosis [7, 2] can be described by valuation algebras. Here, we extend this to include FDI approaches based on ordinary differential equations (ODEs), and show the importance of system structure and diagnosis task in specifying the full diagnosis representation.

This framework has several outcomes. First, it enables a clear separation between models and inference (although the two are linked). Specifically, the model structure encoded as a factor graph that governs inference complexity. For example, tree-structured factor graphs are all poly-time computable. Second, the factor graph encoding of models clarifies the structural difference between AI-based approaches and FDI-based approaches.

[1] University College Cork, Cork, Ireland, email: g.provan@cs.ucc.ie. Supported by Science Foundation Ireland (SFI) Grant SFI/12/RC/2289.

2 REPRESENTING MULTIPLE MODEL TYPES

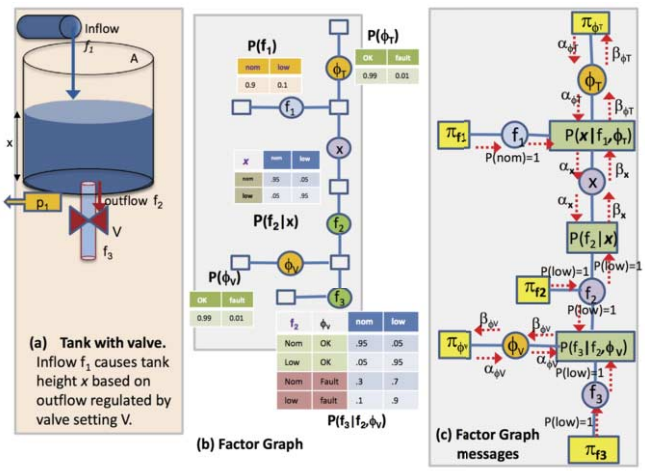


Figure 1. Bayesian network for controlled tank example.

We can represent a system (or plant) model Ψ using a factor graph, which represents the physical connectivity of Ψ in terms of a structured decomposition of Ψ into sub-relations. Consider Figure 1(a), which shows an example of a tank with a valve, where we control the level x in the tank by controlling the inflow f_1 and the valve state V. There are two possible failures in the system: (1) the tank can leak, with failure mode ϕ_T, and (2) the valve can malfunction, with failure mode ϕ_V. This example has variables $\{f_1, f_2, f_3, x, \phi_T, \phi_V\}$ and three relations: $\psi_1(f_1, \phi_T, x)$, $\psi_2(x, f_2)$, and $\psi_3(f_2, f_3, \phi_V)$. ψ_1 represents how the tank's fluid level x is governed by inflow f_1 and fault (leak) ϕ_T, ψ_2 represents how outflow is governed by fluid height x, and ψ_3 represents the valve's impact on flows f_2, f_3.

Given such a decomposition, we can represent the modelling language as a valuation over $\psi_1 \otimes \psi_2 \otimes \psi_3$. For example, if we choose a probabilistic algebra then we obtain a diagnostic BN model, for which Figure 1(b) shows the structure and valuation $P(\boldsymbol{V}) = P(f_1)P(\phi_T)P(\phi_V)P(f_1|\phi_t, x)P(x|f_2)P(f_3|f_2, \phi_V)$. $P(x|f_1, \phi_T)$ defines the conditional dependence of tank level x on the inflow f_1 and the tank fault-state ϕ_T, and (2) $P(f_3|f_2, \phi_V)$ the conditional dependence of flow from this system, f_3, on the tank outflow f_2 and the valve fault-state ϕ_V.

	Approach	Language	Structure	Task	Semi-Ring	RA	FI	Complexity
Atemporal	Reiter	Prop. Logic	DAG	Dx	$\langle\{0,1\},\wedge,\vee\rangle$	$\Psi^{\downarrow\emptyset}$	$\Psi^{\downarrow\phi}$	NP-hard
	ATMS	M-V Logic	DAG	all Dx	$\langle 2^{\mathcal{P}},\cap,\cup\rangle$	$\Psi^{\downarrow\emptyset}$	$\Psi^{\downarrow\phi}$	NP-hard
	Qualitative	Q-Relation	arbitrary	Dx	$\langle 2^{\mathcal{P}},\oslash,\odot\rangle$	$\Psi^{\downarrow\emptyset}$	$\Psi^{\downarrow\phi}$	NP-hard
	Constraint Network	Constraint	arbitrary	all Dx	$\langle 2^{\mathcal{P}},\times,+\rangle$	$\Psi^{\downarrow\emptyset}$	$\Psi^{\downarrow\phi}$	NP-hard
	BN-Posterior	PGM	DAG	$P(\boldsymbol{\phi}\|\boldsymbol{y})$	$\langle[0,1],\times,+\rangle$	$\Psi^{\downarrow\emptyset}$	$\Psi^{\downarrow\phi}$	NP-hard
	BN-MPE	PGM	DAG	MPE	$\langle[0,1],max,\times\rangle$	$\Psi^{\downarrow\emptyset}$	$\Psi^{\downarrow\phi}$	NP-hard
Temporal	DES	M-V Logic	arbitrary	all Dx	$\langle 2^{\mathcal{P}},\cap,\cup\rangle$	$\Psi^{\downarrow\emptyset}$	$\Psi^{\downarrow\phi}$	NP-hard
	DBN	PGM	DAG	$P(\boldsymbol{\phi})$	$\langle[0,1],\times,+\rangle$	$\Psi^{\downarrow\emptyset}$	$\Psi^{\downarrow\phi}$	NP-hard
	HMM	PGM	tree	$P(\boldsymbol{\phi})$	$\langle[0,1],\times,+\rangle$	$\Psi^{\downarrow\mathcal{R}}$	$\Psi^{\downarrow\phi}$	$O(n)$
	FDI	ODE	bipartite	Dx	$\langle[0,1],\times,+\rangle$	$\Psi^{\downarrow\mathcal{R}}$	$\Psi^{\downarrow\phi}$	NP-hard
	Kalman filter	PGM	tree	MPE	$\langle[0,1],\times,+\rangle$	$\Psi^{\downarrow\mathcal{R}}$	$\Psi^{\downarrow\phi}$	$O(n^3)$
	Particle filter	PGM	arbitrary	MPE	$\langle[0,1],\times,+\rangle$	$\Psi^{\downarrow\mathcal{R}}$	$\Psi^{\downarrow\phi}$	NP-hard

Table 1. Classification of Diagnosis Approaches using Generalized Approach. RA and FI denote Residual Analysis and Fault Isolation, respectively. Shaded rows denote stochastic methods, and unshaded rows denote deterministic methods. BN denotes Bayesian-network. For the language, Prop and M-V denote propositional and multivalued respectively; Q-Relation denotes Qualitative Relation; PGM refers to probabilistic graphical model, and ODE to Ordinary Differential Equation. For the task, Dx corresponds to a single diagnosis, $P(\boldsymbol{\phi})$ to the probability distribution over $\boldsymbol{\phi}$, and MPE to Most-Probable Explanation.

We perform inference in Ψ by message-passing and valuation updating [6]. Figure 1(c) shows how we can compute a diagnosis (i.e., evaluate $\phi_T,\ \phi_V$) by passing messages among the nodes starting with the control and observation settings $\mathcal{S}$. If all assignments in $\mathcal{S}$ are nominal (*nom*), the "diagnosis" is $P(\phi_T = fault) = .004$ and $P(\phi_V = fault) = .004$. If scenario $\mathcal{S}$ has control assignment f_1=*nom*, and observation of $f_2,\ f_3$ both as *low*, we obtain a diagnosis given by $P(\phi_T = fault) = .067$ and $P(\phi_V = fault) = .009$, i.e., a faulty tank is the most likely diagnosis.

We can represent several different tank models by keeping the same decomposable structure and changing the valuation. For example, we obtain a qualitative model by replacing (1) the conditional probability tables with qualitative relations, and (2) passing qualitative messages (e.g., {+,-,0} rather than discrete-valued probabilities), and using qualitative inference rather than Bayesian updating.

Table 1 summarizes the properties of several models characterized by our framework, defining the language, model structure, the underlying semi-ring, and the inference complexity. The language and task can be characterized by the valuation semi-ring (Z, O_1, O_2), which consists of a set Z and two operations (O_1, O_2) [3]. The last column of Table 1 shows the inference complexity for which the primary determinant is the topology [1]: any non-tree topology is likely to be NP-hard for computing a task requiring at least one multiple-fault diagnosis, whereas tree topologies are poly-time solvable for the majority of languages and tasks.

A **valuation** is a measure over the possible values of a set $\boldsymbol{V}$ of variables [3]. Each valuation ψ refers to a finite set of variables $d(\psi) \subseteq \boldsymbol{V}$, called its domain. Given the power set $\mathcal{P}$ of $\boldsymbol{V}$ and a set ψ of valuations with their domains in $\mathcal{P}$, we can define 3 key operations: (1) *Labeling*: $\psi \mapsto d(\psi)$, which returns the domain of each valuation; (2) *Combination*: $(\psi_1, \psi_2) \mapsto \psi_1 \otimes \psi_2$, which specifies functional composition, i.e., the aggregation of data from multiple functions; (3) *Projection*: $(\psi, V) \mapsto \psi^{\downarrow V}$ for $V \subseteq d(\psi)$, which specifies the computation of a query (set of variables) of interest.

Given an observation $\boldsymbol{y}$, we specify diagnosis within a valuation algebra as a two-step process of: (1) residual analysis (RA); and (2) fault isolation (FI).

Residual analysis: This inference depends on the type of residual. AI logic-based approaches compute RA using a consistency check, denoted $\Psi^{\downarrow\emptyset}$. FDI continuous-valued systems compute RA as $\mathcal{R} = |\hat{\boldsymbol{y}} - \boldsymbol{y}|$, where $\hat{\boldsymbol{y}}$ is the model's prediction. Residual-specific FG structure may be necessary to enable us to compute $\Psi^{\downarrow\mathcal{R}}$.

Fault isolation: Isolating a diagnosis is equivalent to projecting the marginal over the fault-mode variables $\boldsymbol{\phi}$, denoted $\Psi^{\downarrow\phi} = (\psi_1 \otimes \cdots \otimes \psi_n)^{\downarrow\boldsymbol{\Phi}}$. Diagnostic inference requires all 3 valuation operations, in particular combination and projection. The task also may change the FG structure and operations required. For example, different operations are required for computing a posterior distribution $P(\boldsymbol{\phi}|\boldsymbol{y})$ as opposed to the Most Probable Explanation (MPE).

Given an observation $\boldsymbol{y}$ and prediction $\hat{\boldsymbol{y}}$, the typical objective of a diagnosis process is to identify the system fault-state that minimises the residual vector: $\phi^* = \operatorname{argmin}_{\phi\in\Phi} \mathcal{R}(\Psi, \boldsymbol{y})$. The full paper generalizes the inference metric to define our diagnosis task as jointly minimizing the accuracy (based on $\mathcal{R}$) and the inference complexity.

3 CONCLUSION

This article has presented a general framework for MBD that integrates several approaches developed within different communities, most notably the AI and FDI communities. By characterizing MBD using the triple $(\mathcal{G}, \mathcal{T}, \Gamma)$ we show structural similarities in MBD techniques using the underlying graph $\mathcal{G}$. The valuation algebra Γ enables us to demonstrate the operations and message-passing techniques underlying the MBD approaches. As a consequence, we are able to identify similarities among MBD approaches, thereby paving the way for a more holistic approach to MBD and potential cross-pollination of MBD inference techniques.

REFERENCES

[1] Adnan Darwiche, 'Utilizing device behavior in structure-based diagnosis', in *IJCAI*, pp. 1096–1101. Citeseer, (1999).

[2] Johan De Kleer, 'An assumption-based TMS', *Artificial intelligence*, **28**(2), 127–162, (1986).

[3] J. Kohlas, *Information Algebras: Generic Structures for Inference*, Springer, Heidelberg, 2003.

[4] Jürg Kohlas, Rolf Haenni, and Serafín Moral, 'Propositional information systems', *Journal of Logic and Computation*, **9**(5), 651–681, (1999).

[5] Hans-Andrea Loeliger, 'An introduction to factor graphs', *Signal Processing Magazine, IEEE*, **21**(1), 28–41, (2004).

[6] Marc Pouly and Jürg Kohlas, *Generic Inference: A Unifying Theory for Automated Reasoning*, John Wiley & Sons, 2012.

[7] Raymond Reiter, 'A theory of diagnosis from first principles', *Artificial intelligence*, **32**(1), 57–95, (1987).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1567

On the Impact of Subproblem Orderings on Anytime AND/OR Best-First Search for Lower Bounds

William Lam and **Kalev Kask** and **Rina Dechter** [1] and **Javier Larrosa** [2]

Abstract. Best-first search can be regarded as anytime scheme for producing lower bounds on the optimal solution, a characteristic that is mostly overlooked. We explore this topic in the context of AND/OR best-first search, guided by the MBE heuristic, when solving graphical models. In that context, the impact of the secondary heuristic for subproblem ordering may be significant, especially in the anytime context. Indeed, our paper illustrates this, showing that the new concept of *bucket errors* can advise in providing effective subproblem orderings in AND/OR search.

1 INTRODUCTION

AND/OR best-first search (AOBF) is guided by two heuristic evaluation functions. The first, f_1, evaluates the potential cost of the best solution extending a current partial solution graph. The second heuristic, f_2, prioritizes the tip nodes of a current partial solution which will be expanded next. Quoting Pearl (page 54): "These two functions, serving in two different roles, provide two different types of estimates: f_1 estimates some properties of the set of solution graphs that may emanate from a given candidate base, whereas f_2 estimates the amount of information that a given node expansion may provide regarding the alleged priority of its hosting graph. Most work in search theory focus on the computation of f_1, whereas f_2 is usually chosen in an ad-hoc manner." [6]

Contributions. We explore the impact of the secondary heuristic in context of AOBF [4] for generating lower bounds in an anytime manner for *min-sum* problems over a graphical model (e.g., MAP in Markov networks [7] and weighted CSPs). Our proposed secondary heuristic for subproblem ordering is based on *bucket errors* [2]. We show empirically on three benchmarks that bucket errors provide relevant information on improving the lower bound when expanding a particular node. In particular, it can improve the anytime lower bound often, when compared to a baseline ordering. As far as we know, this is the first investigation of subproblem ordering in AOBF and of anytime best-first search.

2 BACKGROUND

Graphical Models A *graphical model* is a tuple $\mathcal{M} = (\mathbf{X}, \mathbf{D}, \mathbf{F})$, where $\mathbf{X} = \{X_i : i \in V\}$ is a set of variables indexed by a set V, $\mathbf{D} = \{D_i : i \in V\}$ is a set of finite domains of values for each X_i, and $\mathbf{F}$ is a set of discrete functions over subsets of $\mathbf{X}$. We focus on the *min-sum problem*, $C^* = \min_x \sum_{f \in F} f(x)$, which covers a wide range of reasoning problems such as MAP in graphical models or cost minimization in weighted constraint satisfaction problems.

Primal graph, pseudo-tree. The *primal graph* G of a graphical model $\mathcal{M}$ is a graph where each variable X_i is represented by a node and edges connect variables which appear in the same scope of any $f \in \mathbf{F}$. A *pseudo-tree* $\mathcal{T} = (V, E')$ of the primal graph $G = (V, E)$ is a rooted tree over the same nodes V such that every edge in $E - E'$ connects a node to an ancestor in $\mathcal{T}$.

AND/OR Search. The search space of a graphical model can be guided by the decomposition displayed in the pseudo tree, yielding an AND/OR search space. A (partial) solution of the problem is a subtree of the space known as a *(partial) solution tree*.

AND/OR best-first (AOBF) is a state-of-the-art algorithm, a variant of the AO* algorithm [5], specialized for the AND/OR search spaces over graphical models [4]. Keeping track of the explicated search space in memory, the algorithm maintains the current best partial solution tree at every step, together with an updated current best cost-estimate at each of the nodes, whose value at the root is the current best lower-bound estimate. When expanding the current partial solution tree, the algorithm orders its multiple AND leaf nodes, which corresponds to prioritizing between different subproblems and is where the f_2 heuristic comes into play. In most earlier work, the primary heuristic f_1 is used for subproblem orderings. Also, while the algorithm was considered only as a purely exact algorithm, the sequence of lower bounds generated at every step at the root node, can be seen as providing an anytime lower bound on the solution.

Mini-Bucket Elimination (MBE) [1]. The MBE scheme is an approximation which is obtained by applying the exact bucket elimination algorithm [1] to a relaxation of the problem. The MBE algorithm bounds the buckets' scopes by a parameter i, called i-bound, via partitioning the buckets into *mini-buckets*, which can be viewed as duplicating the bucket's variable. This yields lower bounds which are used as heuristics to guide AND/OR search algorithms. By design, MBE's time and space complexity is exponential in the i-bound.

Bucket Error [2]. The notion of bucket error was introduced recently as a function that captures the local error introduced by the mini-buckets. It is defined as the difference between the function-message that would have been computed in an *individual* bucket without partitioning and the message computed by MBE.

3 BUCKET ERRORS FOR ORDERING

Consider an encountered partial solution tree t that has the currently lowest f_1 but cannot be extended to an optimal solution. If $f_1(t) < C^*$, where C^* is the value of the optimal solution, the selection of which of its subproblems to expand next (the f_2 function) can

[1] University of California, Irvine, USA, {willmlam, kkask, dechter}@ics.uci.edu. Supported in part by NSF grants IIS-1526842, and IIS-1254071, the US Air Force under Contract No. FA8750-14-C-0011 under the DARPA PPAML program.
[2] UPC Barcelona Tech, Spain, larrosa@lsi.upc.edu. Supported in part by MINECO/FEDER under project TIN2015-69175-C4-3-R.

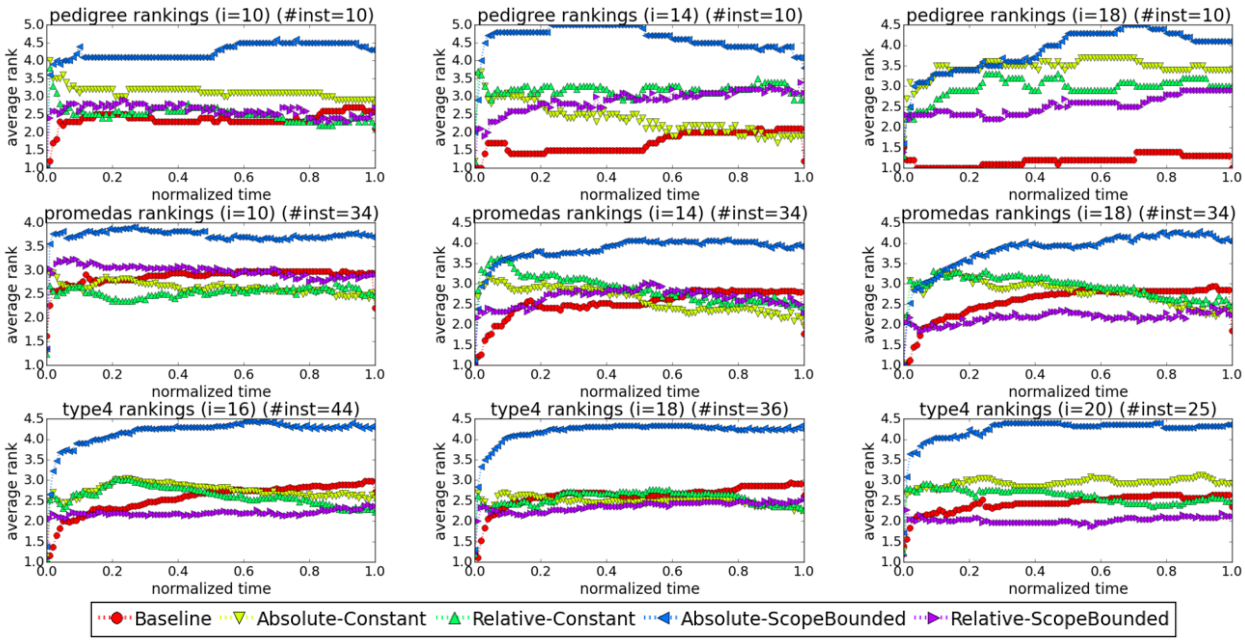


Figure 1: Average rankings over the normalized time relative to the baseline. Lower is better.

influence significantly the number of nodes extending it which will be explored eventually. It is therefore desirable to explore subproblems that yield the largest increase in its heuristic evaluation function first, to discover asap that its least cost extension is actually larger than C^*, thus avoiding the expansion of alternative branches (Best-first search will never expand a partial solution having $f_1 > C^*$.) Overall, we wish to prune partial solution graphs t that lead to sub-optimal solutions in as few node expansions as possible. Therefore, f_2 should prefer a subproblem leading to the largest increase in f_1.

With this observation, the *bucket error* is a natural choice for subproblem ordering, as it can be shown that it approximates the increase in f_1 [2]. We extend this to the notion of *subtree error*, where the bucket error of children are propagated to their ancestors to more accurately capture the mini-bucket error in an entire subproblem. As done before, the bucket-error associated with a variable can be approximated by taking absolute/relative average errors across a sample of the instantiations to the relevant bucket-error function, yielding a single constant value representing the error magnitude for each variable [2]. To generalize, we also use error functions for each variable. Since some error functions may not fit in memory, we bound the scopes of the error functions by removing variables and computing the average/relative error for each instantiation of the remaining variables. This yields 4 variants of our subproblem ordering heuristic (absolute/relative error and constant/scope-bounded error functions).

4 RESULTS AND CONCLUSION

We compare the 4 variants of our approach against the baseline subproblem ordering based on the heuristic evaluation function, f_1. We used mini-bucket elimination with moment-matching (MBE-MM) [3] for all experiments while varying the i-bound to show how differing amounts of error impact the performance. The pseudo-tree was fixed for all settings.

In **Figure 1**, we aggregated the results over each benchmark. We normalized the time scale for each instance to that of the baseline, ranked the bounds yielded by each variant across time, and aggregated across the instances by averaging. The number of instances varies with the different i-bounds since some instances run out of memory when computing the MBE heuristic with higher i-bounds.

We observe that the baseline was better for the pedigree instances, especially at the higher i-bounds. This is due to how the errors in the heuristic here are relatively low, and are therefore less informative. For the more difficult promedas and type4 benchmarks, there is more error in the heuristic, and our proposed subproblem ordering heuristics indeed improve over the baseline ordering. In particular, at the lowest i-bounds, nearly all of our methods improve over the baseline. At higher i-bounds, we see that the *Relative-ScopeBounded* variant outperforms the baseline the most and is the best performing of our 4 variants. We also improve as we move forward in time since the impact of the initial overhead of computing these heuristics becomes less relevant. Overall, our results show that the choice of the subproblem ordering heuristic impacts the performance of AOBF. Our method should be applied to any type of AND/OR best-first search, and future work includes extending to the various memory-efficient A* variants such as IDA*. For more details on the algorithms and experiments, a longer version of this paper is available here.[3]

REFERENCES

[1] Rina Dechter, *Reasoning with Probabilistic and Deterministic Graphical Models: Exact Algorithms*, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2013.
[2] Rina Dechter, Kalev Kask, William Lam, and Javier Larrosa, 'Look-ahead with mini-bucket heuristics for mpe', in *AAAI*, (2016).
[3] Alexander Ihler, Natalia Flerova, Rina Dechter, and Lars Otten, 'Join-graph based cost-shifting schemes', in *Uncertainty in Artificial Intelligence (UAI)*, 397–406, AUAI Press, Corvallis, Oregon, (August 2012).
[4] Radu Marinescu and Rina Dechter, 'Memory intensive and/or search for combinatorial optimization in graphical models', *Artif. Intell.*, **173**(16-17), 1492–1524, (2009).
[5] N. J. Nilsson, *Principles of Artificial Intelligence*, Tioga, Palo Alto, CA, 1980.
[6] J. Pearl, *Heuristics: Intelligent Search Strategies*, Addison-Wesley, 1984.
[7] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, 1988.

[3] http://www.ics.uci.edu/~dechter/publications.html

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1569

Salient Region Detection Based on the Global Contrast Combining Background Measure for Indoor Robots

Na Li[1] and Zhenhua Wang[1] and Lining Sun[1] and Guodong Chen[1, 2]

Abstract. In this paper, we propose a new method of salient region detection for indoor robots, which integrate the background distribution into the primary saliency. *Region roundness* is proposed to describe the compactness of a region to measure background distribution more robustly. In order to validate the proposed method, several influential ones are compared on the DSD dataset. The results demonstrate that the proposed approach outperforms existing methods and is useful for indoor robots.

1 INTRODUCTION

Vision system is the most important perception tool for the indoor robot. Saliency detection method is inspired by the primate-customized visual attention mechanism: select the most relevant information among the plethora of visual information [1]. Plenty of methods [2-4,6,9] perform well on the public datasets, but they cannot work well for indoor scene with complex backgrounds, several salient objects and illumination variations. Namely, the restrictions exist when these methods are applied on indoor robots.

In this paper, the work is focused on indoor robot application of detecting salient regions or objects. To address the problems mentioned above, a new salient region detection model is proposed, which comprises segmentation, primary saliency measure, background distribution measure and combination.

2 MOTHOD

The proposed method can be divided into four steps: segmentation, primary saliency measure, background distribution measure and combination.

For the first step, the graph-based RGB-D segmentation [7] (see also [5]), using depth and color feature, is introduced to keep the completeness of the salient objects.

For the second step, a color histogram (each channel in RGB color space is quantized to *N* bins) is built for each region r_i and the representative colors $\boldsymbol{c_i}:\{c_i^1, c_i^2, ..., c_i^{n_i}\}$ are picked adaptively. Herein, $\boldsymbol{c_i}$ should satisfies:

$$\begin{cases} f(c_i^1) \ge f(c_i^2) \ge ..... \ge f(c_i^{n_i}) \\ \sum_{t=1}^{n_i-1} f(c_i^t) < \varphi, \sum_{t=1}^{n_i} f(c_i^t) \ge \varphi \end{cases} \quad (1)$$

In which $f(c_i^t)$ is occurrence frequency of color bin c_i^t and φ is the minimum of frequency cumsum for each region. We set N=8 and $\varphi = 75\%$.

The primary saliency will be measured next. Salient regions should be distinctive and high-contrast compared with other regions in the image. The primary saliency is evaluated by contrasting colors of a region with all others,

$$RS(r_i) = \sum_{i \ne j} \omega_j * D_c(i, j) \quad (2)$$

In which $D_c(i, j)$ is the color distance of region r_i and r_j,

$$D_c(i, j) = \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} f(c_i^p) * f(c_j^q) * \left\| c_i^p - c_j^q \right\| \quad (3)$$

Here we utilize occurrence frequency as the weight of the color distance to emphasize high-frequency ones. In equation (2), ω_j is the weight of region r_j,

$$\omega_j = AR(r_j) * e^{-\alpha * D_s(r_i, r_j)} \quad (4)$$

In which $AR(r_j)$ is the area ratio of region r_j in the input image and $D_s(r_i, r_j)$ is the spatial distance between region r_i and r_j. α is the scaling factor and set to 3.0 through the experiment.

For the third step, we discover that generally salient regions have the property of compact sizes while background ones distribute widely and near image boundaries. Based on the hypothesis, background distribution $BD(r_i)$ is introduced to reduce false negative and false positive from the previous step,

$$BD(r_i) = \omega_{BD}(r_i) * e^{-\beta * rd(r_i)} \quad (5)$$

In which β controls the strength of $rd(r_i)$ and we use $\beta = 0.5$ in all experiments. $rd(r_i)$, called region roundness, is a new proposed measure to describe the compactness of a region and defined as:

$$rd(r_i) = rd(r_i) / L_c(r_i)^2 \quad (6)$$

In which $A(r_i)$ and $L_c(r_i)$ are area and contour length of region r_i respectively. Region roundness is usually large for salient object regions and small for background regions, and Fig 1 (Each region roundness value is displayed on it) can support our inference.

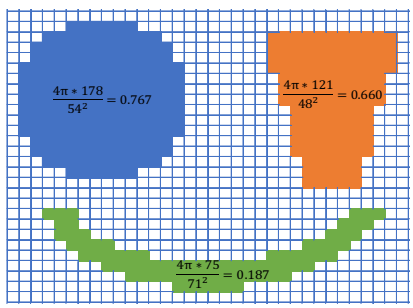


Figure 1. An illustrative example of region roundness.

[1] Robotics and Microsystem Center, Soochow University, Suzhou, China
[2] Corresponding author, email: guodongxyz@163.com

Spatial layout in images have the universality that background regions can be easily connected to image boundaries while foreground objects cannot [6]. Based on the hypothesis, we set a valid weight $\omega_{BD}(r_i)$ to influence background distribution measure,

$$\omega_{BD}(r_i) = 1 - e^{-\{BC(r_i)/\delta_{BC} + DC(r_i)/\delta_{DC}\}} \quad (7)$$

In which $DC(r_i)$ is the spatial distance between image center and region r_i, and $BC(r_i)$ is the boundary connectivity [6] of region r_i to quantify how heavily a region is connected to image boundaries. δ_{BC} and δ_{DC} control $BC(r_i)$ and $DC(r_i)$, and are set as the maximum of $BC(r_i)$ and $DC(r_i)$ separately.

For the final step, to inhibit false positive of background regions and raise false negative of salient regions, the two normalized maps are combined in a way of an exponential function:

$$Sal(r_i) = RS(r_i) * e^{-\gamma * BD(r_i)} \quad (8)$$

In which γ is the scaling factor and set to 2.0.

3 TESTS AND RESULTS

The proposed method is evaluated on DSD dataset [8] that consists of 80 RGB-D images obtained by a robot in a real-world indoor scene. Herein, we first give a visual comparison with five methods (GBVS [9], FT [2], RC [3], SF [4], RBD [6]) on this dataset. Corresponding saliency maps are shown in fig 2. Our method outperforms others visually in regard to the GT (ground truth).

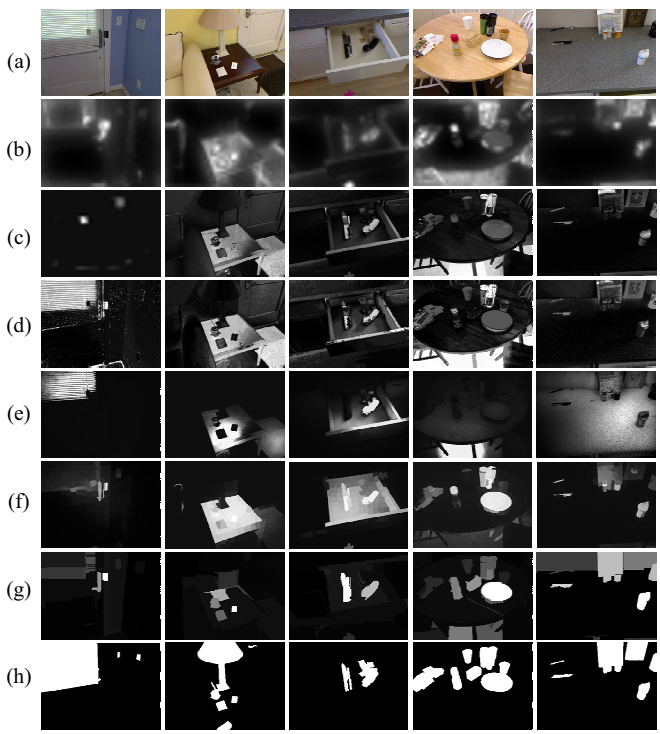


Figure 2. Visual comparison of saliency maps by different algorithms. (a) RGB image (b) GBVS (c) FT (d) RC (e) SF (f) RBD (g) Ours (h) GT

The PR (Precision and Recall) curves and MAE (Mean Absolute Error) are also introduced to quantifiably evaluate the proposed method. The PR curves are commonly used to reliably compare how well various saliency maps highlight salient regions in images. The resulting curves is shown in fig 3(a) and it clearly show that our method performs better than others. The MAE aims to measure how close a saliency map is to GT. Consistent with above results, the MAE results in fig 3(b) also support the excellence of our method.

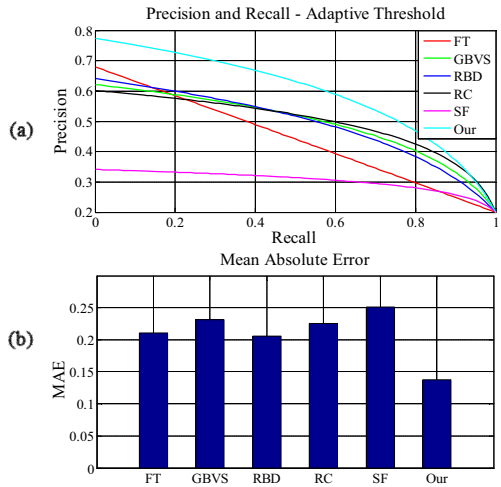


Figure 3. PR curves(a) and MAE(b) on DSD dataset for different methods.

4 CONCLUSIONS

This paper presents an effective method for indoor robots to detect salient regions or objects. However, parameter settings of segmentation weaken the application in indoor robots. We will optimize the segmentation more robustly as future work.

ACKNOWLEDGEMENTS

The research is supported by NSFC (61305118 and U1509202) and JSNSF (BK20130329).

REFERENCES

[1] M. Begum and F. Karray, 'Visual Attention for Robotic Cognition: A Survey', *IEEE Transactions on Autonomous Mental Development*, **3,** 92-105, (2011).

[2] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, 'Frequency-tuned Salient Region Detection', *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1597-1604, 2009.

[3] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, 'Global Contrast Based Salient Region Detection', *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 409-416, 2011.

[4] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, 'Saliency Filters: Contrast Based Filtering for Salient Region Detection', *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 733-740, 2012.

[5] L. Jiang, A. Koch, and A. Zell, 'Salient Regions Detection for Indoor Robots using RGB-D Data', *IEEE International Conference on Robotics and Automation (ICRA)*, 1323-1328, 2015.

[6] W. Zhu, S.Liang, Y. Wei, and J. Sun, 'Saliency Optimization from Robust Background Detection', *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2814-2821, 2014.

[7] P.F. Felzenszwalb, and D.P. Huttenlocher, 'Efficient Graph-Based Image Segmentation', *International Journal of Computer Vision*, **59**, 167-181, (2004).

[8] A. Ciptadi, T. Hermans, J.M. Rehg, 'An In Depth View of Saliency', *In Proceedings of the British Machine Vision Conference (BMVC)*, 9-13, 2013.

[9] J. Harel, C. Koch, and P. Perona. 'Graph-Based Visual Saliency', *Advances in neural information processing systems*, 545-552, 2006.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1571

Semi-Supervised Learning on an Augmented Graph with Class Labels

Nan Li[1] and **Longin Jan Latecki**[2]

Abstract. In this paper, we propose a novel graph-based method for semi-supervised learning. Our method runs a diffusion-based affinity learning algorithm on an augmented graph consisting of not only the nodes of labeled and unlabeled data but also artificial nodes representing class labels. The learned affinities between unlabeled data and class labels are used for classification. Our method achieves superior results on many standard data sets.

1 Introduction

For many real machine learning tasks, labeled data are often expensive to obtain, as they require the efforts of experienced human annotators. On the other hand, unlabeled data may be relatively easy to collect. Semi-supervised learning aims to make use of a large amount of unlabeled data, together with a small amount of labeled data, to build better classifiers.

The data of real classification tasks are often in high-dimensional spaces, where the global affinity measures may break down. That is, a small distance almost certainly indicates two high-dimensional data points are similar, but a large distance provides very little information on the discrepancy. Therefore, some classification algorithms may fail for these tasks. However, these high-dimensional data points may lie on a low-dimensional manifold. In this case, we can learn new affinities with respect to the manifold, which is described by both labeled and unlabeled data, for transductive classifications.

To address these problems, we propose a novel graph-based semi-supervised learning method.

Given a data set with a small number of labeled data, the goal is to predict the labels of the remaining unlabeled data. We first construct an augmented edge-weighted graph. In addition to the nodes of labeled and unlabeled data, we also define artificial nodes to represent the class labels. The weighted edges between nodes of labeled and unlabeled data are defined according to their local affinities, which are transformed from their distances with the Gaussian kernel. The artificial nodes representing class labels only connect to nodes of the corresponding labeled data with constant edge weights. Then we construct a reversible Markov chain from the graph. The transition probabilities, which directly reflect the local geometry, are taken as input of a diffusion-based affinity learning algorithm Tensor Product Graph Diffusion (TPGD) [2]. TPGD learns new affinities by performing diffusion on the tensor product graph obtained from the tensor product of the original graph with itself. It takes into account higher order information, but can be computed iteratively with the same computational complexity and the same amount of storage as diffusion on the original graph. Finally, the learned affinities between unlabeled data and class labels, which are derived from the new affinities of labeled and unlabeled data with respect to the underlying manifold, are used for classification.

[1] Temple University, USA, email: nan.li@temple.edu
[2] Temple University, USA, email: latecki@temple.edu

2 Related Work

Graph-based methods for semi-supervised learning have attracted wide attention due to their good performance. The Gaussian fields and harmonic functions [5] and the local and global consistency [4] are two classic methods in this category. One common feature of them is that the label information is propagated to unlabeled data following the intrinsic geometry of the data manifold, which is described by the smoothness over the weighted graph connecting the data points. The symmetric anisotropic diffusion process (SADP) for semi-supervised learning method proposed in [3] learns the affinities of unlabeled data to the labeled with a diffusion-based algorithm, which is improved in [2] as the TPGD affinity learning algorithm.

3 Our Method

Given are the data set $X = \{x_1, x_2, ..., x_l, x_{l+1}, ..., x_n\}$ and the label set $\mathcal{L} = \{1, 2, ..., c\}$. The first l data $x_i (i \leq l)$ are labeled as $y_i \in \mathcal{L}$ and the remaining data $x_u (l+1 \leq u \leq n)$ are unlabeled.

We construct an augmented edge-weighted graph $G = (V, E, \omega)$ to represent the dataset X. In addition to the nodes of labeled and unlabeled data, V also includes $|\mathcal{L}| = c$ artificial nodes representing the class labels in $\mathcal{L}$. The weighted edges between nodes of both labeled and unlabeled data are defined according to their local affinities, which are transformed from their distances in the data space with the Gaussian kernel,

$$k(x_i, x_j) = \exp\left(-\frac{(x_i - x_j)^2}{2\sigma^2}\right) \tag{1}$$

To be more robust against noise and outliers, we restrict each labeled or unlabeled data only connects to its k nearest neighbors.

For each labeled data, its node connects to the artificial node representing its class label. The weights on these edges are all initialized as 1. For unlabeled data, the initial weights on edges between their nodes and the nodes of class labels are all 0.

From graph $G = (V, E, \omega)$, we construct a reversible Markov chain on V by performing row-normalization twice on the weighted adjacency matrix W,

$$W = \begin{bmatrix} W_{CC} & W_{CD} \\ W_{DC} & W_{DD} \end{bmatrix} \tag{2}$$

where W_{CC} is a $c \times c$ zero matrix representing the weights of edges between nodes of class labels; W_{CD} ($c \times n$) and W_{DC} ($n \times c$) consist of the weights of edges between nodes of class labels and nodes of labeled or unlabeled data; W_{DD} ($n \times n$) consists of the weights of edges between pairs of nodes of labeled or unlabeled data.

We first normalize W_{DD} to a row-stochastic matrix W'_{DD}. By replacing W_{DD} with W'_{DD}, we obtain a new weight matrix W'. Then we normalize W' to a row-stochastic matrix W'' and subtract a very small constant from each entry of W'' to obtain the transition matrix P. The first row-normalization on W_{DD} ensures each row of W'_{DD} sums to 1. In W_{DC}, each of the rows corresponding to the labeled data contains only one non-zero entry, whose value is 1. Therefore, after the second row-normalization on W', all the transition probabilities from nodes of labeled data to nodes of their class labels are of the same value 0.5. By subtracting a very small constant from each entry of W'', we ensure the sum of each row of P is less than 1.

We adopt the Tensor Product Graph Diffusion (TPGD) algorithm [2] for affinity learning and take the graph G with adjacency matrix $A = P$ as the input.

The Tensor Product Graph (TPG) $\mathbb{G} = G \otimes G$ is defined as $\mathbb{G} = (V \times V, \mathbb{A})$. Thus, each node of $\mathbb{G}$ is a pair of nodes in G. The adjacency matrix of $\mathbb{G}$ is defined as $\mathbb{A} = A \otimes A$, where $\otimes$ is the Kronecker product.

The diffusion process on TPG is defined as

$$\mathbb{A}^{(t)} = \sum_{i=0}^{t} \mathbb{A}^i. \tag{3}$$

As proved in [2], since the sum of each row of A is less than 1, the process (3) converges to a fixed and nontrivial solution

$$\lim_{t\to\infty} \mathbb{A}^{(t)} = \lim_{t\to\infty} \sum_{i=0}^{t} \mathbb{A}^i = (I - \mathbb{A})^{-1}. \tag{4}$$

The matrix A^* containing the learned affinities is defined as

$$A^* = \mathrm{vec}^{-1}((I - \mathbb{A})^{-1}\, \mathrm{vec}(I)), \tag{5}$$

where I is an $n \times n$ identity matrix and vec is an operator that stacks the columns of a matrix one after the next into a column vector.

A^* can be approximated with an iterative algorithm,

$$Q^{(t+1)} = A\, Q^{(t)}\, A^T + I, \tag{6}$$

where $Q^{(1)} = A$. As proved in [2], the limit matrix $Q^* = \lim_{t\to\infty} Q^{(t)}$ converges to A^* in (5). Therefore, A^* can be obtained by iterating (6) until convergence.

Finally, according to the learned affinities in A^*, we classify each unlabeled data x_u to be with its nearest class label,

$$y_u = argmax\{a_{u\lambda} | \lambda \in \mathcal{L} = \{1, 2, ..., c\}\} \tag{7}$$

where $a_{u\lambda}$ denotes the learned affinity in A^*.

4 Experimental Results

We evaluate the performance of our method on six real benchmark data sets (BCI, Digital, g241c, g241n, USPS and TEXT) released in [1], and compare it with Gaussian Fields and Harmonic Functions (GFHF) [5], Local and Global Consistency (LGC) [4] and symmetric anisotropic diffusion process (SADP) [3]. All the six data sets are provided in two versions: one with 10 labeled data and the other one with 100 labeled data. Each version comes with 12 different labeled-unlabeled splits. The same parameter settings are used for all methods in the experiments. We use Gaussian kernel (1) to convert Euclidean distances into affinities with $\sigma = \bar{d}/3$, where $\bar{d}$ is the average distance between each data and its kth nearest neighbor. Also, we restrict that each labeled or unlabeled data only connects to its k nearest neighbors. The parameter k for these experiments is uniformly set as 12.

Table 1: Experimental results in terms of % error rate on the benchmark data sets with 10 labeled data

%	BCI	Digital	g241c	g241n	USPS	TEXT
GFHF	50.32	24.26	50.1	49.72	19.92	49.88
LGC	49.91	13.39	48.9	48.9	**15.61**	41.96
SADP	50	13.17	48.22	48.15	19.73	49.83
Ours	**49.81**	**13.05**	**47.24**	**47.8**	17.09	**35.6**

Table 2: Experimental results in terms of % error rate on the benchmark data sets with 100 labeled data

%	BCI	Digital	g241c	g241n	USPS	TEXT
GFHF	48.03	**2.17**	46.25	42.52	11.5	37.79
LGC	48.86	2.56	45.61	41.14	13.57	31.83
SADP	47.03	3.82	44.5	41.43	5.85	30.21
Ours	**46.39**	3.26	**42.38**	**39.71**	**4.85**	**28.14**

As shown in Table 1 and Table 2, our method outperforms the other methods on five data sets with 10 labeled data and five data sets with 100 labeled data. Furthermore, the advantage of our method is demonstrated more clearly by the fact that its performance is better than SADP's on all data sets. Since SADP learns affinities for classification with the same algorithm TPGD as our method does but on a regular graph with only labeled and unlabeled data, these consistent improvements prove the effectiveness of augmenting the graph with artificial nodes representing class labels.

5 Conclusion

The primary contribution of this paper is a novel graph-based semi-supervised learning method, which learns affinities on an augmented graph with class labels for classification. Experimental results on real benchmark data sets demonstrate the proposed method is effective.

REFERENCES

[1] *Semi-Supervised Learning*, eds., O. Chapelle, B. Schölkopf, and A. Zien, MIT Press, Cambridge, MA, 2006.

[2] Xingwei Yang, Lakshman Prasad, and Longin Jan Latecki, 'Affinity learning with diffusion on tensor product graph', *IEEE transactions on pattern analysis and machine intelligence*, **35**(1), 28–38, (2013).

[3] Xingwei Yang, Daniel B Szyld, and Longin Jan Latecki, 'Diffusion on a tensor product graph for semi-supervised learning and interactive image segmentation', *Advances in Imaging and Electron Physics*, **169**, 147–172, (2011).

[4] 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**(16), 321–328, (2004).

[5] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al., 'Semi-supervised learning using gaussian fields and harmonic functions', in *ICML*, volume 3, pp. 912–919, (2003).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1573

Identifying and Rewarding Subcrowds in Crowdsourcing

Siyuan Liu and **Xiuyi Fan** and **Chunyan Miao**[1]

Abstract. Identifying and rewarding truthful workers are key to the sustainability of crowdsourcing platforms. In this paper, we present a clustering based rewarding mechanism that rewards workers based on their truthfulness while accommodating the differences in workers' preferences. Experimental results show that the proposed approach can effectively discover subcrowds under various conditions, and truthful workers are better rewarded than less truthful ones.

1 INTRODUCTION

Identifying and rewarding truthful workers are key to the sustainability of crowdsourcing platforms. However, in consensus tasks [1], workers may have an unknown number of different trustful answers. To accommodate this, we propose a partitional clustering technique to identify and reward *subcrowds*, a group of workers having similar preferences and giving similar answers to the consensus tasks. Unlike many other clustering algorithms which require the prior knowledge of the number of clusters, our approach estimates the number of clusters. Thus, we assign each worker to a single cluster and reward the worker based on the distance from his/her answers to the cluster center. Experimental results show that the proposed clustering approach is able to identify subcrowds even with a significant amount of the population being untruthful. Results also show that the workers will receive more rewards if they provide more truthful answers.

2 IDENTIFYING AND REWARDING SUBCROWDS

Suppose a crowdsourced consensus task is composed of N questions. The answers from a worker w for these N questions is a vector/point $v_w = [a_w^1, a_w^2, \ldots, a_w^N]$ in an N dimension space. Thus, our goal is to classify these answer vectors into clusters.

Since the number of subcrowds is unknown, we need to firstly develop a clustering algorithm that estimates the number of clusters as well as partitioning the space into clusters. The developed algorithm is shown in Algorithm 1. We first randomly select a small subset V' from the set of collected answer vectors V as observation points in Line 1. Then for each vector v' in the subset, we calculate the distance between v' and any other vector v in V to create the distance histogram $hist$ in Line 5. We use the discrete metric and the L2 norm to measure distances for discrete and continuous answers, as given in Equation 1 and Equation 2, respectively.

$$dist(x, y) = |\{(x_i, y_i)|x_i \neq y_i, i = 1, 2, \ldots, N\}|. \quad (1)$$

$$dist(x, y) = (\sum_{i=1}^{N} |x_i - y_i|^2)^{\frac{1}{2}}. \quad (2)$$

[1] Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore, email: {syliu, xyfan, ascymiao}@ntu.edu.sg

Procedure: FindCenter(V)
Input : V, collected answer vectors;
Output : C, a set of initial centers;

```
1  Randomly select observation points V' ⊂ V;
2  C = ∅; hist = ∅;
3  foreach v' ∈ V' do
4      foreach v ∈ V do
5          hist[dist(v, v')] += 1;
6      foreach d ∈ hist do
7          if d is a local maximum in hist then
8              C_d = {v ∈ V | dist(v', v) == d};
9              C = C ∪ mean(C_d);
10 return C as initial centers.
```

Algorithm 1: Initial center estimation.

In Lines 6 and 7, we identify all local maxima in the histogram, as a local maximum indicates a dense area. In Lines 8 and 9, we identify all points in a dense area and set an initial center to be the center of this area. After we repeat the procedure for all the vectors in V', the cumulated set of initial centers are returned as C. Then we assign all points to their nearest centers in C to form clusters and move to the procedure of merging them, as follows.

For each cluster, s, with center c_s, we first find its radius r_s, defined as the distance from c_s to the farthest point in s. Then, for every two clusters s and s', if the distance between the two centers c_s and $c_{s'}$ are smaller than their radius, s and s' are then merged to form s_m. When there are no clusters to be merged, S will be returned as the resulting clusters.

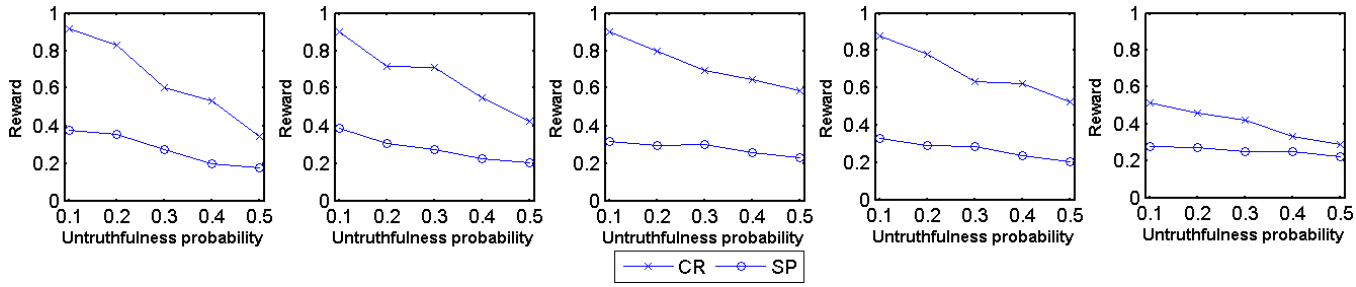
After clustering, we reward each worker based on its distance to its nearest cluster center. Namely, given a worker with answer vector v, let c_v be the cluster center that is closest to v, then the rewarding function R is:

$$R(v) = 1 - \frac{dist(v, c_v)}{N}. \quad (3)$$

The rewarding algorithm is based on the assumption that the distance $dist(v, c_v)$ increases as a worker's untruthfulness increases, which we believe is reasonable when the subcrowds share the same truthful answers and the workers behave consistently (within the subcrowd) upon providing answers for all the questions.

3 EXPERIMENTAL RESULTS

We use a discrete crowdsourcing dataset derived from [3] collected from Baidu Test to conduct experiments. Each test in the dataset is composed of 100 questions. Each question in the test contains 4 images and the task is to select the clearest one. The truthful answers can be classified into K types, where K is an integer in the range of



(a) $p = 0.1$ (b) $p = 0.2$ (c) $p = 0.3$ (d) $p = 0.4$ (e) $p = 0.5$

Figure 2. Comparison with the side-payment approach. CR is the proposed clustering reward; SP is the side-payment reward. $p = 0.1, \ldots, 0.5$ are population untruthfulness.

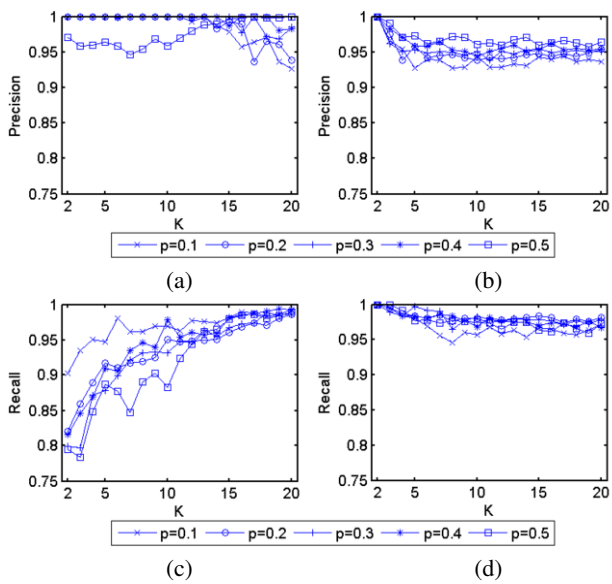


Figure 1. (a) Average precision for proposed approach; (b) Average precision for k-means; (c) Average recall for proposed approach; (d) Average recall for k-means

$[2, 20]$. For each type, we simulate 100 workers with untruthfulness p in $\{0.1, 0.2, \ldots, 0.9\}$ meaning that a worker has p chance to randomly select its answer[2]. Effectively, the dataset containing samples forming $2 - 20$ clusters with each cluster containing 100 vectors. The spread of a cluster is controlled by the untruthfulness p, e.g., for $p = 0.3$, for every answer that is in the sample, there is 30% chance that it is randomly selected. Each simulation is run 60 times and average results are presented.

We study the clustering accuracy (i.e., the accuracy in identifying subcrowds) using precision and recall[3]. To put our results into context, we compare our approach with the classic k-means with known k. Note that this gives k-means a strong edge as unlike feeding the correct k into k-means, our approach also estimates the number of clusters. Figure 1 presents the average precisions and recalls with worker untruthfulness in the range of $[0.1, 0.5]$.

As Figure 1 shows, the proposed approach achieves a higher precision result but lower recall result than k-means when the population untruthfulness is not greater than 0.5, suggesting that the proposed approach can achieve a high true positive. The recall increases with the number of ground truth clusters, suggesting that false negative becomes smaller when there are more underlying clusters.

With the proposed clustering algorithm presenting promising results, we experiment its application on rewarding workers. We reused the dataset described before with 15 clusters and an additional worker changing his untruthfulness ap from 0.1 to 0.5 to study the reward achieved by the worker with different untruthfulness in various population truthfulness environments. Figure 2 plots the normalized reward result for the additional worker when the untruthfulness of the population also changes from 0.1 to 0.5.

In these figures, the x-axis is the worker's untruthfulness, and the y-axis is the normalized reward. We compare our results with the side-payment (SP) incentive mechanism [2]. SP rewards workers by comparing two randomly selected workers. Both workers are rewarded if their answers are identical; otherwise no worker is rewarded. From Figure 2, we can see that with our clustering based approach (i.e., CR), the worker with lower untruthfulness receives more reward, regardless how (un-)truthful the entire population is.

4 CONCLUSION

Developing mechanisms to promote worker truthfulness is a key problem in crowdsourcing. In this paper, we present a clustering based approach to identify subcrowds and reward workers based on the clustering result. The approach has the following advantages: (1) it identifies subcrowds even when there exist a large amount of untruthful answers, and (2) it rewards more to workers providing more truthful answers. In the future, we will continue improving the clustering techniques and conducting a more realistic testing.

ACKNOWLEDGEMENTS

This research is supported by the National Research Foundation, Prime Ministers Office, Singapore under its IDM Futures Funding Initiative.

REFERENCES

[1] D. C. Brabham, 'Crowdsourcing as a model for problem solving: An introduction and cases', *Convergence: The International Journal of Research into New Media Technologies*, **14**(1), 75–90, (2012).
[2] R. Jurca, *Truthful Reputation Mechanisms for Online Systems*, Ph.D. dissertation, EPFL, 2007.
[3] S. Liu, C. Miao, Y. Liu, H. Yu, J. Zhang, and C. Leung, 'An incentive mechanism to elicit truthful opinions for crowdsourced multiple choice consensus tasks', in *WI-IAT*, (2015).
[4] G. Sautter and K. Böhm, 'High-throughput crowdsourcing mechanisms for complex tasks', *Social Network Analysis and Mining*, **3**(4), 873–888, (2013).

[2] It is known that untruthful workers will mostly select random answers [4].
[3] Precision is defined as True Positive / (True Positive + False Positive); recall is defined as True Positive / (True Positive + False Negative).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1575

Data Set Operations to Hide Decision Tree Rules

Dimitris Kalles[1] and **Vassilios S. Verykios**[1] and **Georgios Feretzakis**[1] and **Athanasios Papagelis**[2]

Abstract. This paper focuses on preserving the privacy of sensitive patterns when inducing decision trees. Our record augmentation approach for hiding sensitive classification rules in binary datasets is preferred over other heuristic solutions like output perturbation or cryptographic techniques since the raw data itself is readily available for public use. We describe the process and an indicative experiment using a prototype hiding tool.

1 INTRODUCTION

Privacy preserving data mining [1] tries to alleviate the problems stemming from the use of data mining algorithms to the privacy of the data subjects recorded in the data and the information or knowledge hidden in these data. An early approach dealt with the induction of decision trees from anonymized data, which had been adequately corrupted with noise to survive from privacy attacks [2]. An aspect of knowledge hiding research [3] deals with classification rule hiding [4] and aims to protect sensitive patterns arising from the application of data mining techniques.

We propose a technique to hide sensitive rules without compromising the information value of the entire dataset. After an expert selects the sensitive rules, we modify class labels at the tree node corresponding to the tail of the sensitive pattern, to eliminate the gain attained by the information metric that caused the splitting. Then, we set the values of non-class attributes, adding new instances along the path to the root, where required, to allow non-sensitive patterns to remain as intact as possible. This allows the sanitized data to be published and, even, shared with competitors, as can be the case with retail banking [5].

2 A HEURISTIC SOLUTION FOR HIDING

Figure 1 shows a baseline problem, which assumes a binary decision tree representation, with binary-valued, symbolic attributes (W, X, Y and Z) and binary classes (C_1 and C_2).Therein, hiding R_3 implies that the splitting in node Z (which is shown in the left tree) should be suppressed, hiding R_2 as well (in the right tree). A first idea to hide R_3 would be to remove from the training data all the instances of the leaf corresponding to R_3 and to retrain the tree from the resulting (reduced) dataset. However this action may incur a substantial tree restructuring. Another approach would be to turn into a new leaf the direct parent of the R_3 leaf. However, this would not modify the actual dataset, thus an adversary could recover the original tree.

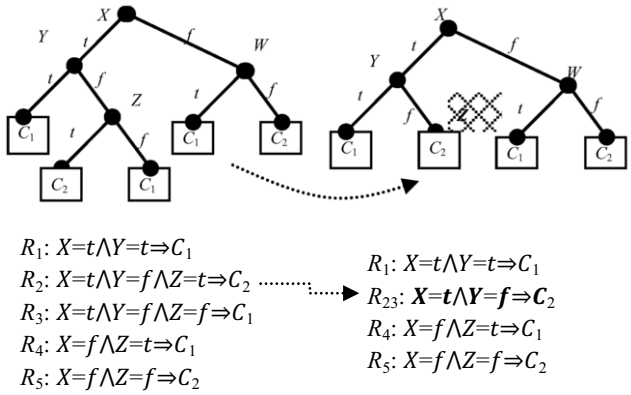


Figure 1. Hiding in a binary decision tree and the associated rule sets.

To achieve hiding by modifying the original data set in a minimal way, we may interpret "minimal" in terms of changes in the data set or in terms of whether the *sanitized* decision tree produced via hiding is syntactically close to the original one.

Since hiding at Z modifies the statistics along the path from Z to the root and splitting along this path depends on these statistics, the relative ranking of the attributes may change, if we run the same induction algorithm on the modified data set. To avoid ending up with a completely different tree, we first employ a bottom-up pass to change the class label of instances at the leaves and then to add some new instances on the path to the root, to preserve the key statistics (the *positive*/*negative* ratio) at the intermediate nodes.

We then employ a top-down pass to complete the specification of the newly added instances. Having set the values of some attributes for the newly added instances is only a partial instance specification, since we have not set those instance values for any other attribute other than the ones present in the path from the root to the node where the instance addition took place. Unspecified values must be set to ensure that currently selected attributes at all nodes do not get displaced by competing attributes; we do that by exploiting the convexity property of the information gain difference function.

Using Figure 2 as an example of the bottom-up pass, we show the original tree (in the left) with class distributions of instances across edges, and the resulting tree (in the right, with extra instances appropriately indicated).

We use the information gain as the splitting heuristic. To hide the leaf W_R we change the five positive instances to negative ones. As a result the parent node, W, becomes a one-class node with minimum (zero) entropy. All nodes located upwards to node W until the root Z also absorb the $(+5n,-5p)$ operation.

This conversion would leave X with $13n+11p$ instances. But, as its initial $8n+16p$ distribution contributed to Y's splitting attribute, A_Y, which in turn created X (and then W), we should preserve the

[1]School of Science and Technology, Hellenic Open University, Patras, Greece, email:{kalles,verykios}@eap.gr, georgios.feretzakis@ac.eap.gr
[2]Epignosis Ltd, Athens, Greece, email: papagel@efrontlearning.net

information gain of A_Y, since the entropy of a node only depends on the ratio p:n of its instance classes. Now, in the branch YX we have already added some negative instances, while we have also eliminated some positive ones. Ensuring that X's entropy will not increase can be guaranteed by not falling below the 2:1 ratio of one (*any*) class over the other. A greedy option is to add $9n$ instances, to bring the 13:11 ratio to 22:11 (now, $22n$+$11p$ instances arrive at X). These $9n$ instances propagate upwards. To preserve the 3:1 ratio of Y, we must add $47p$ instances. These additions are all shown in the right part.

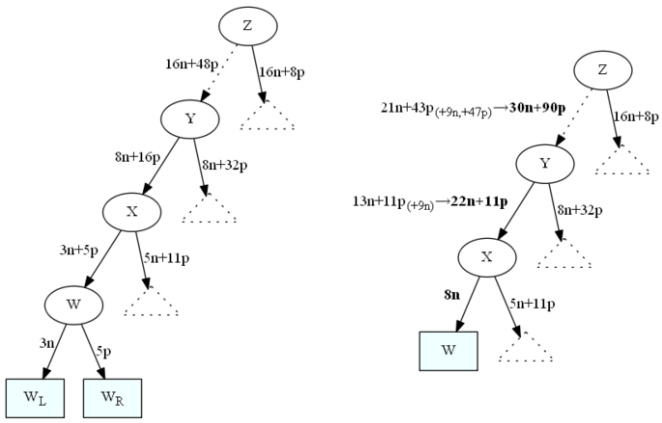


Figure 2. An example of bottom-up instance propagation.

3 A PROTOTYPE IMPLEMENTATION

A prototype system allows a user to experiment with our hiding heuristic in binary-class, binary-value data sets. For a brief experiment we used a home-grown data generator; we generated 1,000 instances for a 5-attribute problem and distributed those instances uniformly over 11 rules (Table 1).

Table 1. Rule notation: (*t*,_,_,*f*,_) means "*if* (A_1 = *t*) & (A_4 = *f*) *then* ...".

(t,t,t,t,t):p	(t,t,f,t,_):p
(t,t,t,t,f):n	(t,t,f,f,_):n
(t,t,t,f,t):p	
(t,t,t,f,f):n	

(t,f,t, _,_):p	(t,t,_, _,_):p
(t,f,f, _,_):n	
(f,f,f, _,_):n	
(f,f,t, _,_):n	

For the original decision tree we then observed, for each leaf node, the number of instances we would need to add to hide just that node. The average increase is about 67% with deep fringe nodes (longer rules) generating relatively light changes and shallow fringe nodes (shorter rules) generating larger ones.

A further experiment highlighted that large increases occur when we want to hide eminent rules. For example, when we tested our technique with a modified version of the rule set in Table 1, one that was produced by removing all rules of length 3 and 4, it turned out that the average increase over all leaves was about 73%, with some leaves accounting for a nearly 400% increase and some others for a mere 10%. When we removed all rules of length 2 and 3, the average increase was 80%. As short, eminent, rules involve fewer attributes, skewing the statistics for these attributes entails a substantial dataset modification.

Large numbers of instances to be added do not mean that the tree structure will also change a lot; we usually succeed to keep the form of the sanitized tree as close as possible to the original one. Still, the growth ratio can be quite large and this motivates the grouping of hiding requests.

4 CONCLUSIONS AND FURTHER WORK

Our heuristic allows one to specify which leaves of a decision tree should be hidden and then add instances to the original data set so that the next time one tries to build the tree, the to-be-hidden nodes will have disappeared, because the instances corresponding to those nodes will have been absorbed by neighboring ones.

Experimentation with several data sets would allow us to estimate the quality of revised versions of the heuristic, such as one that strives to keep the p:n ratio of a node itself (and not its parent), or one that attempts to remove instances instead of swapping their class labels, or still another that further relaxes the p:n ratio concept during the top-down phase by distributing all unspecified instances evenly among the left and right outgoing branch from a node and proceeding recursively to the leaves (which is the one we actually implemented). Trading off ease of implementation with performance is an obvious priority for experimental research. One also needs to look at the similarity of the original tree with the sanitized one (for example, using syntactic similarity [6], or using several measures of "minimal impact" in how one modifies decision trees [7][8][9]).

The current version (http://www.splendor.gr/trees) is a prototype of a service to accommodate hiding requests.

REFERENCES

[1] V.S. Verykios, E. Bertino, I.N. Fovino, L.P. Provenza, Y. Saygin, Y. Theodoridis: State of the Art in Privacy Preserving Data Mining. *SIGMOD Record* 33(1), 50-57 (2004)

[2] R. Agrawal, R. Srikant: Privacy-Preserving Data Mining. In *ACM SIGMOD Conference of Management of Data*, pp. 439-450 (2000)

[3] Gkoulalas-Divanis, A., Verykios, V.S.: Privacy Preserving Data Mining: How far can we go? In *Handbook of Research on Data Mining in Public and Private Sectors: Organizational and Government Applications*. Eds. A. Syvajarvi and J. Stenvall. IGI Global (2009)

[4] Delis, A., Verykios, V.S., Tsitsonis, A.: A Data Perturbation Approach to Sensitive Classification Rule Hiding. In: *25th Symposium On Applied Computing* (2010)

[5] R. Li, D. de Vries, J. Roddick: Bands of Privacy Preserving Objectives: Classification of PPDM Strategies. In: *9th Australasian Data Mining Conference*, pp. 137-151 (2011)

[6] H. Zantema, H.L.Bodlaender: Finding Small Equivalent Decision Trees is Hard. *International Journal of Foundations of Computer Science* 11(2), 343-354 (2000)

[7] D. Kalles, D.T. Morris: Efficient Incremental Induction of Decision Trees. *Machine Learning* 24(3), 231-242 (1996)

[8] D. Kalles, A.Papagelis: Stable decision trees: Using local anarchy for efficient incremental learning. *International Journal on Artificial Intelligence Tools* 9(1), 79-95 (2000)

[9] D. Kalles, A.Papagelis: Lossless fitness inheritance in genetic algorithms for decision trees. *Soft Computing* 14(9), 973-993 (2010)

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1577

Evolutionary Agent-Based Modeling of Past Societies' Organization Structure

Angelos Chliaoutakis and **Georgios Chalkiadakis** [1]

Abstract. In this work, we extend a generic *agent-based model* for simulating ancient societies, by blending, for the first time, *evolutionary game theory* with multiagent systems' *self-organization*. Our approach models the evolution of social behaviours in a population of strategically interacting agents corresponding to households in the *early Minoan* era. To this end, agents participate in repeated games by means of which they exchange utility (corresponding to resources) with others. The results of the games contribute to both the continuous re-organization of the social structure, and the progressive adoption of the most successful agent strategies. Agent population is not fixed, but fluctuates over time. The particularity of the domain necessitates that agents in our games receive *non-static* payoffs, in contrast to most games studied in the literature; and that the evolutionary dynamics are formulated via assessing the perceived *fitness* of the agents, defined in terms of how successful they are in accumulating utility. Our results show that societies of *strategic* agents that self-organize via adopting the aforementioned evolutionary approach, demonstrate a sustainability that largely matches that of self-organizing societies of more cooperative agents; and that strategic *cooperation* is in fact, in many instances, an emergent behaviour in this domain.

1 Introduction

Over the past two decades, archaeology has utilized agent-based models (ABM) for simulating ancient societies [2]. This is due to the ABMs' ability to represent individuals or societies, and encompass uncertainty inherent in archaeological theories. At the same time, incorporating ideas from multiagent systems (MAS) research in ABMs can enhance agent sophistication, and contribute on the application of strategic principles for selecting among agent behaviours [4].

To this end, a recently developed ABM with autonomous, utility-based agents explores alternative hypotheses regarding the social organization of ancient societies, by employing MAS ideas and algorithms [1]. The model incorporates different social organization paradigms and subsistence technologies (e.g., types of farming). Moreover, it employs a self-organization approach that allows the exploration of the historical social dynamics–i.e., the evolution of social relationships in a given society, while being grounded on archaeological evidence. However, the various social organization paradigms explored in that work assume a cooperative attitude on behalf of the agents. Specifically, agents were assumed to be willing to provide resources out of their stock in order to help agents in need, and such transfers drive the evolution of the social structure. In reality though, people are often driven by more individualistic instincts and exhibit more egotistic societal behaviour. Therefore, if one is to model societal transformation accurately, agent behaviour has to be analysed from a strategic perspective as well. Assuming that agent interactions are based on rational decision-making, and also influenced by their very effect on the society as a whole, then the evolution of the social dynamics can be studied via a game-theoretic approach. The "mathematics" of evolution are the subject of *evolutionary game theory (EGT)* [3], which takes an interest in the *replicator dynamics* by which strategies evolve.

In this work, we adopt such an approach for the first time, and provide an alternative "social self-organization" approach to that of [1]: here, social self-organization is driven by the interactions of *strategic* agents operating within a given social organization group, and the effects these interactions have on agent utility. As such, our ABM employs a self-organization social paradigm where the evolution of the social organization structure is driven by the interaction of agent strategies in an evolutionary game-theoretic sense [3]. This allows us to study the evolution and adaptation of strategic behaviours of agents operating in the artificial ancient community, and the effect these have on the society as a whole. We are not aware of any archaeological ABM that explicitly adopts an evolutionary game-theoretic approach. By contrast, our work here shows how EGT can be utilized within an archaeological ABM.

2 A utility-based ABM

We build on top of the ABM developed in [1] for simulating an artificial ancient society of agents evolving in a 2D grid environmental topology. The agents correspond to *households*, which are considered to be the main social unit of production in Minoan societies for the period of interest (3,100-1,100 BCE) [5], each containing up to a maximum number of *individuals* (household inhabitants). Households are *utility-based autonomous* agents who can settle, or occasionally re-settle in order to improve their utility, and cultivate in a specific environmental location.

The total number of agents in the system changes over time, as the annual levels of births and deaths is based on the amount of energy consumed by the household agent during the year. This in turn depends on the energy harvested, that is, the agent's *utility*. These rates, produce a *population growth rate* of 0.1%, when households consume adequate resources. This corresponds to estimated world-wide population growth rates during the Bronze Age.

The ABM incorporates a *self-organization* social paradigm, where agents within a settlement continuously re-assess their relations with others, and this affects the way resources are ultimately distributed among the community members, leading to "social mobility" in their relations. Self-organization gives rise, naturally, to implicit agent hi-

[1] School of Electronic and Computer Engineering, Technical University of Crete, Greece, email: {angelos, gehalk}@intelligence.tuc.gr

erarchies. Agents are assumed to be helping out agents in need (if they possess enough resources in their own storage), resulting to a continuous *targeted redistribution of resources*, so that utility flows from the more wealthy agents to those more in need within the organization, maintaining a dynamically stratified social structure. Simulation results indicate that a *heterarchical* social structure, having emerged by the continuous re-adaptation of social relations among Minoan *households*, might well have existed in the area of study.

3 An evolutionary game-theoretic extension

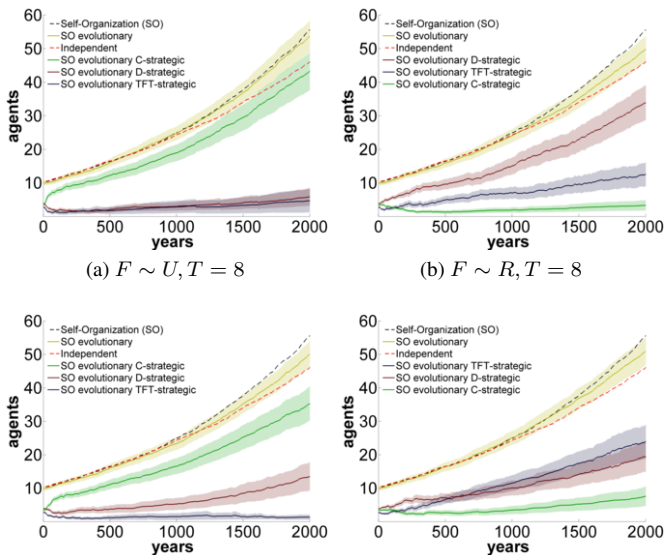
We simulate *repeated "stage games"* played by pairs of household agents, with agents belonging to the same settlement. Agents are considered as "players" in "stage games" that take place every time-step corresponding to one year. At any given time-step, a single player may be interacting at a one-on-one basis with all other agents within the settlement simultaneously. We assume a finite, but not fixed, population size (since new households are created or old ones cease to exist). Intuitively, the games model resource exchanges (utility transfers) among the households. In contrast to most matrix games studied in the literature, our agents receive *non-static payoffs* (depending on their current utility, largely acquired via working the lands). This in effect leads to an alternative model to the classic fitness-based evolution strategy selection: a strategy's reproductive success depends on *dynamic* payoffs, and thus agents using the same strategy do not necessarily receive the same payoff when interacting with other agents. We assume three different player strategic behaviours: a *cooperative* one, C, willing to share resources with another player; a *defective* one, D, refusing to share resources; and one which starts with cooperation and then behaves as the other player did in the previous game round, namely *Tit-for-Tat*, TFT. Considering these different strategic agent types as playing games against each other, we explore the evolutionary dynamics which arise. If we assume static payoffs in our "stage games", defection is the dominant strategy for any agent, and mutual defection is the only strong Nash equilibrium.

In our work, a series of (yearly) time steps during which each agent employs a specific strategy when playing in the stage games, is followed by a strategy review stage during which agents assess and possibly modify their strategies; while the results of each stage game played contribute to the continuous alteration of the social structure, which evolves as in [1], given the evolution of the differences in relative wealth among the agents. Strategy review and adoption is performed in various ways. Specifically, fitness can be evaluated with respect to solely the reward achieved in the games, or the overall utility of the strategic agent (derived from game-playing and land cultivation); while the relative success of the agent's current strategy can be assessed at either the settlement or the societal level, with respect to the average fitness of all strategies at that level, or the average fitness of the strategy itself (calculated across agents adopting this particular strategy); and the adoption of an alternative strategy can be deterministic or stochastic.

4 Simulations and results

Several scenarios were taken into account for the experimental setup, with different parameterisation. We evaluate the impact of the evolutionary self-organization social paradigm to population viability. ABM's initial settings are the same as in [1] for evaluation purposes. We adopt a uniform distribution of initial strategies, depending on agents numbers within a settlement for every simulation run. Agents review their strategy every 8 or 16 years ($T = 8$ or $T = 16$). Simulation results are *averages over 30 simulation runs* across a period of $2,000$ years. We compare the performance (in terms of population growth achieved) of strategic agents that play games and use self-organization, which we term "SO evolutionary" agents, against those that *(i)* are benevolent and self-organize, as in [1], termed "SO" agents; or *(ii)* adopt the "independent" social behaviour, trying to maximize their utility without interacting with others [1].

Overall, scenarios that sustain a higher average population of "SO evolutionary" agents, are those where agent fitness is evaluated *wrt. utility*, while agents adopt new strategies in a *stochastic* manner. Moreover, better performance is observed when agent fitness is compared to that of the *settlement* group, rather than the entire society; and especially when the performance of only the agents in the settlement that adopt the *same* strategic behaviour is taken into account, as illustrated in Fig. 1. Notably, we observe high percentages of emergent cooperative behaviour, despite this behaviour being in contrast to that prescribed by the stage game Nash equilibrium.



(a) $F \sim U, T = 8$ (b) $F \sim R, T = 8$

(c) $F \sim U, T = 16$ (d) $F \sim R, T = 16$

Figure 1: Agent population for scenarios with *stochastic* strategy review and F calculated across agents in the *settlement* that share the *same strategy*. $F \sim U$ / $F \sim R$: agent fitness function is calculated *wrt.* utility or reward accumulated in games, respectively.

REFERENCES

[1] Angelos Chliaoutakis and Georgios Chalkiadakis, 'Agent-based modeling of ancient societies and their organization structure', *Autonomous Agents and Multi-Agent Systems*, (2016).

[2] Scott Heckbert, 'Mayasim: An agent-based model of the ancient maya social-ecological system', *Journal of Artificial Societies and Social Simulation*, **16**(4), 11, (2013).

[3] Jorgen Weibull, *Evolutionary Game Theory*, MIT Press, Cambridge, 1997.

[4] Michael P. Wellman, 'Putting the agent in agent-based modeling', *Invited talk at AAMAS*, (2014).

[5] T. Whitelaw, 'House, households and community at Early Minoan Fournou Korifi: Methods and models for interpretation', in *Building Communities: House, Settlement and Society in the Aegean and Beyond*, pp. 65–76. British School at Athens Studies, (2007).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1579

Strategic Path Planning Allowing On-the-Fly Updates

Ofri Keidar and **Noa Agmon**[1]

Abstract. This work deals with the problem of strategic path planning while avoiding detection by a mobile adversary. In this problem, an evading agent is placed on a graph, where one or more nodes are defined as *safehouses*. The agent's goal is to find a path from its current location to a safehouse, while minimizing the probability of meeting a mobile adversarial agent at a node along its path (i.e., being captured). We examine several models of this problem, where each one has different assumptions on what the agents know about their opponent, all using a framework for computing node utility. We use several risk attitudes for computing the utility values, whose impact on the actual performance of the path planning algorithms is highlighted by an empirical analysis. Furthermore, we allow the agents to use information gained along their movement, in order to efficiently update their motion strategies on-the-fly. Analytic and empiric analysis show that on-the-fly updates increase the probability that our agent reaches its destination safely.

1 Introduction

The problem of path planning is one of the fundamental problems in the field of agents and robotics [5,7,9,10]. The goal in path planning is to find a sequence of world locations which allows the agent to arrive at its destination while optimizing some criteria, usually minimizing travel cost while avoiding obstacles.

In this paper we introduce a new variant of traditional path planning problem: **Stra**tegic **P**ath **P**lanning (or STRAPP, in short). In this problem, we aim at planning a path for our agent (denoted as R), while avoiding being captured by a mobile adversarial agent (denoted as C). The agents travel about a graph, representing a map of the environment (referred to as *map graph*), where a set of nodes in this graph are defined as *safehouses*. The goal of R is to arrive at one of the safehouses without being intercepted by C on its way there (*capture* it). This problem is applicable in various domains, e.g., delivering humanitarian aid into a hostile area, evading enemy forces in battlefield and even modeling better video games agents.

The problem of traveling in an environment while avoiding threats has been studied from different perspectives [2,4,6,11]. In the problem of Pursuit-Evasion [1,3,8], two rival agents move around the environment (e.g., along the edges of a graph), until the pursuer moves to the evader's location. Most research in pursuit evasion focuses on aspects concerning topology of the graph, for example, on defining properties related to graph theory of the given graph, in order to characterize graphs where the pursuer is guaranteed to capture the evader or finding minimal number of pursuers required. In our problem, however, the evader R moves to a certain destination, and does not try only evade its pursuer. Furthermore, our problem addresses strategic behavior and game theory concepts. R's path is planned based on strategies that take into account its risk attitude.

[1] Bar Ilan University, Israel, email: ofri.keidar@biu.ac.il, agmon@cs.biu.ac.il

We formally introduce the STRAPP problem, and examine different variants of it. We present a framework for computing utilities associated with each node in the graph, computed in polynomial time. This framework is used for finding solutions to the STRAPP problem in all examined variants, which differ in the level of knowledge the agents have on their opponents, and on the risk attitude they adopt.

2 STRAPP Problem Definition

The STRAPP (**Stra**tegic **P**ath **P**lanning) problem is formally defined as follows:
Given a graph $G = (V, E)$, representing a map of the environment (referred to as *map graph*), $V_G \subseteq V$ a set of goal nodes (*safehouses*) and two distinct initial positions of an agent R and an adversarial agent C, find a strategy that will maximize R's chances of reaching some node $v_g \in V_G$ without being captured by C. R is captured by C if both agents reside the same node. R wins if it reached a goal node $v_g \in V_G$ without being captured, while C wins if it captures R.

Note that the strategy may be deterministic or stochastic, and changes based on the knowledge the agents have on their opponent's strategy and location, and on the risk attitude adopted by the agents.

3 Estimating Safety of Map Graph Nodes

We define a *utility* value for each map graph node $v \in V$. The utility of $v \in V$ expresses how *safe* it is for R if moves to v, i.e., how probable it is to evade capture and reach a goal node (i.e., win). This value is derived by evaluating the game configurations (i.e., game states) where R resides at v.

More specifically, a game configuration holds current location of both R,C. The *configuration graph* $G_{conf} = (V_{conf}, E_{conf})$ is defined with a node for each configuration and an edge between any pair of consecutive game states. Configurations matching game states where R wins (*win configuration*) are given a utility of 1, while those matching game states where R loses (*lose configurations*) are given a utility value of 0. For all other configurations, the utility depends on that of its neighbors, and this value is propagated from the *terminal configurations* (i.e., win and lose configurations):
Configurations are traversed along G_{conf} in ascending order from any terminal configuration. The utility value of a configuration $\mathcal{V}$ is computed based only on its neighbors whose utility value had already been computed. Various risk attitudes can be used: risk averse (utility of $\mathcal{V}$ is minimal utility among visited neighbors), risk neutral (average utility among neighbors) or risk seeking (maximal utility among neighbors).

Once all configurations have been given a utility value, the utility for a map graph node $v \in V$ is the average utility of all configurations $\mathcal{V} \in V_{conf}$, such that $\mathcal{V} =< v, u >, u \in V$. This manner

relates to a risk neutral type of player. Other risk attitudes can be applied, e.g., risk averse (utility of a node v is the minimal utility of a configuration $\mathcal{V} =< v, u >$) or risk seeking (utility of v is the maximal one of a configuration $\mathcal{V} =< v, u >$).

4 Constructing Stochastic Motion Strategies

We assume that the adversarial agent C is capable of performing the same computations as we do. If our agent R would have followed a known deterministic movement policy, then given its initial location, C is capable of knowing its current location at each turn. Hence, there are cases where R cannot win in such model. However, if both players choose their next move according to a stochastic strategy, even if these strategies are known for each player, then there is a non-zero probability of R choosing a move that results in safe arrival to a goal node $v_g \in V_G$. As a matter of fact, in this case C benefits also from a stochastic motion pattern. A stochastic strategy P_v (i.e., a probability distribution over possible actions) is associated for each node $v \in V$ ($P_v[u]$ is the transition probability from v to u, $(v,u) \in E$), in order to employ uncertainty among the opponent.

Given the utility value of each map graph node, $P_v[u]$ ($(v,u) \in E$) is u's utility value normalized by utility values of all v's neighbors. This implies that R has a greater chance to move towards *safer* nodes, i.e., nodes with greater utility. Moving towards the neighboring node with the maximal utility value makes R's moves predictable for C, which can compute these utility values (C is assumed to have equal computational capabilities). However, such stochastic motion leads to uncertainty regarding R's assumed path.

We have proven that if C follows these stochastic strategies, as well, then an equilibrium is achieved.

5 Strategy Updates On-the-Fly

So far, strategies were computed offline, such that they aim to reduce the probability that C captures R, based on the map graph's topology. However, relying solely on graph topology, i.e., offline planning, means no reaction to new information gained while moving around the map graph. If some nodes can be observed by the other nodes (which can be considered as *viewpoints*), these nodes may provide information (or, *knowledge*) regarding an agent, e.g., tracks the agent had left behind or perhaps whether the agent currently resides at the node. When the agents follow the strategies computed as stated in Section 4, they can use these viewpoints in order to acquire information concerning their opponent's location or visited nodes, and update their strategies accordingly (each agent and its own objective). The updates are extremely efficient, i.e., linear in the number of neighbors.

The contribution of these runtime updates had been proven both theoretically and empirically.

6 Empirical evaluation

We have evaluated our path planning strategies, using different risk attitudes for computing the utility functions, and examined the effect of online strategies update. Utility functions were used in order to compute a utility value for a map graph node or a configuration node (given the values of the node's neighbors) and also used to evaluate the information obtained at a node visited by an agent.

A collection of graphs with 10 to 60 (with jumps of 5) nodes was randomly generated (40 of each size), so were the visibility edges. For each number of nodes, 10% of the nodes where randomly set as goal nodes (i.e., safehouses).

An experiment was executed for each combination of node, configuration and information utility functions for R,C. Each experiment was repeated 20 times for each graph within the graphs collection. Each time new starting locations for both agents were randomly chosen (not among the safehouses). For each graph size, combination of node, configuration and information utilities, the average winning rate of R was calculated (R*'s winning rate*).

Figure 1 shows the results when both agents update their strategies on-the-fly. R was tested with several behaviors for on-the-fly strategy updates. The curve labeled as OFFLINE is where R did not perform any on-the-fly updates (i.e., runs offline). In order to specifically examine the influence of on-the-fly updates on R's winning rate, when R did update its strategies on-the-fly, executions where R did not observe C even once were discarded.

ANOVA test with $\alpha = 0.05$, followed by post hoc test, has confirmed that R's winning rate is significantly increased with on-the-fly updates.

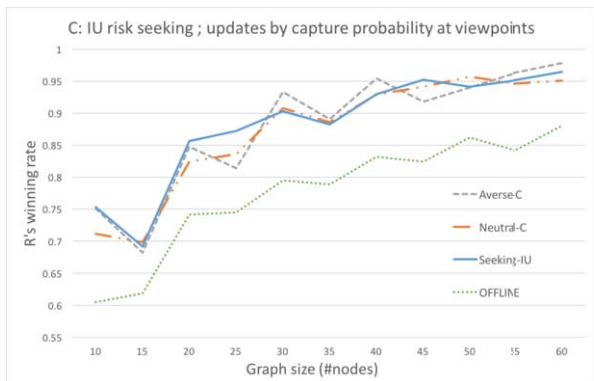


Figure 1: R*'s winning rate when both* R,C *update their strategies on-the-fly.* C*'s information utility is risk seeking, at a viewpoint* C *updates by capture probability*

REFERENCES

[1] Brian Alspach, 'Searching and sweeping graphs: a brief survey', *Le matematiche*, **59**(1, 2), 5–37, (2006).
[2] Richard B Borie, Craig A Tovey, and Sven Koenig, 'Algorithms and complexity results for pursuit-evasion problems.', in *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, volume 9, pp. 59–66, (2009).
[3] Timothy H Chung, Geoffrey A Hollinger, and Volkan Isler, 'Search and pursuit-evasion in mobile robotics', *Autonomous Robots*, **31**(4), 299–316, (2011).
[4] Fedor V Fomin and Dimitrios M Thilikos, 'An annotated bibliography on guaranteed graph searching', *Theoretical Computer Science*, **399**(3), 236–245, (2008).
[5] Jean Claude Latombe, *Robot Motion Planning*, Boston: Kluwer Academic Publishers, 1991.
[6] Mohamed Marzouqi and Ray Jarvis, 'Covert path planning for autonomous robot navigation in known environments', in *Proc. Australasian Conference on Robotics and Automation, Brisbane*. Citeseer, (2003).
[7] Andreas C Nearchou, 'Path planning of a mobile robot using genetic heuristics', *Robotica*, **16**(05), 575–588, (1998).
[8] Richard Nowakowski and Peter Winkler, 'Vertex-to-vertex pursuit in a graph', *Discrete Mathematics*, **43**(2), 235–239, (1983).
[9] Clement Petres, Yan Pailhas, Pedro Patron, Yvan Petillot, Jonathan Evans, and David Lane, 'Path planning for autonomous underwater vehicles', *IEEE Transactions on Robotics*, **23**(2), 331–341, (2007).
[10] Anthony Stentz, 'Optimal and efficient path planning for partially-known environments', in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3310–3317, (1994).
[11] Roi Yehoshua and Noa Agmon, 'Adversarial modeling in the robotic coverage problem', in *Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, (2015).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1581

Finding Diverse High-Quality Plans for Hypothesis Generation

Shirin Sohrabi and **Anton V. Riabov** and **Octavian Udrea** and **Oktie Hassanzadeh** [1]

Abstract.

In this paper, we address the problem of finding diverse high-quality plans motivated by the hypothesis generation problem. To this end, we present a planner called TK* that first efficiently solves the "top-k" cost-optimal planning problem to find k best plans, followed by clustering to produce diverse plans as cluster representatives.

1 Introduction

New applications that use AI planning to generate explanations and hypotheses have given rise to a new class of planning problems, requiring finding multiple alternative plans while minimizing the cost of those plans [11, 7, 9]. Hypotheses or explanations about a system, such as a monitored network host that could be infected by malware, are generated as candidate plans given a planning problem definition describing the sequence of observations that can be noisy, incomplete, or missing, and a domain model capturing the possible state transitions for the modeled system, as well as the many-to-many correspondence between the states and the observations. The plans must minimize both the penalties for unexplained observations and the cost of state transitions. Additionally, among those candidate plans, a small number of the most diverse plans must be selected as representatives for further analysis.

The malware detection problem or more generally the hypothesis generation problem is encoded as an AI planning problem, where the generated plans correspond to the hypotheses, and furthermore, the min-cost (or top-quality) plans correspond to the plausible hypotheses. Plausible hypotheses are those that the domain expert believes to be more plausible (more likely) compared to the other hypotheses. Plausibility can be encoded as an action cost, where higher costs indicate lower plausibility. Hence, the notion of the top-k plans maps to finding k plans with the lowest cost.

In this paper, we propose an approach for finding a set of low-cost diverse plans for hypothesis generation. To this end, we have developed a planner that first efficiently solves the "top-k" cost-optimal planning problem to find k best plans, followed by clustering to produce diverse plans as cluster representatives. Our framework is modular allowing different planning algorithms, similarity measures, and clustering algorithms in different combinations. Experiments set in hypothesis generation domains show that the top-k planning problem can be solved in time comparable to cost-optimal planning using Fast-Downward. We further empirically evaluate multiple clustering algorithms and similarity measures, and characterize the tradeoffs in choosing parameters and similarity measures.

[1] IBM T.J. Watson Research Center, Yorktown Heights, NY, USA, email: {ssohrab, riabov, udrea, hassanzadeh}@us.ibm.com

2 Top-k Planning Using K*

We define the top-k planning problem as $R = (F, A, I, \mathcal{G}, k)$, where F is a finite set of fluent symbols, A is a set of actions with non-negative costs, I is a clause over F defining the initial state, $\mathcal{G}$ is a clause over F defining the goal state, and k is the number of plans to find. Let $R' = (F, A, I, \mathcal{G})$ be the cost optimal planning problem with n valid plans. The set of plans $\Pi = \{\alpha_1, ..., \alpha_m\}$, where $m = k$ if $k \leq n$, $m = n$ otherwise, is the solution to R if and only if each $\alpha_i \in \Pi$ is a plan for the cost-optimal planning problem R' and there does not exist a plan α' for R', $\alpha' \notin \Pi$, and a plan $\alpha_i \in \Pi$ such that $cost(\alpha') < cost(\alpha_i)$. When $k > n$, Π contains all n valid plans, otherwise it contains k plans. Π can contain both optimal plans and sub-optimal plans, and for each plan in Π all valid plans of lower cost are in Π. If $\Pi \neq \emptyset$, it contains at least one optimal plan.

To solve the top-k planning problem, R, we will apply the k shortest path algorithm, K*, to find the top-k plans. K shortest paths problem is an extension of the shortest path problem where in addition to finding one shortest path, we need to find a set of paths that represent the k shortest paths. K shortest path problem is defined as $Q = (G, s, t, k)$, where $G = (V, E)$ is a graph with a finite set of nodes V and a finite set of edges E, s is the source node, t is the destination node, and k is the number of shortest paths to find. The K^* algorithm [1] is an improved variant of the Eppstein's k shortest paths algorithm [2] (we refer to as EA). The EA algorithm constructs a complex data structure called *path graph* $P(G)$ that stores the all paths in G, where each node in $P(G)$ represents a sidetrack edge. This is followed by the use of Dijkstra search on $P(G)$ to extract the k shortest paths. The major bottleneck of the EA algorithm is the construction of the complete state transition graph, which may include a huge number of states that are very far away from the goal.

In short, the K^* algorithm works as follows. The first step is to apply a forward A* search to construct a portion of graph G. The second step is suspending A* search, updating $P(G)$ similarly to EA, to include nodes and sidetracks discovered by A*, applying Dijkstra to $P(G)$ to extract solution paths, and resuming the A* search. The use of A* search to dynamically expand G enables the use of heuristic search and also allows extraction of the solution paths before G is fully explored. While K^* algorithm has the same worst-case complexity as the EA algorithm, it has better performance in practice because unlike the EA algorithm, K^* does not require the graph G to be completely defined when the search starts.

Our planner, *TK**, applies K^* to search in state space, with dynamic grounding of actions, similarly to how Fast-Downward and other planners apply A^*. The K^* scheduling condition is evaluated by comparing the state of A* and Dijkstra searches, as defined in K^* algorithm. It determines whether new links must be added to G be-

fore resuming Dijkstra search on updated $P(G)$. There is no separate grounding stage, since actions are ground at the same time when they are applied during A* search. The amount of A* expansion required before resuming Dijkstra (in our implementation, 20%) controls the efficiency tradeoff, and 20% is the same value that was used in experiments in the original K^* paper [1]. Soundness and completeness of *TK** follows directly from the soundness and completeness of the K^* algorithm. For further details of the *TK** planner see [10].

3 Finding Diverse Plans via Clustering

In practice, many of the generated top-k plans are only slightly different from each other. That is, they do seem to be duplicates of each other, except for one or more states or actions that are different. To consolidate similar plans produced by the top-k planner, we compute a similarity score between plans and apply three clustering algorithms that create clusters of plans where each cluster is disjoint and each plan belongs to only one cluster. We then may choose to present only the cluster representatives from a subset of these clusters to the user or to the automated system for further investigation.

Finding if two plans are similar has been studied mainly under plan stability for replanning and finding diverse plans (e.g., [3, 6]). Two plans can be compared based on their actions, states, or causal links. We also consider comparing plans based on their costs or their final states. Each similarity measure assigns a number between 0 (unrelated) or 1 (same). Two plans are said to be similar if their similarity score is above a predefined threshold θ. The similarity measures can be used individually or be combined using a weighted average.

We have implemented several similarity measures including Generalized Edit Similarity (GES) and Jaccard Similarity, an inverse of the plan distance from [6]. An important desired property of GES is that it not only considers the similarity between sequences, but also considers the similarity between tokens. Therefore, we are able to use any extra domain-dependent knowledge at hand about the relationship between, for example, actions to determine if two plans belong to the same cluster. This allows further semantic information to be included in similarity calculations.

We have also implemented three clustering algorithms: Center-Link, Single-Link, and Average-Link [10]. Each of these algorithms require visiting each plan only once in order to decide the cluster they belong to; however, depending on which algorithm is used, the plans are compared to representative element of the cluster or all plans.

4 Experimental Evaluation

We used both manually crafted and random problems to create our evaluation benchmark. Our problems are based on the hypothesis generation application described by Sohrabi et al. [11]. This application is a good example of a challenging top-k planning problem, and generated problems typically have a very large number of possible plans with different costs. We report a summary of our evaluations.

Top-k Planning Performance We compare the performance of our top-k planner, *TK**, with k=50 and k=1000 to Gamer [5] (Gamer 2014 version) and Fast-Downward [4] (2015 version, with A^*). Both find a single cost-optimal plan, which is equivalent to k=1. Overall, our results showed that *TK** is very efficient at finding top-k plans, and in our implementation and our set of problems performs at least as fast or faster than Fast-Downward and Gamer, which is essential for use in applications. Due to soundness and completeness of K^*, *TK** is guaranteed to produce top-k plans and that was confirmed in our experiments. Overall, these experiment results support our claim that top-k problems can be solved just efficiently as cost-optimal ones, at least within a certain class of planning domains.

Evaluation of Clusters We evaluate the different clustering algorithms and similarity measures we used. Our results show that Center-Link algorithm is the best algorithm with respect to time as fewer number of similarity comparisons is performed since each plan is only compared to the representatives. Average-Link produces more clusters compared to the other two. As the threshold increases, the number of clusters also increases for all algorithms. With respect to stability and uniqueness measures [8], the results does not show a superior clustering algorithm. Furthermore, the results show that grouping based on cost or last state may be fastest but these similarity measures give the worst results with respect to stability and uniqueness. On the other hand, using Jaccard produces most diverse plans with respect to uniqueness and GES also produces most diverse plans with respect to stability.

Comparison With Diverse Planners We selected two representative diverse planners, LPG-d [6] (with d=0.1) and Div (Multi-queue A^* MQA_{TD}) [8], and compared to our implementation that included top-k and Average-link clustering, using Jaccard similarity. We averaged over 5 instances of each size and had a 30 minutes time limit. We measure time in seconds, shown under T, plan diversity by stability, S, and by uniqueness, U, using formula from [8]. Subset of the results are shown in the following table. The results show that Div places greater emphasis on plan cost but sometimes produces multiple copies of the same plan, resulting in poor diversity. LPG-d produced diverse plans but with higher average cost. Our approach, the top-k plus clustering, produces the lowest average cost with somewhat lower diversity compared to LPG-d.

	Top-k + Average Link				**LPG-d**				**Div**			
Prob	**T**	**Cost**	**S**	**U**	**T**	**Cost**	**S**	**U**	**T**	**Cost**	**S**	**U**
1	1	1502	0.51	1	1	3513	0.80	1	1	1789	0.36	0.37
2	1	1586	0.41	0.99	59	8426	0.84	1	1	3861	0.44	0.54
3	3	1492	0.20	0.99	384	16520	0.87	1	1	7262	0.46	0.53

REFERENCES

[1] Husain Aljazzar and Stefan Leue, 'K*: A heuristic search algorithm for finding the k shortest paths', *AIJ*, **175**(18), 2129–2154, (2011).

[2] David Eppstein, 'Finding the k shortest paths', *SIAM Journal on Computing*, **28**(2), 652–673, (1998).

[3] Maria Fox, Alfonso Gerevini, Derek Long, and Ivan Serina, 'Plan stability: Replanning versus plan repair', in *ICAPS*, pp. 212–221, (2006).

[4] Malte Helmert, 'The Fast Downward planning system', *JAIR*, **26**, 191–246, (2006).

[5] Peter Kissmann, Stefan Edelkamp, and Jörg Hoffmann, 'Gamer and Dynamic-Gamer symbolic search at IPC 2014', in *IPC-2014*, (2014).

[6] Tuan Nguyen, Minh Do, Alfonso Gerevini, Ivan Serina, Biplav Srivastava, and Subbarao Kambhampati, 'Generating diverse plans to handle unknown and partially known user preferences', *AIJ*, **190**, 1–31, (2012).

[7] Anton V. Riabov, Shirin Sohrabi, Daby M. Sow, Deepak S. Turaga, Octavian Udrea, and Long H. Vu, 'Planning-based reasoning for automated large-scale data analysis', in *ICAPS*, pp. 282–290, (2015).

[8] Mark Roberts, Adele E. Howe, and Indrajit Ray, 'Evaluating diversity in classical planning', in *ICAPS*, pp. 253–261, (2014).

[9] Shirin Sohrabi, Anton Riabov, and Octavian Udrea, 'Plan recognition as planning revisited', in *IJCAI*, (2016).

[10] Shirin Sohrabi, Anton Riabov, Octavian Udrea, and Oktie Hassanzadeh, 'Finding diverse high-quality plans for hypothesis generation', in *MRC workshop at ECAI*, (2016).

[11] Shirin Sohrabi, Octavian Udrea, and Anton Riabov, 'Hypothesis exploration for malware detection using planning', in *AAAI*, (2013).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1583

Using a Deep Understanding of Network Activities for Network Vulnerability Assessment

Mona Lange[1] and **Felix Kuhr**[2] and **Ralf Möller**[3]

Abstract. In data-communication networks, network reliability is of great concern to both network operators and customers. Therefore, network operators want to determine what services could be affected by software vulnerabilities being exploited that are present within their data-communication network. To determine what services could be affected by a software vulnerability being exploited, it is fundamentally important to know the ongoing tasks in a network. A particular task may depend on multiple network services, spanning many network devices. Unfortunately, dependency details are often not documented and are difficult to discover by relying on human expert knowledge. In monitored networks huge amounts of data are available and by applying data mining techniques, we are able to extract information of ongoing network activities. From a data mining perspective, we are interested to test the potential of applying data mining techniques to real-life applications.

1 Introduction

Over the last few years, approximately 2500 software vulnerabilities were discovered every year [13]. The United States intelligence community has identified malicious actors exploiting cyberspace as a top national security threat [3]. Furthermore, IBM's 2015 cyber security intelligence index reveals that approximately half of all cyber attacks originate from within a company's own network [4]. Hence, network devices that are not connected to the internet also have to be considered as potential entry points for cyber attacks. Due to the large number of software vulnerabilities and the security threat they impose, understanding their impact on a monitored network has become an important objective. So network administrators are faced with the challenge of assessing the security impact of vulnerabilities on the network in order to choose appropriate mitigation actions.

For deriving how susceptible a network is to attackers exploiting software vulnerabilities, it is essential to understand what ongoing network activities could potentially be affected by a cyber attack. A network is built with a higher purpose or mission in mind and this mission leads to interactions of network devices and applications causing network dependencies. A monitored infrastructure's mission can be derived through human labor, however missions are subject to frequent change and often knowledge of how an activity links to network devices and applications is not available. This is why an automatic network service dependency methodology called Mission Oriented Network Analysis (MONA) [9] was introduced to derive these missions as network activity patterns. MONA was compared to three state of the art network service dependency discovery methodologies: NSDMiner [11], Sherlock [1] and Orion [2]. NSDMiner addresses the same problem of network service dependency for network stability and automatic manageability. Sherlock is another approach, which learns an inference graph of network service dependency based on co-occurrence within network traffic. A well-known approach is called Orion, which was developed to use spike detection analysis in the delay distribution of flow pairs to infer dependencies. MONA was compared via F-measures to all these state of the art methodologies and was shown to have a better performance. Current network vulnerability approaches [10] focus on identifying critical nodes in a network without focusing on the impact of software vulnerabilities. Even though, software vulnerabilities can be remotely exploitable and sometimes even exploits are readily available online, network vulnerability assessment currently does not take currently present known vulnerabilities into account.

Developing a deeper understanding of network activities allows network vulnerability assessment to analyze what network services would be potentially be affected by a software vulnerability that was detected in a monitored network. Knowing what network activities would be affected by a software vulnerability being exploited, supports network operators in developing a deeper understanding on how their network is affected by software vulnerabilities.

2 Network Vulnerability Assessment

Network vulnerability assessment consists of two parts: detecting present software vulnerabilities in a monitored network and analyzing a network's sensitivity to particular software vulnerabilities. In a monitored network, vulnerability scanners detect present software vulnerabilities. According to the ISO 27005 standard, a vulnerability is a "weakness of an asset or group of assets that can be exploited by one or more threats" [7]. Whereas an asset is defined by ISO13355 ISO/IEC TR13355-1 [6] as being "anything that can have value to the organization, its business operations and their continuity, including information resources that support the organization's mission". The non-profit organization MITRE since 1999 defines common Vulnerabilities and Exposures (CVE) identifiers for software vulnerabilities [8]. The purpose of vulnerability scanning is to identify all software vulnerabilities, which can be linked to a monitored data-communication network.

2.1 Vulnerability scanning

Vulnerability scanning a data-communication network is the process of assessing whether software vulnerabilities can be linked to monitored network devices. Software vulnerabilities can be linked to operating systems, software or firmware [5, 12]. Network vulnerability

[1] Universität zu Lübeck, Germany, email: lange@ifis.uni-luebeck.de
[2] Hamburg University of Technology, Germany, email: felix.kuhr@tuhh.de
[3] Universität zu Lübeck, Germany, email: moeller@uni-luebeck.de

analysis helps network operators to verify whether a software vulnerability linked to an application within the monitored network might endanger ongoing network activities. In the following we rely on the network model introduced in [9]. Vulnerability scanning provides us with a mapping function $SVULN$, which links CVE identifiers $cveId_j$ to network services in a monitored data-communication network. Such that we are able to associate a network service $s_i \in S$

$$SVULN : s_i \rightarrow cveId_j \tag{1}$$

with a $cveId_j$. Given that an operating system of network device d_j is affected by a vulnerability $cveId_i$

$$SVULN : HOSTS(d_j) \rightarrow cveId_i, \tag{2}$$

all hosted network services are linked to the vulnerability. Hence, we assume that an affected operating system will lead to application hosted by that network device being compromised. A software vulnerability with confidentiality impact signifies the threat of information disclosure, in comparison a vulnerability with an integrity impact signifies the threat of data modification and a vulnerability with availability impact could lead to performance degradation. As network activities often span multiple network services for a higher mission, not only network services directly linked to a vulnerability could be affected by an attacker exploiting this vulnerability. All network services relying on requests or responses from a network service linked to a vulnerability with a confidentiality, integrity or availability impact could also be affected.

Consider a vulnerability with a confidentiality impact. Given that a network service is linked to this vulnerability, all information provided by other network services could be leaked as well. Hence, these network services would also be affected by this vulnerability. A network service, which is linked to a vulnerability with an integrity threat, implies that requests sent from this network service could potentially be modified. Similarly, a network service relying on information from another network service, which is linked to a vulnerability with an availability impact, would also be affected by performance degradation of this vulnerability. The set of affected network services AS, which are directly affected by detected software vulnerabilities is defined as

$$AS = SCC\left((\forall s_i \in S : SVULN(s_i), map(asSet, SDEP))\right), \tag{3}$$

where SCC denotes the strongly connected components of the hypergraph given as parameter ($asSet$ maps a tuple into a set of components).

3 Motivating Example

The disaster recovery site of an energy distribution network, provided an Italian water and energy distribution company, was available for non-invasive experimentation. Based on this network, we are able to collect and analyze real-life network traffic and also scan the network for present software vulnerabilities. Figure 1 shows all network service dependencies detected by MONA. These network service dependencies were considered complete and correctly identified by network operators.

Figure 2 shows the result of network dependency based vulnerability assessment for software vulnerabilities CVE-2007-5423 and CVE-2010-2075 that were detected via network scanning on network device mferp2. Both software vulnerabilities can be exploited by automated code. CVE-2007-5423 is a vulnerability that allows remote attackers to execute arbitrary code in TikiWiki 1.9.8

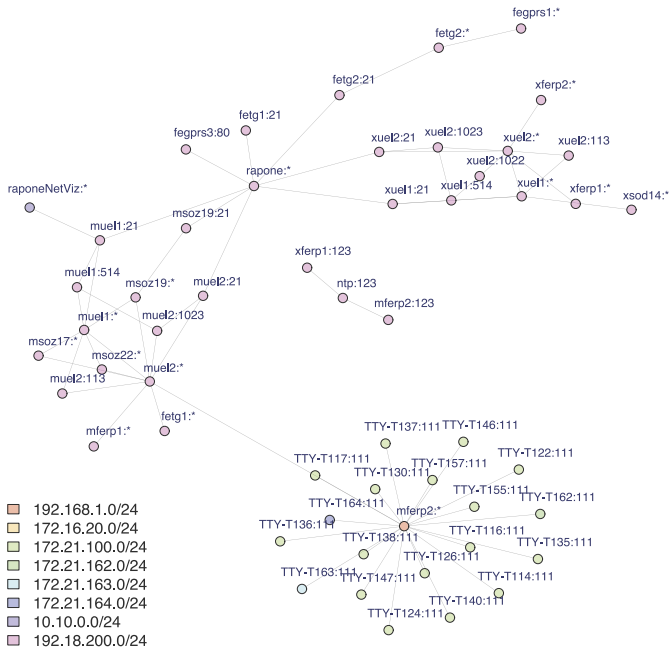


Figure 1: Network service dependency analysis in an energy distribution network.

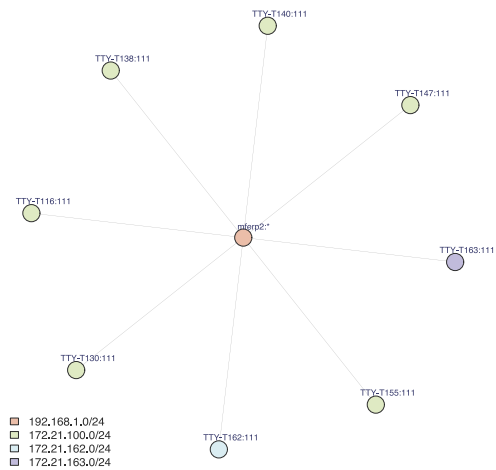


Figure 2: Network dependency based vulnerability assessment for vulnerabilities CVE-2007-5423 and CVE-2010-2075, who were detected on mferp2.

and CVE-2010-2075 is an unauthorized-access vulnerability due to a backdoor in UnrealIRCd 3.2.8.1. TTY-T[116-163] are remote terminal units of substations, which are dependent on requests from the front end server mferp2. Hence, network based vulnerability assessment concludes that TTY-T[116-163] are affected by CVE-2007-5423 and CVE-2010-2075, which were detected on mferp2.

4 Conclusion

We have introduced a novel network based vulnerability analysis approach. Network service dependency analysis allows the automatic detection of ongoing network activities. Based on automatically detected network service dependencies, we are able to link exploitable software vulnerabilities to ongoing network activities. The proposed framework is fully automated and is able to integrate vulnerability specification from the bug-reporting community and helps network operators develop a deeper understanding on how networks are affected by software vulnerabilities.

Acknowledgments

This work was partly supported by the Seventh Framework Programme (FP7) of the European Commission as part of the PANOPTESEC integrated research project (GA 610416).

REFERENCES

[1] Paramvir Bahl, Ranveer Chandra, Albert Greenberg, Srikanth Kandula, David A Maltz, and Ming Zhang, 'Towards highly reliable enterprise network services via inference of multi-level dependencies', in *ACM SIGCOMM Computer Communication Review*, volume 37, pp. 13–24. ACM, (2007).

[2] Xu Chen, Ming Zhang, Zhuoqing Morley Mao, and Paramvir Bahl, 'Automating network application dependency discovery: Experiences, limitations, and new solutions.', in *OSDI*, volume 8, pp. 117–130, (2008).

[3] James R. Clapper. Statement for the record, worldwide threat assessment of the us intelligence community. http://www.dni.gov/index.php/newsroom/testimonies/209-congressional-testimonies-2015/1174-statement-for-the-record-worldwide-threat-assessment-of-the-u-s-ic-before-the-sasc, 2014.

[4] IBM Corporation. 2015 cyber security intelligence index, july 2015.

[5] Bhadreshsinh G Gohil, Rishi K Pathak, and Axaykumar A Patel, 'Federated network security administration framework', (2013).

[6] ISO ISO and IEC Std, *ISO/IEC 13335-1: 2004*, 2004.

[7] ISO ISO and IEC Std, 'Iso 27005: 2011', *Information technology–Security techniques–Information security risk management. ISO*, (2011).

[8] MITRE. Common vulnerabilities and exposures. https://cve.mitre.org/, 2000.

[9] Ralf Möller Mona Lange, 'Time Series Data Mining for Network Service Dependency Analysis', in *The 9th International Conference on Computational Intelligence in Security for Information Systems*. Springer International Publishing, (2016).

[10] Alan T Murray, 'An overview of network vulnerability modeling approaches', *GeoJournal*, **78**(2), 209–221, (2013).

[11] Arun Natarajan, Peng Ning, Yao Liu, Sushil Jajodia, and Steve E Hutchinson, *NSDMiner: Automated discovery of network service dependencies*, IEEE, 2012.

[12] The Government of the Hong Kong Special Administrative Region. An overview of vulnerability scanners. http://www.infosec.gov.hk/english/technical/files/vulnerability.pdf, 2008.

[13] TechTarget. Growing threats make security vulnerability management essential. http://searchsecurity.techtarget.com/video/Growing-threats-make-security-vulnerability-management-essential, 2015.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1586

All-Transfer Learning for Deep Neural Networks and Its Application to Sepsis Classification

Yoshihide Sawada[1] and **Yoshikuni Sato**[1] and **Toru Nakada**[1] and **Kei Ujimoto**[2] and **Nobuhiro Hayashi**[2]

Abstract. In this article, we propose a transfer learning method for deep neural networks (DNNs). Deep learning has been widely used in many applications. However, applying deep learning is problematic when a large amount of training data are not available. One of the conventional methods for solving this problem is transfer learning for DNNs. In the field of image recognition, state-of-the-art transfer learning methods for DNNs re-use parameters trained on source domain data except for the output layer. However, this method may result in poor classification performance when the amount of target domain data is significantly small. To address this problem, we propose a method called All-Transfer Deep Learning, which enables the transfer of all parameters of a DNN. With this method, we can compute the relationship between the source and target labels by the source domain knowledge. We applied our method to actual two-dimensional electrophoresis image (TDEI) classification for determining if an individual suffers from sepsis; the first attempt to apply a classification approach to TDEIs for proteomics, which has attracted considerable attention as an extension beyond genomics. The results suggest that our proposed method outperforms conventional transfer learning methods for DNNs.

1 Introduction

In recent years, high recognition performance of systemic inflammatory response syndrome (SIRS) is an important task in clinical sites. Especially, sepsis, which is a type of SIRS leading to septic shock, affects many people around the world with a mortality rate of approximately 25%.

One of the diagnostic methods to decide sepsis or not is to use two-dimensional electrophoresis images (TDEIs, Figure 1) of proteomics, which is currently attracting considerable attention in the biological field as the next step beyond genomics. In this article, we conduct an automatic sepsis/non-sepsis classification from TDEIs.

Normally, TDEIs are analyzed for detection of specific spot corresponding to a protein as a bio-marker, using computer assistance. However, many diseases such as sepsis are multifactorial ones which cause minute changes at many spots and unexpected spots in some cases. Therefore, a classification approach by machine learning is required from clinical sites. However, few articles have been devoted to research pertaining to TDEIs because collecting TDEIs is difficult due to low-throughput data acquisition. This clearly indicates the need for the classification approach for a significantly small amount of training data is increasing.

In this article, we propose a transfer learning method that re-uses all parameters of the deep neural network (DNN) trained on the source domain data, and which we have named *All Transfer Deep Learning* (*ATDL*). ATDL evaluates the relationship between the source and the target labels on the basis of the source domain knowledge. By using the relationship, ATDL regularizes all layers including the output layer. This means that it can reduce the risk of falling into the local-optimal solution caused by the randomness of the initial values of output layer. Therefore, we belive that this method is effective especially when the amount of target domain data is significantly small, such as medical image recognition field.

Experimental results show that the classification accuracy is improved by 4 percentage points compared to the conventional transfer learning methods for DNNs, and the classification accuracy of over 90% was achieved.

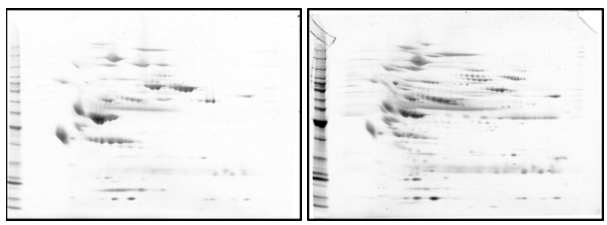


Figure 1. Examples of TDEIs. Left: sepsis, right: non-sepsis. X- and Y-axes represent degrees of molecular weights and isoelectric points, respectively, and black regions represent protein spots.

2 All Transfer Deep Learning

The outline of the ATDL training process is shown in Fig. 2. Firstly, ATDL trains the DNN, $\mathfrak{D}^s$, to solve the task of the source domain (Fig. 2(A)). In this work, we construct $\mathfrak{D}^s$ on the basis of standard stacked denoising autoencoders (SdA). Secondly, ATDL computes the output vectors of each target vector by inputting them into the DNN trained on the source domain data (Fig. 2(B)). Then, ATDL computes *relation vectors* of each target label. A relation vector denotes a vector representing the relationship between the source and the target labels on the basis of the source domain knowledge, $\mathfrak{D}^s$, by using the output vectors $\boldsymbol{v}_o(.)$ as follows.

$$\boldsymbol{r}_l = \frac{1}{N^t(l)} \sum_{j}^{N^t(l)} \boldsymbol{v}_o(\boldsymbol{x}^t_{l,j}), \tag{1}$$

where $\boldsymbol{x}^t_{l,j} \in \mathbb{R}^{D_x}$ and $N^t(l)$ are the j-th target domain vector and the amount of target domain data corresponding to the l-th target label. Finally, we fine-tune all parameters in such a way that

[1] Advanced Research Division, Panasonic Corporation
[2] Department of Life Science, Tokyo Institute of Technology

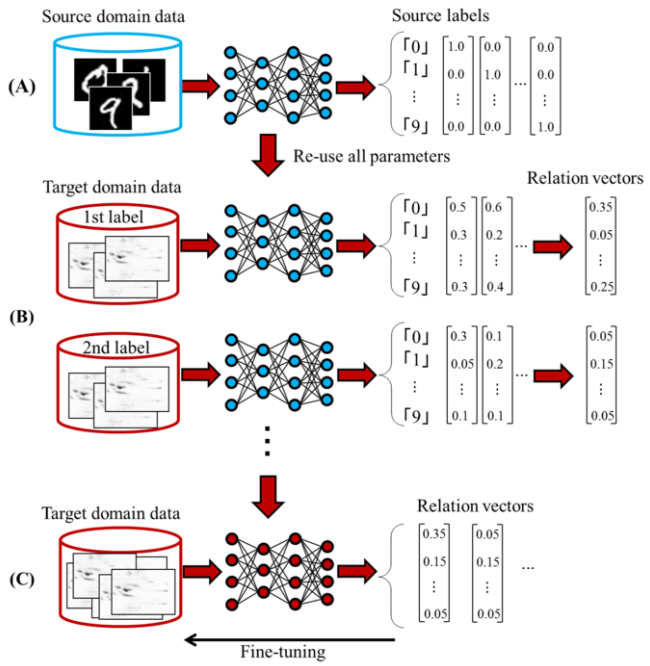


Figure 2. Outline of the proposed IATDL.

the variance between the output and relation vectors is sufficiently small (Fig. 2(C)). By using the steps corresponding to Fig. 2(B) and Fig. 2(C), we can transfer $\mathfrak{D}^s$ to the DNN for the target task, $\mathfrak{D}^t$, regularizing all parameters including the output layer. In the classification process, $\mathfrak{D}^t$ predicts the label $\hat{l}$ of the test vector like to minimize the Mahalanobis distance.

3 Experimental Results

Table 1. Classification performance of actual sepsis-data classification as function of the number of hidden layers.

	PPV	NPV	MCC	F1	ACC
PCA + logistic regression	**0.875**	0.805	0.545	0.609	0.816
Non-transfer (L=1)	0.725	0.983	0.755	0.829	0.878
SSL (L=1)	0.682	**1**	0.736	0.811	0.857
Oquab et al. [2] (L=1)	0.732	**1**	0.783	0.845	0.888
Agrawal et al. [1] (L=1)	0.750	**1**	0.800	0.857	0.898
ATDL (L=3)	**0.875**	0.970	**0.859**	**0.903**	**0.939**

We conducted experiments on TDEI classification for determining if an individual suffers from sepsis. We compared the classification performance of five methods: non-transfer learning, semi-supervised learning (SSL), transfer learning by Agrawal et al. [1], that by Oquab et al. [2], and ATDL.

The SSL is to compute $\boldsymbol{h}_i$ using $\boldsymbol{x}^s$ and $\boldsymbol{x}^t$ and fine-tunes using only $\boldsymbol{x}^t$. Agrawal's method removes the output layer of $\mathfrak{D}^s$ and adds a new output layer. In addition to these two steps, Oquab's method contains an additional adaptation layer to compensate for the different statistics of the source and target domain data. Then, Agrawal's method fine-tunes all layers including the hidden layers, and Oquab's method fine-tunes only the adaptation and output layers.

Table 2. List of source TDEIs. These images represent different extraction and refining protocols of proteins.

# of source TDEIs	Type of protocol
$N^s(1) = 25$	Change amount of protein
$N^s(2) = 4$	Change concentration protocol
$N^s(3) = 30$	Unprocessed
$N^s(4) = 49$	Removal of only top-2 abundant proteins
$N^s(5) = 11$	Focus on top-2 abundant proteins
$N^s(6) = 15$	Focus on 14 abundant proteins
$N^s(7) = 12$	Plasma sample instead of serum
$N^s(8) = 19$	Removal of Sugar chain
$N^s(9) = 15$	Other protocols

For the actual sepsis data classification, we collected the following number of target TDEIs, with sepsis data of 30 and non-sepsis data of 68. We evaluated the classification accuracy on the basis of the two-fold cross validation. As the source domain data, we first use TDEIs with different labels from the target domain data of sepsis or non-sepsis. The source task was to classify the differences between the extraction and refining protocol of protein [3]. shown in Table 2.

We compared the classification accuracy of ATDL with conventional methods with respect to changing the number of hidden layers $L = 1, 2, 3$ and 4.

Table 1 shows the best classification accuracies (ACCs) of each method including the baseline, PCA + logistic regression (used 188 features). It also lists the positive predictive values (PPVs), negative predictive values (NPVs), Matthews correlation coefficients (MCCs), and F1-scores (F1s) as reference. As shown in this table, classification accuracy improved by using transfer learning.

4 Conclusion

We proposed ATDL, a novel transfer learning method for DNNs, for a significantly small amount of training data. It computes the relation vectors that represent the characteristics of target labels by the source domain knowledge. By using the relation vectors, ATDL enables the transfer of all knowledge of DNNs including the output layer.

We applied ATDL to actual sepsis-data classification. The experimental results showed that ATDL outperformed other methods.

To the best of our knowledge, this work involved the first trial in which TDEIs were analyzed using the classification approach, which resulted in over 90% accuracy. Therefore, we believe that this article will be influential not only in machine learning, but also medical and biological fields.

5 Acknowledgments

We would like to acknowledge Dr. Iba, Professor of Juntendo University School of Medicine, for his help in collecting samples to generate TDEIs.

REFERENCES

[1] Pulkit Agrawal, Ross Girshick, and Jitendra Malik, 'Analyzing the performance of multilayer neural networks for object recognition', in *Proceedings of the European Conference on Computer Vision*, 329–344, (2014).

[2] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic, 'Learning and transferring mid-level image representations using convolutional neural networks', in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1717–1724, (2014).

[3] John M Walker, *The proteomics protocols handbook*, Springer, 2005.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1588

Computing Extensions' Probabilities in Probabilistic Abstract Argumentation: Beyond Independence

Bettina Fazzinga and **Sergio Flesca** and **Filippo Furfaro** [1]

Abstract. We characterize the complexity of the problem of computing the probabilities of the extensions in probabilistic abstract argumentation. We consider all the most popular semantics of extensions (*admissible*, *stable*, *preferred*, *complete*, *grounded*, *ideal-set*, *ideal* and *semi-stable*) and different forms of correlations that can be defined between arguments and defeats. We show that the complexity of the problem ranges from FP to $FP^{\#P}$-complete, with $FP^{||NP}$-complete cases, depending on the semantics of the extensions and the imposed correlations.

1 INTRODUCTION

In the last decade, several argumentation frameworks have been proposed, with the aim of suitably modeling disputes between two or more parties. Typically, argumentation frameworks model both the possibility of parties to present arguments supporting their theses, and the possibility that some arguments rebut other arguments. Although argumentation is strongly related to philosophy and law, it has gained remarkable interest in AI as a reasoning model for representing dialogues, making decisions, and handling inconsistency and uncertainty [3, 4, 15].

A powerful yet simple argumentation framework is that proposed in the seminal paper [5], called *abstract argumentation framework* (AAF). An AAF is a pair $\langle A, D\rangle$ consisting of a set A of *arguments*, and of a binary relation D over A, whose pairs are called *defeats* or, equivalently, *attacks*. Basically, an argument is an abstract entity that may attack and/or be attacked by other arguments, and an attack expresses the fact that an argument rebuts/weakens another argument.

Example 1 *The defense attorney of Mary and Marc wants to reason about the possible outcome of the trial of the robbery case involving his clients. The arguments of the case are the following, where Anne is a potential witness:*

a: "Mary says she was at the park when the robbery took place, and therefore denies being involved in the robbery";
b: "Marc says he was at home when the robbery took place, and therefore denies being involved in the robbery";
c: "Anne says that she is certain that he saw Mary outside the bank just before the robbery took place, and she also thinks that possibly she saw Marc there too".

The arguments a and b support the innocence of the defendants, and argument c means that a potential witness instills doubts about the innocence of both Mary and Marc. This scenario can be modeled by the AAF $\mathcal{A}$, whose set of arguments is $\{a, b, c\}$, and whose defeat relation consists of the defeats $\delta_{ac} = (a, c), \delta_{ca} = (c, a), \delta_{bc} = (b, c)$ and $\delta_{cb} = (c, b)$, meaning that arguments a and b are both attacked by c and they both counter-attack c.

Several semantics for AAFs, such as *admissible*, *stable*, *preferred*, *complete*, *grounded*, and *ideal-set*, have been proposed [5, 6, 2] to identify "reasonable" sets of arguments, called *extensions*. Basically, each of these semantics corresponds to some properties that "certify" whether a set of arguments can be profitably used to support a point of view in a discussion. For instance, a set S of arguments is an extension according to the *admissible* semantics if it has two properties: it is "*conflict-free*" (that is, there is no defeat between arguments in S), and every argument (outside S) attacking an argument in S is counterattacked by an argument in S. Intuitively enough, the fact that a set is an extension according to the admissible semantics means that, using the arguments in S, you do not contradict yourself, and you can rebut to anyone using an argument outside S to contradict yours. The other semantics correspond to other ways of determining whether a set of arguments would be a "good point" in a dispute.

As a matter of fact, in the real world, arguments and defeats are often uncertain, thus, several proposals have been made to model uncertainty in AAFs, by considering weights, preferences, or probabilities associated with arguments and/or defeats. In this regard, [7, 12, 9, 8, 11, 10, 14, 16] have recently extended the original Dung framework in order to achieve probabilistic abstract argumentation frameworks (prAAFs), where uncertainty of arguments and defeats is modeled by exploiting the probability theory. In particular, [14] proposed a form of prAAF (here denoted as IND, shorthand for "*independence*") where each argument and defeat can be associated with a probability value (and arguments and defeats are viewed as *independent* events), whereas [7] proposed a form of prAAF (here denoted as EX, shorthand for "*extensive*") where uncertainty can be taken into account by extensively specifying a probability distribution function (pdf) over the possible scenarios, as shown in the following example.

Example 2 (continuing Example 1) *In the case of modeling the uncertainty by assigning probabilities to possible scenarios, as done in prAAFs of form* EX*, suppose that the lawyer thinks that only the following 4 scenarios are possible:*
S_1: "Ann will not testify";
S_2: "Ann will testify, and the jury will deem that her argument c undermines those of Mary and Marc (arguments a, b), and vice versa";
S_3: "Ann will testify, and the jury will deem that her argument c undermines Mary's and Marc's arguments a, b, while, owing to the bad reputations of Mary and Marc, a and b will be not perceived as strong enough to undermine argument c";

[1] Bettina Fazzinga is with ICAR-CNR, Italy, Sergio Flesca and Filippo Furfaro are with DIMES-UNICAL, Italy, email: fazzinga@icar.cnr.it, flesca@dimes.unical.it, furfaro@dimes.unical.it

S_4: "Ann will testify, and the jury will deem that her argument c undermines Mary's argument a but not Marc's argument b, since Ann was uncertain about Marc's presence. In the other direction, a and b will be not perceived as strong enough to undermine c".

Each S_i is encoded by the AAF α_i in the following list:
$\alpha_1 = \langle\{a,b\},\emptyset\rangle$, $\alpha_2 = \langle\{a,b,c\},\{\delta_{ac},\delta_{ca},\delta_{bc},\delta_{cb}\}\rangle$,
$\alpha_3 = \langle\{a,b,c\},\{\delta_{ca},\delta_{cb}\}\rangle$, $\alpha_4 = \langle\{a,b,c\},\{\delta_{ca}\}\rangle$.

Basically, the form of prAAF EX *allows the lawyer to define, one by one, which scenarios are possible, and then to assign a probability to the AAF corresponding to each scenario, on the basis of her/his perception of how likely the scenario is. For instance, the pdf set by the lawyer could be such that: $P(\alpha_1) = 0.1$ and $P(\alpha_2) = P(\alpha_3) = P(\alpha_4) = (1-P(\alpha_1))/3 = 0.3$, meaning that the lawyer thinks that there is 10% probability that Mary will not manage to testify (owing to her ill-health), and that, in the case she testifies, the other three scenarios are equi-probable.*

Example 3 (continuing Example 1) *In the case that a prAAF of form* IND *is used, the lawyer can associate each argument and defeat with a probability. For instance, the lawyer may set $P(c) = 0.9$ (meaning that there is 10% probability that Mary will not manage to testify) and $P(a) = P(b) = 1$ (meaning that Mary and Marc will certainly testify). Moreover, she/he could set $P(\delta_{ca}) = 1$ (meaning that she/he is certain that the jury will consider Ann's argument as a solid rebuttal of Mary's argument). Analogously, she/he could set $P(\delta_{cb}) = 0.8$ and $P(\delta_{ac}) = P(\delta_{bc}) = 0.4$.*

Given this, since the arguments are considered independent, the possible scenarios modeled by IND *are not only $\alpha_1,\ldots,\alpha_4$ of the previous example, but all the AAFs $\langle A_i, D_i\rangle$ where A_i is a subset of the arguments and D_i a subset of the defeats between the arguments in A_i. Specifically, there are 9 possible AAFs, where 4 out of 9 are equal to $\alpha_1,\ldots\alpha_4$ of the previous example, and the others are $\alpha_5 = \langle\{a,b,c\},\{\delta_{ac},\delta_{ca}\}\rangle$, $\alpha_6 = \langle\{a,b,c\},\{\delta_{ca},\delta_{bc}\}\rangle$, $\alpha_7 = \langle\{a,b,c\},\{\delta_{ca},\delta_{cb},\delta_{ac}\}\rangle$, $\alpha_8 = \langle\{a,b,c\},\{\delta_{ca},\delta_{cb},\delta_{bc}\}\rangle$, $\alpha_9 = \langle\{a,b,c\},\{\delta_{ca},\delta_{ac},\delta_{bc}\}\rangle$. Moreover, the probability assigned to each AAF $\langle A, D\rangle$ is the result of a product, whose factors are the probabilities (resp., the complements of the probabilities) of the arguments in A (resp., not in A) and the probabilities (resp., the complements of the probabilities) of the defeats between arguments in A that are in D (resp., are not in D). For instance, $P(\alpha_1) = P(a)\times P(b)\times(1-P(c)) = 0.1$ and $P(\alpha_3) = P(a)\times P(b)\times P(c)\times P(\delta_{ca})\times P(\delta_{cb})\times(1-P(\delta_{ac}))\times(1-P(\delta_{bc})) = 0.26$.*

2 COMPLEXITY OF $\text{PROB}^{sem}_{\mathcal{F}}(S)$

The complexity of the fundamental problem of computing the extensions' probabilities over a prAAF of form $\mathcal{F}$ (denoted, in the following, as $\text{PROB}^{sem}_{\mathcal{F}}$) has been thoroughly characterized in [10], in the specific case that $\mathcal{F}$ is IND. Much less is known about the complexity of the same problem over different forms of prAAFs.

We here consider a general form of prAAF (called GEN) with three main amenities: 1) it generalizes EX, since it also enables an "extensive" definition of the pdf over the possible AAFs; 2) it generalizes IND, since it also allows us to impose independence between arguments and defeats; 3) in order to encode a pdf over the possible AAFs, it exploits the representation model of *world-set descriptors* (wsds) and *world-set sets* (ws-sets), that is known to be a succinct and complete model for representing possible worlds and probabilities over them [1, 13]. This paradigm GEN can be exploited to define different syntactic classes of wsds and ws-sets, each allowing different forms of correlations (mutual exclusion, co-occurrence, etc.).

We consider the following well-known semantics: *admissible* (ad), *stable* (st), *complete* (co), *grounded* (gr), *preferred* (pr), *ideal-set* (ids), *ideal* (ide), and *semi-stable* (sst), and we show that the complexity of $\text{PROB}^{sem}(S)$ ranges from FP to $FP^{\#P}$, depending on the semantics of the extensions and the syntactic class of the wsds.

Theorem 1 *Let sem be a semantics in $\{\texttt{ad},\texttt{st},\texttt{gr},\texttt{co}\}$ and $\mathcal{F}$ a prAAF of form* GEN. *The complexity of $\text{PROB}^{sem}_{\mathcal{F}}(S)$ ranges from FP to $FP^{\#P}$-complete depending on the syntactic class of the wsds.*

Theorem 2 *Let sem be a semantics in $\{\texttt{pr},\texttt{ide},\texttt{ids},\texttt{sst}\}$ and $\mathcal{F}$ a prAAF of form* GEN. *The complexity of $\text{PROB}^{sem}_{\mathcal{F}}(S)$ ranges from $FP^{||}$-complete to $FP^{\#P}$-complete depending on the syntactic class of the wsds.*

3 CONCLUSION

The problem of characterizing the complexity of the fundamental problem $\text{PROB}^{sem}(S)$ of evaluating the probabilities of extensions in probabilistic abstract argumentation frameworks has been addressed, showing that the complexity of $\text{PROB}^{sem}(S)$ ranges from FP to $FP^{\#P}$, depending on the semantics of the extensions and the syntactic class of the wsds.

REFERENCES

[1] Lyublena Antova, Thomas Jansen, Christoph Koch, and Dan Olteanu, 'Fast and simple relational processing of uncertain data', in *Proc. Int. Conf. on Data Engineering (ICDE)*, pp. 983–992, (2008).

[2] Pietro Baroni and Massimiliano Giacomin, 'Semantics of abstract argument systems', in *Argumentation in Artificial Intelligence*, 25–44, (2009).

[3] Trevor J. M. Bench-Capon and Paul E. Dunne, 'Argumentation in artificial intelligence', *Artif. Intell.*, **171**(10-15), 619–641, (2007).

[4] *Elements Of Argumentation*, eds., Philippe Besnard and Anthony Hunter, The MIT Press, 2008.

[5] Phan Minh 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).

[6] Phan Minh Dung, Paolo Mancarella, and Francesca Toni, 'Computing ideal sceptical argumentation', *Artif. Intell.*, **171**(10-15), 642–674, (2007).

[7] Phan Minh Dung and Phan Minh Thang, 'Towards (probabilistic) argumentation for jury-based dispute resolution', in *COMMA*, pp. 171–182, (2010).

[8] Bettina Fazzinga, Sergio Flesca, and Francesco Parisi, 'Efficiently estimating the probability of extensions in abstract argumentation', in *Scalable Uncertainty Management - 7th International Conference, SUM*, pp. 106–119, (2013).

[9] Bettina Fazzinga, Sergio Flesca, and Francesco Parisi, 'On the complexity of probabilistic abstract argumentation', in *IJCAI*, (2013).

[10] Bettina Fazzinga, Sergio Flesca, and Francesco Parisi, 'On the complexity of probabilistic abstract argumentation frameworks', *ACM Trans. Comput. Log.*, **16**(3), 22, (2015).

[11] Bettina Fazzinga, Sergio Flesca, and Francesco Parisi, 'On efficiently estimating the probability of extensions in abstract argumentation frameworks', *Int. J. Approx. Reasoning*, **69**, 106–132, (2016).

[12] Bettina Fazzinga, Sergio Flesca, Francesco Parisi, and Adriana Pietramala, 'PARTY: A mobile system for efficiently assessing the probability of extensions in a debate', in *Proc. of International Conference on Database and Expert Systems Applications*, pp. 220–235, (2015).

[13] Christoph Koch and Dan Olteanu, 'Conditioning probabilistic databases', *PVLDB*, **1**(1), 313–325, (2008).

[14] Hengfei Li, Nir Oren, and Timothy J. Norman, 'Probabilistic argumentation frameworks', in *TAFA*, (2011).

[15] *Argumentation in Artificial Intelligence*, eds., Iyad Rahwan and Guillermo R. Simari, Springer, 2009.

[16] Tjitze Rienstra, 'Towards a probabilistic dung-style argumentation system', in *AT*, pp. 138–152, (2012).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1590

Explained Activity Recognition with Computational Assumption-Based Argumentation

Xiuyi Fan[1], **Siyuan Liu**[1], **Huiguo Zhang**[1], **Cyril Leung**[2], **Chunyan Miao**[1]

Abstract. Activity recognition is a key problem in multi-sensor systems. In this work, we introduce *Computational Assumption-based Argumentation*, an argumentation approach that seamlessly combines sensor data processing with high-level inference. Our method gives classification results comparable to machine learning based approaches with reduced training time while also giving explanations.

1 Introduction

We present an argumentation based approach for activity recognition. *Computational Assumption-based Argumentation (CABA)* is an argumentation framework that connects low-level sensor data processing with high-level argumentative reasoning. In the proposed CABA framework, sensor data is processed to form arguments. Together with pre-defined arguments based on domain knowledge representing activities, they jointly construct argumentative inferences such that "winning arguments" represent recognized activities.

CABA frameworks are extensions of the widely recognized Assumption-based Argumentation (ABA) frameworks [4] with added *Computation Units (CUs)*. A CU represents a purposefully designed (numerical) computation that is difficult to represent with plain ABA. CUs are seamlessly built into CABA arguments in ways assumptions are built into ABA arguments. With CUs introduced, the *acceptability* of a CABA argument depends on attack relations and its CUs. Thus, low-level data processing is "packaged" into CUs whereas high-level reasoning as defined by standard argumentation semantics. Upon recognizing activities, explanation for classification is provided. We leverage our previous work on argumentation explanation [1, 2, 3] in the development of CABA explanation.

We test our CABA based activity recognition algorithm in a smart home equipped with: (1) two Grid-Eye inferred sensors[3], (2) two force sensors, (3) one noise sensor, and (4) one electric current detector. We focus on six activities: (1) eat, (2) watch TV, (3) read books, (4) sleep, (5) visit and (6) other. We assume that at any moment, there is one and only one activity taking place.

2 CABA and Explained Activity Recognition

Computation Units (CUs) are core components of CABA. Formally:

Definition 1. A *computation unit (cu)* is a tuple $u = \langle \mathsf{T}, \mathsf{C}, \mathsf{E} \rangle$:
• $\mathsf{T} \subseteq \mathtt{D}_u$: T is the *Data*, and $\mathtt{D}_u$ is the *Domain*;
• $\mathsf{C} : \mathtt{D}_u \mapsto \mathtt{R}_u$: C is the *Computation Function (Computation)*, and $\mathtt{R}_u$ is the *Range*;
• $\mathsf{E} : \mathtt{R}_u \mapsto \{\top, \bot\}$: E is the *Evaluation Function (Evaluation)*.
We say that $u = \langle \mathsf{T}, \mathsf{C}, \mathsf{E} \rangle$ is *successful* iff $\mathsf{E} = \top$ and *well-formed* iff both C and E are total and computable.

In this work, we focus on well-formed CUs. We introduce four CUs: u_{tv}, u_{tb}, u_{bed} and u_{one} representing whether the TV is on, the table is occupied, the bed is occupied and there is a single person, respectively.
1. **TV on:** $u_{tv} = \langle \mathsf{T}_{tv}, \mathsf{C}_{tv}, \mathsf{E}_{tv} \rangle$ in which:
$\mathsf{T}_{tv} \subseteq \{0, 1\}$ is the output from the current sensor;
$\mathsf{C}_{tv}(\mathsf{T}_{tv}) = \mathsf{T}_{tv}$; and
$\mathsf{E}_{tv} = \top$ if $\mathsf{C}_{tv} = 1$ and $\mathsf{E}_{tv} = \bot$ otherwise.
2. **Table occupied:** $u_{tb} = \langle \mathsf{T}_{tb}, \mathsf{C}_{tb}, \mathsf{E}_{tb} \rangle$ in which:
$\mathsf{T}_{tb} \subseteq \mathbb{N}^{8\times 8}$ is the output from GridEye 1;[4]
$\mathsf{C}_{tb}(\mathsf{T}_{tb}) = CL_{tb}(\mathsf{T}_{tb}, \omega_{tb})$ is a classifier such that, with some parameter ω_{tb}, $CL_{tb}(\mathsf{T}_{tb}, \omega_{tb}) \in \{\top, \bot\}$ indicates whether there is any person sitting next to the dining table; and
$\mathsf{E}_{tb} = \mathsf{C}_{tb}$.
3. **Bed occupied:** $u_{bed} = \langle \mathsf{T}_{bed}, \mathsf{C}_{bed}, \mathsf{E}_{bed} \rangle$ in which:
$\mathsf{T}_{bed} \subseteq \mathbb{N}^{8\times 8}$ is the output from GridEye 2;
$\mathsf{C}_{bed}(\mathsf{T}_{bed}) = CL_{bed}(\mathsf{T}_{bed}, \omega_{bed})$ is a classifier such that, with some parameter ω_{bed}, $CL_{bed}(\mathsf{T}_{bed}, \omega_{bed}) \in \{\top, \bot\}$ indicates whether there is any person on the bed; and
$\mathsf{E}_{bed} = \mathsf{C}_{bed}$.
4. **One person in room:** $u_{one} = \langle \mathsf{T}_{one}, \mathsf{C}_{one}, \mathsf{E}_{one} \rangle$ in which:
$\mathsf{T}_{one} \subseteq \mathbb{N}^{132\times 1}$ is the output from all of our sensors;
$\mathsf{C}_{one}(\mathsf{T}_{one}) = CL_{one}(\mathsf{T}_{one}, \omega_{one})$ is a classifier such that, with some parameter ω_{one}, $CL_{one}(\mathsf{T}_{one}, \omega_{one}) \in \{\top, \bot\}$ indicates whether there is a single person in the room; and
$\mathsf{E}_{one} = \mathsf{C}_{one}$.

Definition 2. *Computational Assumption-based Argumentation frameworks* are tuples $\langle \mathcal{U}, \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle$ where
• $\mathcal{U}$ is a set of well-formed CUs;
• $\langle \mathcal{L}, \mathcal{R}, \mathcal{U} \rangle$ is a deductive system, with $\mathcal{L}$ the *language* and $\mathcal{R}$ a set of *rules* of the form $s_0 \leftarrow s_1, \ldots, s_m (m \geq 0, s_i \in \mathcal{L} \cup \mathcal{U}$ for $i > 0$, $s_i \in \mathcal{L}$ for $i = 0)$;
• $\mathcal{A} \subseteq \mathcal{L}$ is a (non-empty) set, whose elements are *assumptions*;
• $\mathcal{C}$ is a total mapping from $\mathcal{A}$ into $2^{\mathcal{L}} - \{\{\}\}$, where each $s \in \mathcal{C}(a)$ is a *contrary* of a, for $a \in \mathcal{A}$.
Given a rule ρ of the form $s_0 \leftarrow s_1, \ldots, s_m$, s_0 is referred as the *head* and $s_1, \ldots, s_m$ as the *body* of ρ.

We use the following CABA framework for activity recognition.
• $\mathcal{U}$ contains the following CUs: u_{tv} u_{tb} u_{bed} u_{one}

[1] Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore
[2] Electrical and Computer Engineering, The University of British Columbia, Canada
[3] https://na.industrial.panasonic.com/products/sensors/sensors-automotive-industrial-applications/grid-eye-infrared-array-sensor
[4] The output of a GridEye is an 8-by-8 integer matrix.

• $\mathcal{L}$ contains the following sentences:

watchTV	eat	sleep	visit
notWatch	notEat	notSleep	notVisit
read	other	notRead	noAct
TVon	tableOccupied	bedOccupied	onePerson
TVoff	emptyTable	emptyBed	twoAct

• $\mathcal{R}$ contains the following rules:

notWatchTV ← TVoff	notEat ← emptyTable
notSleep ← emptyBed	notVisit ← onePerson
notRead ← TVon	notRead ← tableOccupied
notRead ← bedOccupied	other ← noAct
twoAct ← watchTV, eat	twoAct ← watchTV, sleep
twoAct ← watchTV, visit	twoAct ← watchTV, read
twoAct ← eat, sleep	twoAct ← eat, visit
twoAct ← eat, read	twoAct ← sleep, visit
twoAct ← sleep, read	twoAct ← visit, read
TVon ← u_{tv}	tableOccupied ← u_{tb}
bedOccupied ← u_{bed}	onePerson ← u_{one}

• $\mathcal{A}$ contains the following assumptions:

watchTV	eat	sleep	visit	read
TVoff	emptyTable	emptyBed	noAct	

• $\mathcal{C}$ are:

$\mathcal{C}$(noAct) = {watchTV, eat, sleep, read, visit}
$\mathcal{C}$(watchTV) = {notWatchTV, twoAct} $\mathcal{C}$(eat) = {notEat, twoAct}
$\mathcal{C}$(sleep) = {notSleep, twoAct} $\mathcal{C}$(read) = {notRead, twoAct}
$\mathcal{C}$(visit) = {notVisit, twoAct} $\mathcal{C}$(TVoff) = {TVon}
$\mathcal{C}$(emptyTable) = {tableOccupied} $\mathcal{C}$(emptyBed) = {bedOccupied}

We define CABA arguments and attacks as follows.

Definition 3. A CABA argument for (claim) $s \in \mathcal{L}$ *supported by* $\Delta \subseteq \mathcal{A}$ *based on* $U \subseteq \mathcal{U}$ (denoted $[\Delta, U] \vdash s$) is a finite tree with nodes labeled by sentences in $\mathcal{L}$, CUs in U or by $\tau \notin \mathcal{L} \cup \mathcal{U}$, the root labeled by s, leaves labeled by either τ, assumptions in Δ, or CUs in U, and non-leaves labelled by s' with, as children, sentences in the body of some rule with head s'.

A CABA argument $A = [\Delta, U] \vdash s$ is *applicable* iff for all CUs $u = \langle \mathsf{T}_u, \mathsf{C}_u, \mathsf{E}_u \rangle \in U$, $\mathsf{E}_u = \top$. For a CABA argument $A = [\Delta, U] \vdash s$, if $U = \{\}$, then A is abbreviated to $\Delta \vdash s$. Given a CABA framework *F*, an argument *is in F* iff all its rules, assumptions and CUs are in *F*. $\mathtt{A}^F$ denotes the set of all arguments in *F*.

Definition 4. Given a CABA framework *F*, an argument $[A_1, U_1] \vdash s_1$ (in *F*) *attacks* an argument $[A_2, U_2] \vdash s_2$ (in *F*) iff s_1 is a contrary of some assumption in A_2. $\mathtt{R}^F$ denotes the set of all attacks in *F*.

We let admissibility apply in CABA with additional conditions: (1) a set of arguments is admissible only if they are applicable and (2) an admissible set of arguments only needs to counter-attack all attacks from applicable attackers. We formalize explanations for non-admissible CABA arguments as follows.

Definition 5. Given a CABA framework *F* with arguments $\mathtt{A}^F$ and attacks $\mathtt{R}^F$, and CABA argument $A \in \mathtt{A}^F$ such that A is not admissible in *F*, then, if there exists some $As \subseteq \mathtt{A}^F$, such that: (1) A is admissible in $\langle \mathtt{A}^F, \mathtt{R}^F \rangle \setminus As$, and (2) there is no $As' \subset As$ such that A is admissible in $\langle \mathtt{A}^F, \mathtt{R}^F \rangle \setminus As'$, then As is an *explanation* of A. Otherwise, $\{A\}$ is the *explanation* of A.

Suppose that for some sensor data, both u_{tb} and u_{one} are successful whereas u_{tv} and u_{bed} are not successful. We see that {watchTV} ⊢ watchTV is not admissible. However, if arguments
$A = \{\text{TVoff}\} \vdash \text{notWatchTV}$, $B = [\{\}, u_{tb}] \vdash \text{TableOccupied}$
are removed, then {watchTV} ⊢ watchTV becomes admissible. Hence, $\{A, B\}$ is an explanation for {watchTV} ⊢ watchTV. We can interpret this as:

An explanation for "not watching TV" is that the TV is off and there is a person using the dining table.

To evaluate CABA based activity recognition, we have had four individuals performing the six activities. Each person performs the six activities in two runs. Data collected from six runs from three testing subjects are used for training with the remaining two runs from the fourth subject for testing. Overall, there are 7781 instances of training samples and 1437 instances of testing samples.

With the CABA framework presented earlier, to perform activity recognition we use four CUs, u_{tv}, u_{tb}, u_{bed} and u_{one}. We let u_{tv} use the value directly from the current sensor. To construct u_{tb} and u_{one}, we choose perceptron classifiers for their simplicity. Specifically, for u_{tb}, we construct a 64-node perceptron model with a single output node; for u_{one}, we construct a 132-node perceptron model with a single output node. To construct u_{bed}, we use binary thresholding based on GridEye 2's output to test if the bed is occupied.

We compare our CABA-based classifier with Naive Bayes, Decision Tree and Neural Networks using precision, recall and training time. The results are summarized in Table 1. These results illustrate the effectiveness of introducing domain knowledge in argumentation form in solving activity recognition problems.

	Precision	Recall	Training Time (s)
Naive Bayes	0.558	0.578	2.89
Decision Tree	0.576	0.397	0.46
NN 2-Hidden Layers	0.801	0.797	549.32
Deep Neural Network	0.816	0.812	1865.34
CABA-Classifier	0.821	0.831	53.55

Table 1. Performance comparison between Naive Bayes, Decision Tree, Neural Networks 2 and 8 hidden layers and the proposed CABA-classifier.

3 Conclusion

In this work, we presented CABA to seamlessly connect low-level data processing with high-level inference based reasoning. We used CABA to solve an activity recognition problem, with promising results comparable to traditional machine learning algorithms. The advantage of CABA is twofold. Firstly, used as a channel for injecting domain knowledge into problem solving, CABA significantly reduces the training time required for model construction. Secondly, the argumentative structure of CABA provides the basis for generating explanations for the modeled computation.

Acknowledgments

This research was supported by the National Research Foundation Singapore under its Interactive Digital Media (IDM) Strategic Research Programme.

REFERENCES

[1] X. Fan and F. Toni, 'On computing explanation in abstract argumentation', in *Proc. ECAI*, (2014).
[2] X. Fan and F. Toni, 'On computing explanations for non-acceptable arguments', in *Proc. TAFA*, (2015).
[3] X. Fan and F. Toni, 'On computing explanations in argumentation', in *Proc. of AAAI*, (2015).
[4] F. Toni, 'A tutorial on assumption-based argumentation', *Argument & Computation, Special Issue: Tutorials on Structured Argumentation*, **5**(1), 89–117, (2014).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1592

Execution Errors Enable the Evolution of Fairness in the Ultimatum Game

Fernando P. Santos[1,2] and **Francisco C. Santos**[1,2] and **Ana Paiva**[1] and **Jorge M. Pacheco**[2,3]

The goal of designing autonomous and successful agents is often attempted by providing mechanisms to choose actions that maximise some reward function. When agents interact with a static environment, the provided reward functions are well-defined and the implementation of traditional learning algorithms turns to be feasible. However, agents are not only intended to act in isolation. Often, they interact in dynamic multiagent systems whose decentralised nature of decision making, huge number of opponents and evolving behaviour stems a complex adaptive system [3]. This way, it is an important challenge to unveil the long term outcome of agents' strategies, both in terms of individual goals and social desirability [9]. This endeavour can be conveniently achieved through the employment of new tools from, e.g., population dynamics [4] and complex systems research, in order to grasp the effects of implementing agents whose strategies, even rational in the context of static environments, may turn to be disadvantageous (individually and socially) when successively applied by members of a dynamic population.

In this paper, we present a paradigmatic scenario in which behavioural errors are pernicious if committed in isolation, yet are the source of long-term success when considering adaptive populations. Moreover, errors support population states in which fairness (less inequality) is augmented. We assume that the goals and strategies of agents are formalised through the famous Ultimatum Game (**UG**) [2]. We focus on the changes regarding the frequency of agents adopting each strategy, over time. This process of social learning, essentially analogous to the evolution of animal traits in a population, enables us to use the tools of Evolutionary Game Theory (EGT), originally applied in the context of theoretical biology [4]. We describe analytically the behavioural outcome in a discretised strategy space of the UG, in the limit of small exploration rates (or the so-called mutations) [1]. This allows us to replicate the results of large-scale simulations [7], yet avoiding the burden of computational resources.

1 Ultimatum Game

The rules of UG are quite simple: some amount of a resource is conditionally endowed to one agent (Proposer) that must suggest a division with a Responder; secondly, the Responder will accept or reject the offer. The agents divide the money as it was proposed, if the Responder accepts. By rejecting, no one gets anything.

The rational behaviour in UG can be defined using a game-theoretical equilibrium analysis, through a simple backward induction. Facing the decision of rejecting (earn 0) or accepting (earn some money, even if a really small quantity), the Responder would always prefer to accept. Secure about this certain acceptance, the Proposer will offer the minimum possible, maximising his own share. Denoting by p the fraction of the resource offered by the Proposer, $p \in [0,1]$, and by q the acceptance threshold of the Responder, $q \in [0,1]$, acceptance will occur whenever $p \geq q$ and the *subgame perfect equilibrium* [5] of this game is defined by values of p and q slightly above 0. This outcome is said to be unfair, as it presents a profound inequality between the gains of Proposer and Responder. The strategies of agents that value fairness are characterised by prescribing a more equalitarian outcome: a fair Proposer suggests an offer close to 0.5 and a fair Responder rejects unfair offers, much lower than 0.5 (i.e. $p = 0.5$ and $q = 0.5$).

2 Analytical Framework and Results

We consider the existence of two populations (Proposers and Responders) each one composed by Z agents. Let us assume that a Proposer and a Responder may choose one of S strategies, corresponding to increasing divisions of 1. A Proposer choosing strategy $m \in \{1,2,...,S\}$ will offer the corresponding to $p_m = \frac{1}{S}m$ and a Responder choosing strategy $n \in \{1,2,...,S\}$ will accept any offer equal or above $q_n = \frac{1}{S}n$. The two-person encounter between a Proposer and a Responder thus yields $1 - p_m$ to the Proposers and p_m to the Responder if the proposal is accepted ($n : q_n \leq p_m$) and 0 to both agents otherwise.

We are concerned with the role of systematic errors in the execution of the desired strategy, namely, by the Responders. We assume that each Responder with strategy n (and threshold of acceptance q_n) will actually use a threshold of q'_n, calculated as $q'_n = q_n + U(-\epsilon, \epsilon)$, where $U(-\epsilon, \epsilon)$ corresponds to an error sampled from a uniform distribution between $-\epsilon$ and ϵ. Thereby, a Responder (using strategy n) accepts an offer $p_m \in [q_n - \epsilon, q_n + \epsilon]$ with a probability given by $P(q'_n \leq p_m) = P(q_n + U(-\epsilon,\epsilon) \leq p_m) = P(U(-\epsilon,\epsilon) \leq p_m - q_n) = \int_{-\epsilon}^{p_m - q_n} \frac{1}{2\epsilon} d(p_m - q_n) = \frac{p_m - q_n + \epsilon}{2\epsilon}$. The probability of acceptance is 0 if $p_m < q_n - \epsilon$ and is 1 if $p_m \geq q_n + \epsilon$. The resulting payoff of a pair (proposal, acceptance threshold) is, thereby, linearly weighted by the probability of acceptance, considering the execution error (ϵ).

As we assume a well-mixed population, the fitness of each individual is given by the average payoff earned when playing with all the agents in the opposite population. Considering the existence of S different strategies in the opposite population of the one from which agent A belongs; denoting k_i as the number of agents using strategy i, in the opposite population of A; and regarding $R_{j,i}$ as the payoff (reward) earned by an agent A using strategy j, against an agent with

[1] INESC-ID and Instituto Superior Técnico, Universidade de Lisboa, email: fernando.pedro@tecnico.ulisboa.pt
[2] ATP-Group, Lisboa, Portugal
[3] CBMA and Departamento de Matemática e Aplicações, Universidade do Minho

strategy i, the fitness of agent A is given by $F_{A_j} = \sum_{i=1}^{S} \frac{k_i}{Z} R_{j,i}$.

The adoption of strategies will evolve following an imitation process. We assume that at each step two agents are chosen, one that will imitate (agent A) and one whose fitness and strategy will serve as model (agent B). The imitation probability will be calculated using a function — $(1+e^{-\beta(F_B-F_A)})^{-1}$ — that grows monotonously with the fitness difference $F_B - F_A$ [10]. The variable β in the equation above is well-suited to control the selection pressure, allowing to manipulate the extent to which imitation depends on the fitness difference. Assuming that two agents are randomly sampled from the population in which k_i agents are using strategy i (the remaining are using strategy j), the probability of having ± 1 (more or less 1) individual using strategy i is given by

$$T^{\pm}(k_i) = \frac{Z-k_i}{Z}\frac{k_i}{Z-1}\left(1+e^{\mp\beta(F_i(\overline{k}_s)-F_j(\overline{k}_s))}\right)^{-1} \quad (1)$$

assuming that in the opposite population the number of agents using another strategy s is $\overline{k}_s$ and that the population size is Z. Note that $\frac{Z-k_i}{Z}$ (and $\frac{k_i}{Z-1}$) represent the sampling probabilities of choosing one agent with strategy $j(i)$.

With probability μ, a mutation occurs and individuals change their strategy to a random one, exploring a new behaviour regardless the observation of others. The imitation process described above will occur with probability $(1-\mu)$. If we assume that $\mu \to 0$, we are able to derive analytical conclusions through a simpler apparatus [1]. Under this regime in which mutations are extremely rare, a *mutant* strategy will either fixate in the population or will completely vanish [1] and the number of different strategies present in the population is at most two. The time between two mutation events is usually so large that the population will always evolve to a monomorphic state (i.e., all agents using the same strategy) before the next mutation occurs. This fact allows us to conveniently use Equation (1) in the calculation of the transition probabilities between intermediate states (where a mutant is still invading and two strategies co-exist in the population). The transitions between monomorphic states are described through the fixation probability of every single mutant of strategy i in every resident population of strategy j. A strategy i will fixate in a population composed by $Z-1$ individuals using strategy j with a probability given by

$$\rho_{i\to j}(\overline{k}_s) = \frac{1-e^{-\beta\Delta F(\overline{k}_s)}}{1-e^{-Z\beta\Delta F(\overline{k}_s)}} \quad (2)$$

These probabilities define an embedded Markov Chain, governed by the stochastic matrix T, in which $T_{i,j} = \rho_{i\to j}$ defines the fixation probability of a mutant with strategy i in a population with $Z-1$ individuals using strategy j. A derivation from Equation (1) to Equation (2) and more details regarding Equation (2) can be found in [4, 10]. To calculate π, the stationary distribution of this Markov Process, we compute the normalised eigenvector associated with the eigenvalue 1 of the transposed of T. $\pi_{a,b}$ represents the fraction of time, on average, that is spent when the population of Proposers is using strategy a and the population of Responders is using strategy b. The number of possible states depends on the discretisation considered in the strategy space available in Ultimatum Game. If the Proposer and Responder have, each, S available strategies, there are S^2 different monomorphic states. The resulting average fitness is provided by the average fitness of a population in each monomorphic state, weighted by the time spent in that state. Thereby, the average fitness of the population of Proposers is given by $\overline{F} = \sum_{a=1,b=1}^{S} \pi_{a,b} R_{a,b}$

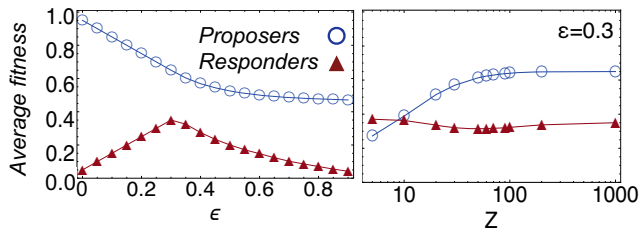


Figure 1. Analytical results showing the role of errors (ϵ) and population sizes (Z) in the overall fitness of Proposers and Responders. $\beta = 10$, $Z = 100$, $S = 20$

and the average fitness of the population of Responders is given by $\overline{F} = \sum_{a=1,b=1}^{S} \pi_{a,b} R_{b,a}$.

With this framework, we are able to show that the fitness of the Responders will be maximised if they commit a significant execution error, sampled from an interval close to $[-0.3, 0.3]$ (Figure 1). Moreover, this framework is well suited to capture the stochastic effects of considering finite populations and even the role of different population sizes. We additionally verified that increasing β and Z promotes determinism in the imitation process which benefits the fitness of the Proposers and undermines the fair distribution of gains between Proposers and Responders. Even employing a different methodology, these results are in line with the discussions performed in [6, 8].

ACKNOWLEDGEMENTS

This research was supported by Fundação para a Ciência e Tecnologia (FCT) through grants SFRH/BD/94736/2013, PTDC/EEI-SII/5081/2014, PTDC/MAT/STA/3358/2014 and by multi-annual funding of CBMA and INESC-ID (under the projects UID/BIA/04050/2013 and UID/CEC/50021/2013) provided by FCT.

REFERENCES

[1] Drew Fudenberg and Lorens A Imhof, 'Imitation processes with small mutations', *J Econ Theory*, **131**(1), 251–262, (2006).
[2] Werner Güth, Rolf Schmittberger, and Bernd Schwarze, 'An experimental analysis of ultimatum bargaining', *J Econ Behav Organ*, **3**(4), 367–388, (1982).
[3] John H Miller and Scott E Page, *Complex adaptive systems: an introduction to computational models of social life*, Princeton University Press, 2009.
[4] Martin A Nowak, *Evolutionary dynamics: exploring the equations of life*, Harvard University Press, 2006.
[5] Martin J Osborne, *An introduction to game theory*, Oxford University Press New York, 2004.
[6] David G Rand, Corina E Tarnita, Hisashi Ohtsuki, and Martin A Nowak, 'Evolution of fairness in the one-shot anonymous ultimatum game', *Proc Natl Acad Sci USA*, **110**(7), 2581–2586, (2013).
[7] Fernando P Santos, Francisco C Santos, and Ana Paiva, 'The evolutionary perks of being irrational', in *Proceedings of the 2015 AAMAS*, pp. 1847–1848. International Foundation for Autonomous Agents and Multiagent Systems, (2015).
[8] Fernando P Santos, Francisco C Santos, Ana Paiva, and Jorge M Pacheco, 'Evolutionary dynamics of group fairness', *J Theor Biol*, **378**, 96–102, (2015).
[9] Yoav Shoham, Rob Powers, and Trond Grenager, 'If multi-agent learning is the answer, what is the question?', *Artif Intell*, **171**(7), 365–377, (2007).
[10] Arne Traulsen, Martin A Nowak, and Jorge M Pacheco, 'Stochastic dynamics of invasion and fixation', *Phys Rev E*, **74**(1), 011909, (2006).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1594

Encoding Cryptographic Functions to SAT Using TRANSALG System

Ilya Otpuschennikov[1], Alexander Semenov[1], Irina Gribanova[1], Oleg Zaikin[1], Stepan Kochemazov[1]

Abstract. In this paper we propose the technology for constructing propositional encodings of discrete functions. It is aimed at solving inversion problems of considered functions using state-of-the-art SAT solvers. We implemented this technology in the form of the software system called TRANSALG, and used it to construct SAT encodings for a number of cryptanalysis problems. By applying SAT solvers to these encodings we managed to invert several cryptographic functions. In particular, we used the SAT encodings produced by TRANSALG to construct the family of two-block MD5 collisions in which the first 10 bytes are zeros. In addition to that we used TRANSALG encoding for the widely known A5/1 keystream generator to solve several dozen of its cryptanalysis instances in a distributed computing environment. Also in the present paper we compare the functionality of TRANSALG with that of similar software systems.

1 FOUNDATIONS OF SAT-BASED CRYPTANALYSIS

By $\{0,1\}^*$ we denote the set of all binary words of an arbitrary finite length. By discrete functions we mean arbitrary (possibly, partial) functions of the kind: $f : \{0,1\}^* \to \{0,1\}^*$. Hereinafter we consider only total computable discrete functions. In other words we assume that f is specified by some program (algorithm) A_f, that has finite runtime on each word from $\{0,1\}^*$. The program A_f specifies a family of functions of the kind $f_n : \{0,1\}^n \to \{0,1\}^*$, $n \in N_1$. Below we consider the problem of inversion of an arbitrary function f_n as follows: based on the known $y \in Range f_n$ and the known algorithm A_f, find such $x \in \{0,1\}^n$ that $f(x) = y$. Many cryptanalysis problems can be formulated as inversion problems of discrete functions. For example, suppose that given a secret key $x \in \{0,1\}^n$, f_n generates a pseudorandom sequence (in general, of an arbitrary length), that is later used to cipher some plaintext via bit-wise XOR. Such a sequence is called a *keystream*. Knowing some fragment of plaintext lets us know the corresponding fragment of keystream, i.e. some word y for which we can consider the problem of finding such $x \in \{0,1\}^n$, that $f_n(x) = y$. Regarding cryptographic keystream generators this corresponds to the so called known plaintext attack. Let us give another example. Total functions of the kind $f : \{0,1\}^* \to \{0,1\}^c$, where c is some constant, are called *hash functions*. If n is the length of the input message and $n > c$, then there exist such $x_1, x_2, x_1 \neq x_2$, that $f_n(x_1) = f_n(x_2)$. Such a pair x_1, x_2 is called a collision of a hash function f. A cryptographic hash function is considered compromised if one is able to find collisions of that function in reasonable time.

SAT-based cryptanalysis is a quite new direction in cryptanalysis and its basic paradigm suggests that specific inversion problem of a considered function is reduced to a SAT instance. Recall, that Boolean Satisfiability Problem (SAT) consists in the following: for an arbitrary Boolean formula to decide whether it is satisfiable or not. Using the ideas by S.A. Cook [2] and J.C. King [6] it can be shown that for an arbitrary function f_n from the class described above, the corresponding inversion problem can be effectively reduced to SAT for some satisfiable CNF.

The main result of our paper is the TRANSALG software system designed specifically to construct SAT encodings for cryptographic functions and apply to constructed encodings state-of-the-art SAT solvers. It is based on the concept of symbolic execution [6]. The features of TRANSALG translation procedures were described in [9]. In the present paper we show how TRANSALG can be applied to solving inversion problems of several cryptographic functions and compare its effectiveness with that of similar systems: GRAIN OF SALT [11], URSA [5], and CRYPTOL+SAW [4].

Let us now point out key differences between TRANSALG and aforementioned systems. The distinctive feature of TRANSALG is that it can construct and explicitly output the so-called *template CNF* $C(f_n)$. When it constructs $C(f_n)$ it employs the concept of symbolic execution of program A_f fully reflecting this process in the memory of abstract computing machine. As a result, in a template CNF $C(f_n)$ all elementary operations with the memory of abstract machine are represented in the form of Boolean equations over sets of Boolean variables. TRANSALG makes it possible to work with these variables directly, thus providing a number of useful features for cryptanalysis. In particular, we can quickly generate families of cryptographic instances: to make one SAT instance for function inversion it is sufficient to add to a template CNF unit clauses encoding the corresponding output. That is why template CNFs are very handy when one uses partitioning strategy to solve some hard SAT instance in a distributed computing environment. Also TRANSALG can identify variables corresponding to inputs and outputs of considered function, so external tools can be used to check correctness of SAT encodings and to analyze the results of solving SAT. In particular, thanks to this we can use any SAT solvers and preprocessors. TRANSALG allows to monitor the values of program variables inside program A_f at any step of computing, and, therefore, to assert any conditions on these variables. For example, thanks to this it is easy to write in a program A_f the conditions specifying the differential path for finding collisions of cryptographic hash functions. In other considered systems (URSA, Cryptol) there arise significant difficulties when writing such conditions. Finally, let us note that the connec-

[1] Matrosov Institute for System Dynamics and Control Theory of SB RAS, Irkutsk, Russia, email: otilya@yandex.ru, biclop.rambler@yandex.ru, the42dimension@gmail.com, zaikin.icc@gmail.com, veinamond@gmail.com

tion between the structure of CNF $C(f_n)$ and an original algorithm, reflected by TRANSALG, can play an important role in implementation of several cryptographic attacks (such as guess-and-determine attacks [1]) in parallel.

2 SAT-BASED CRYPTANALYSIS OF SEVERAL CRYPTOGRAPHIC SYSTEMS USING TRANSALG

In the first series of experiments we considered SAT-based cryptanalysis of the Bivium, Trivium and Grain keystream generators. Note that similar problems were studied earlier in [3, 11, 12]. In accordance with these papers we considered the inversion problems for the following functions: $f^{Bivium} : \{0,1\}^{177} \to \{0,1\}^{200}$, $f^{Grain} : \{0,1\}^{160} \to \{0,1\}^{160}$, $f^{Trivium} : \{0,1\}^{288} \to \{0,1\}^{300}$.

In our experiments we assumed that a number of bits from the secret key are known. To these bits we will below refer as *guessing bits* [1]. In other words, assume that we consider the inversion problem of function $f_n : \{0,1\}^n \to \{0,1\}^m$ in some point $y \in Range f_n$. Let $C(f_n)$ be the template CNF for f_n and let X^{in} be the set of variables encoding the input of f_n. Let us choose as the set of guessing bits some set X', $X' \subseteq X^{in}$. By *GeneratorK* we mean the SAT instances which encode cryptanalysis of the corresponding generator, modified by assigning values to variables from set X',$|X'| = K$. Essentially, *GeneratorK* means a series of SAT instances that differ in values of variables from X'. We considered such series of 100 instances each. On instances from each series we ran state-of-the-art CDCL SAT solvers that rated high in SAT competition 2014. In case of the encodings produced by URSA we were forced to use only the solvers CLASP and ARGOSAT embedded into this system. On average all considered tools constructed SAT encodings with more or less similar solving time (for each system we tracked the best results using all solvers). In terms of SAT instances solved within the time limit of one hour, TRANSALG lost to competition at most 6% on Bivium30 and Trivium142, but won at least 30% on Grain102 .

Earlier we applied TRANSALG to perform the SAT-based cryptanalysis of the widely known A5/1 keystream generator, that is still used to cipher GSM traffic in many countries. In detail the corresponding computational experiment was described in [10]. We managed to solve non-weakened cryptanalysis instances for this generator in a specially constructed distributed environment. Note, that it was possible to effectively parallelize this problem thanks to the functional features of TRANSALG outlined above.

The next cryptanalysis problem we considered was the problem of finding collisions of cryptographic hash functions MD4 and MD5. The first successful application of SAT-solvers to this problem was described in [8]. The authors of [8] note that to find one MD4 collision it took them about 10 minutes (500 seconds), while finding MD5 collisions proved to be more difficult. The SAT encodings for the corresponding problems constructed using TRANSALG system were much more compact than that from [8]. In our experiments in less than 500 seconds it was possible to generate 1000 MD4 collisions using one mainstream processor core (with the help of CRYPTOMINISAT SAT solver [12]). To find two-block MD5 collisions we employed PLINGELING and TREENGELING SAT solvers. With their help we found several dozens two-block MD5 collisions with first 10 zero bytes. Note, that we implemented a SAT variant of differential attack by X.Wang et al. [13]. With this purpose we added to template CNF additional constraints encoding the differential path. In the TRANSALG system thanks to its translation concept this step can be performed effectively, while in other systems it requires significant amount work to be implemented.

Finally we compared the effectiveness of SAT and SMT approaches to inversion of cryptographic functions. We performed SAT-based cryptanalysis of Geffe generator [7], where the mixing function was $majority(x_1, x_2, x_3)$. We considered the cryptanalysis problem in the following form: to find 96-bit secret key using the known keystream fragment of length 200 bits. We constructed a series of 100 instances of this kind. Each instance was considered both as SAT and as SMT. The SAT encodings were constructed using TRANSALG system and were solved using MINISAT solver. The SMT encoding were constructed using CRYPTOL+SAW and solved using SMT-solvers BOOLECTOR, YICES, CVC4 and Z3. In the considered series of experiments SMT approach lost to SAT in both the number of problems solved within 1 minute (83 vs 100) and in average time on solved instances (35 seconds vs 7).

The extended version of this paper can be found online [2].

ACKNOWLEDGEMENTS

The research was funded by Russian Science Foundation (project no 16-11-10046).

REFERENCES

[1] Gregory V. Bard, *Algebraic Cryptanalysis*, Springer Publishing Company, Incorporated, 1st edn., 2009.

[2] Stephen A. Cook, 'The complexity of theorem-proving procedures', in *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pp. 151–158, (1971).

[3] Tobias Eibach, Enrico Pilz, and Gunnar Völkel, 'Attacking Bivium using SAT solvers', in *SAT*, eds., Hans Kleine Büning and Xishun Zhao, volume 4996 of *Lecture Notes in Computer Science*, pp. 63–76. Springer, (2008).

[4] Levent Erkök and John Matthews, 'High assurance programming in cryptol', in *Fifth Cyber Security and Information Intelligence Research Workshop, CSIIRW '09, Knoxville, TN, USA, April 13-15, 2009*, eds., Frederick T. Sheldon, Greg Peterson, Axel W. Krings, Robert K. Abercrombie, and Ali Mili, p. 60. ACM, (2009).

[5] Predrag Janicic, 'URSA: a System for Uniform Reduction to SAT', *Logical Methods in Computer Science*, **8**(3), 1–39, (2012).

[6] James C. King, 'Symbolic execution and program testing', *Commun. ACM*, **19**(7), 385–394, (July 1976).

[7] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot, *Handbook of Applied Cryptography*, CRC Press, Inc., Boca Raton, FL, USA, 1st edn., 1996.

[8] Ilya Mironov and Lintao Zhang, 'Applications of SAT solvers to cryptanalysis of hash functions', in *SAT*, eds., Armin Biere and Carla P. Gomes, volume 4121 of *Lecture Notes in Computer Science*, pp. 102–115. Springer, (2006).

[9] Ilya Otpuschemmikov, Alexander Semenov, and Stepan Kochemazov, 'Transalg: a tool for translating procedural descriptions of discrete functions to SAT', in *WCSE 2015-IPCE: Proceedings of The 5th International Workshop on Computer Science and Engineering: Information Processing and Control Engineering*, pp. 289–294, (2015).

[10] Alexander Semenov and Oleg Zaikin, 'Algorithm for finding partitionings of hard variants of boolean satisfiability problem with application to inversion of some cryptographic functions', *SpringerPlus*, **5**(1), 1–16, (2016).

[11] Mate Soos, 'Grain of Salt - an automated way to test stream ciphers through SAT solvers', in *Tools'10: Proceedings of the Workshop on Tools for Cryptanalysis*, pp. 131–144, (2010).

[12] Mate Soos, Karsten Nohl, and Claude Castelluccia, 'Extending SAT solvers to cryptographic problems', in *SAT*, ed., Oliver Kullmann, volume 5584 of *Lecture Notes in Computer Science*, pp. 244–257. Springer, (2009).

[13] Xiaoyun Wang and Hongbo Yu, 'How to break MD5 and other hash functions', in *Advances in Cryptology - EUROCRYPT 2005, Proceedings*, pp. 19–35, (2005).

[2] http://arxiv.org/abs/1607.00888

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1596

Explanatory Diagnosis of an Ontology Stream via Reasoning About Actions

Quan Yu[1] and **Hai Wan**[2] and **Jiangtao Xu**[2] and **Freddy Lécué**[3] and **Liang Chang**[4]

Abstract. Explanatory diagnosis of an ontology stream aims to explain the changes hidden in the ontology stream by a sequence of actions. In this paper, we present a framework for explanatory diagnosis of an ontology stream, which allows the actions to be uncertain. In order to capture the semantics of actions, we introduce a new update operator and effect-guided bold-repair. By combining these operators with a query mechanism of description logics $\mathcal{EL}^{++}$ supporting inconsistency-tolerant semantics, we present a formal definition for the explanatory diagnosis problem of ontology streams.

1 Introduction and Related Work

The task of diagnosing streams has received particular attentions from the semantic web and diagnosis communities. Freddy Lécué proposed a framework for diagnosing anomalies in an *ontology stream* [3]. However, the task of *explanatory diagnosis* is not achieved on stream evolution and its changes. McIlraith presented a formal characterization of explanatory diagnosis in the situation calculus [5]. Yu et al. also provided a formal characterization of explanatory diagnosis in dynamic epistemic logic [7]. In this paper, we focus on the task of explanatory diagnosis of an ontology stream, which uses a sequence of actions to explain changes occurring over time.

In a road traffic context, a change is a transition from "*cleared*" to "*congested*" road in a particular time of the stream and its diagnoses can be represented as an action sequence *e.g.*, a road work, road incident. These explanations can be derived from different points of time and events between them, from which we can indirectly derive a new ontology stream of diagnoses. We introduce actions with either certain or uncertain effects. For the update of an ontology, we present a new update operator and an effect-guided bold-repair operator. Combined with (i) query mechanisms, supporting inconsistency-tolerant semantics, and (ii) both the semantics and dynamics of action sequences, we formalize the explanatory diagnosis problem in the framework of description logics (DL) $\mathcal{EL}^{++}$.

2 Background

Our approach is illustrated with DL $\mathcal{EL}^{++}$ [3], which is the basis of many more expressive DLs. As usual DLs, answers to queries are formed by constants/ terms denoting individuals explicitly mentioned in the ABox [1]. The definition of Union of Conjunctive Queries (UCQ) and Extension of UCQ (ECQ) can found in [1]. An ontology stream [2] is defined as a dynamic and evolutive version of ontologies. An ontology stream $\mathcal{O}_m^n$ from point of time m to point of time n is a sequence of ontologies $(\mathcal{O}_m^n(m), \mathcal{O}_m^n(m+1), \cdots, \mathcal{O}_m^n(n))$ where $m, n \in \mathbb{N}$ and $m < n$.

2.1 Actions and Events

We first present the definitions of certain-effect action.

Definition 1 (Certain-effect Action) *A certain action α is a pair $(pre, effs)$ where:*

- *pre is the precondition of α, which is an ECQ;*
- *$effs$ is the set of effects of α. Each effect eff has the form $Q \rightsquigarrow F$ where (i) Q is an ECQ, (ii) F is a set of facts to be added to the ABox. It is a set of non-ground ABox assertions which include constants in the initial ABox A_0 and free variables of Q.*

Our definition of certain-effect action is an adaptation of that in [1]. Then we present the uncertain effect action as follows:

Definition 2 (Uncertain-effect Action) *An uncertain-effect action α_u is a pair $(pre, Effs)$ where:*

- *pre is the precondition of α_u, which is an ECQ;*
- *$Effs = (p, effs)$ is a set of pairs. p describes the probability of $effs$ after executing action α_u, which means that each action α_u has many action effects with different probability.*

If not otherwise specified, we slightly abuse these two types of actions. We denote the precondition of α as pre_α, the effects of α as $effs_\alpha$, and each effect of α as eff_α. Note that the free variables of pre_α are the parameters of action α. We use $\alpha[\theta] = (pre_\alpha\theta, effs_\alpha\theta)$ to denote the action is substituted by partial substitution θ. We complement our framework by introducing the concept of events which decide whether an action is applicable in a current knowledge base.

Definition 3 (Event) *An event e_i, occurring in time i of ontology stream $\mathcal{O}_m^n$, is a set of non-ground ABox assertions which include constants and free variables. We use $\mathcal{E}_{m+1}^{n-1}(i)$ to denote events occurring in time $i \in (m, n)$.*

2.2 Update Operator

In the following, we introduce the definitions of (i) applicability for actions in an ABox, and (ii) update operator for describing how to update an ontology with an action.

Definition 4 (Applicable) *Given (i) a terminology T, (ii) an action $\alpha[\theta] = (pre_\alpha\theta, effs_\alpha\theta)$, (iii) the current ABox A denoted by the empty set, and (iv) an event e. $\alpha[\theta]$ is applicable in A according to e, if $A \cup e$ is consistent w.r.t. T and $\exists\sigma$ s.t.$T, A \cup e\theta\sigma \models_{BR} pre_\alpha\theta\sigma$ (The notion of $\models_{BR}$ will be introduced in the next subsection).*

[1] School of Mathematics and Statistics, Qiannan Normal University for Nationalities, China. email: yuquanlogic@126.com
[2] School of Data and Computer Science, Sun Yat-sen University, China. Hai Wan is the corresponding author, email: wanhai@mail.sysu.edu.cn
[3] Accenture Tech Labs, Ireland. INRIA, France.
[4] School of Computer Science and Information Security, Guilin University of Electronic Technology, China.

Definition 5 (Update with Action) *Given an action* $\alpha[\theta] = (pre_\alpha\theta, effs_\alpha\theta)$*, and the current* $O = (T, A)$*. The results of application* $\alpha[\theta]$ *in* O *is a new ontology* $O' = (T, A')$*, which is defined as* $O' = A \circledast \alpha[\theta]$*, where* $A \circledast \alpha[\theta] = A \cup E_\alpha$*, and* $E_\alpha = \bigcup_{(Q \rightsquigarrow F) \in effs_\alpha} \bigcup_{\sigma \in ANS(Q\theta, T, A)} F\theta\sigma$.

2.3 Repairs

Based on the defintion of Bold-repair [4], we present the definition of *effect-guided bold-repair* to fit our framework. While we use an action α to update an ABox A_0, the resulting ABox A may become inconsistent. Suppose that the effects E_α is with a higher priority, we repair the old ABox A_0 according to the fixed TBox T and E_α.

Definition 6 (Effects Guided Bold-repair) *An effects guided bold-repair (*B_{rep} *in short) of an ABox* A *and an action* α*, denoted as* $B_{rep}(A, \alpha)$*, is a subset* A' *of* $A - E_\alpha$ s.t.*:*

- $T \cup E_\alpha \cup A'$ *has a model;*
- *there does not exists* A'' *such that* $A' \subset A'' \subseteq A - E_\alpha$ *and* $T \cup E_\alpha \cup A''$ *has a model.*

We use the defintion of inconsistency-tolerant semantics to decide if the precondition holds before updating the ontology by an action.

Definition 7 (Inconsistency-tolerant Semantics) *Let* $KB(T, A)$ *be an* $\mathcal{EL}^{++}$ *knowledge base,* α *be an action, and* q *be an UCQ. We say that* q *is* BR *entailed by* $KB(T, A)$*, written* $T, A \models_{BR} q$*, if* $T, A' \models q$ *for every* B_{rep} A' *of* A *according to action* α.

Theorem 1 *Let* $KB(T, A)$ *be an* $\mathcal{EL}^{++}$ *knowledge base, let* α *be an action, and let* q *be an UCQ. Then the complexity of deciding whether* $T, A \models_{BR} q$ *is in* EXPTIME.

Theorem 1 tells us that the complexity of query an UCQ over an $\mathcal{EL}^{++}$ knowledge base under the inconsistency-tolerant semantics is in EXPTIME. The complexity of ECQ will not change.

Theorem 2 *Let* $KB(T, A)$ *be an* $\mathcal{EL}^{++}$ *knowledge base, let* α *be an action, and let* Q *be an ECQ. Then the complexity of deciding whether* $T, A \models_{BR} Q$ *is in* EXPTIME.

3 Explanatory Diagnosis of an Ontology Stream

3.1 Ontology Tree

Because the evolution from $\mathcal{O}_m^n(i)$ to $\mathcal{O}_m^n(i+1)$ of an ontology stream $\mathcal{O}_m^n$ may be caused by the occurrence of one or several actions, we use *ontology tree* to capture the dynamics of semantics of an ontology stream. An ontology tree is a tree where each vertex is an ontology, each solid directed edge means an action, and each dotted directed edge indicates the effect of the action, the endpoint of which is the result of the updating the start point by an action.

3.2 Explanatory Diagnosis Problem

In our framework, fault concept is a special kinds of concept from a TBox, which is used for describing some phenomenons that we monitor from a stream perspective e.g., a congested road.

Definition 8 (Anomaly) *Given a* $KB(T, A)$*, suppose* $\mathcal{C}$ *is a set of predefined fault concepts. An assertion* $C(a)$ *is called as an anomaly, if* $C \in \mathcal{C}$*. Every concept in* $\mathcal{C}$ *is called a faulty concept.*

Definition 9 (Explanatory Diagnosis Problem) An explanatory diagnosis problem $\mathcal{P}$ is a tuple $\langle \mathcal{O}_m^n(m), \mathcal{O}_m^n(n), A_s, \mathcal{E}_{m+1}^{n-1}, C(a)\rangle$ which satisfies $\mathcal{O}_m^n(m) \not\models C(a)$ and $\mathcal{O}_m^n(n) \models C(a)$, where

- A_s is the set of actions, including certain effect actions and uncertain effect actions;
- $\mathcal{E}_{m+1}^{n-1}$ is the set of events which occur from time m to n;
- $C(a)$ is an anomaly, which also called as an observation.

Definition 10 (Diagnosis) *A diagnosis of a given explanatory diagnosis problem* $\mathcal{P} = \langle \mathcal{O}_m^n(m), \mathcal{O}_m^n(n), A_s, \mathcal{E}_{m+1}^{n-1}, C(a)\rangle$ *is a sequence of actions* δ*, such that* $B_{rep}(\mathcal{O}_m^n(m) \circledast \delta) \models C(a)$.

It should be noted that, for any nonempty action sequence δ and an action α, we have $B_{rep}(A_0 \circledast \alpha\delta) = B_{rep}((B_{rep}(A_0 \circledast \alpha)) \circledast \delta)$.

Theorem 3 *Explanatory diagnosis existence for* $\mathcal{EL}^{++}$ *with a finite individual domain is decidable in* EXPTIME *in the size of the individual domain.*

We assume familiarity with standard notions of Markov Decision Process (MDP) and value iteration [6]. A solution to a MDP is an optimal policy that maximizes value function of every state $s \in S$.

Considering that the set of action sequences is exponentially growing on a time basis, we can use MDP to compute the most likely action sequence, *i.e.*, diagnosis. Given an explanatory diagnosis problem $\mathcal{P}$, we can first generate the ontology tree, then solving problem $\mathcal{P}$ can be convert into a largest value path finding problem. It's easy to prove that there is a diagnosis δ which has the largest value function in all diagnoses as the following proposition.

Proposition 1 *Each diagnosis of an explanatory diagnosis problem* $\mathcal{P}$ *is an ontology stream.*

4 Conclusion and Future Work

We have developed a framework of explanatory diagnosis of an ontology stream via reasoning about actions. The semantics is captured by DL evolving over time in an ontology stream. We introduced a new update operator, modeling actions with certain and uncertain effects, and presented query mechanisms supporting inconsistency-tolerant semantics. Future work will extend our framework towards scalable explanatory diagnoses, specialty for large cities.

5 Acknowledgement

The authors is supported by NSF Grant Nos.(61463044, 61363030, and 61573386) and Grant No.LH[2014]7421 from the Joint Fund of the NSF of Guizhou province of China.

References

[1] Babak Bagheri Hariri, Diego Calvanese, Marco Montali, Giuseppe De Giacomo, Riccardo De Masellis, and Paolo Felli, 'Description logic knowledge and action bases', *JAIR*, **46**, 651–686, (2013).
[2] Zhisheng Huang and Heiner Stuckenschmidt, 'Reasoning with multi-version ontologies: A temporal logic approach', in *Proc. of ISWC*, pp. 398–412, (2005).
[3] Freddy Lécué, 'Towards scalable exploration of diagnoses in an ontology stream', in *Proc. of AAAI*, pp. 87–93, (2014).
[4] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo, 'Inconsistency-tolerant semantics for description logics', in *Proc. of RR*, pp. 103–117, (2010).
[5] Sheila A. McIlraith, 'Explanatory diagnosis: Conjecturing actions to explain observations', in *Proc. of KR*, pp. 167–179, (1998).
[6] M. L Puterman, *Markov decision processes: Discrete stochastic dynamic programming*, Wiley, 1994.
[7] Quan Yu, Ximing Wen, and Yongmei Liu, 'Multi-agent epistemic explanatory diagnosis via reasoning about actions', in *Proc. of IJCAI*, pp. 1183–1190, (2013).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1598

DARDIS: Distributed And Randomized DIspatching and Scheduling

Thomas Bridi [1] and **Michele Lombardi** [1] and **Andrea Bartolini** [2 3] and **Luca Benini** [2 3] and **Michela Milano** [1]

Abstract. Scheduling and dispatching are critical enabling technologies in supercomputing and grid computing. In these contexts, scalability is an issue: we have to allocate and schedule up to tens of thousands of tasks on tens of thousands of resources. This problem scale is out of reach for complete and centralized scheduling approaches.

We propose a distributed allocation and scheduling paradigm called DARDIS that is lightweight, scalable and fully customizable in many domains. In DARDIS each task offloads to the available resources the computation of a probability index associated with each possible start time for the given task on the specific resource. The task then selects the proper resource and start time on the basis of the above probability.

1 Introduction

Large-scale computing infrastructure like grids and High-Performance Computing (HPC) facilities require efficient workload scheduling and dispatching.

Consider for example the number of computational nodes a scheduler has to manage for high-performance computers like the top 1 HPC in 2015 or the future exascale HPC. This machine features a number of nodes estimated between 50'000 and 1'000'000 [3]. Classical job schedulers are rule-based. These are heuristic schedulers that use rules to prioritize jobs. In these scheduling systems, a job requests a set of resources on which the job will execute. The scheduler checks for each job if it can execute on a node while respecting the capacity of the target resources. If the job can use the requested amount of resources, the job is executed. It is quite clear that for these large scale machines a centralized, optimization-based scheduler [2, 1], is not a feasible option. Hence, scalable, distributed schedulers are needed to handle thousands of nodes while at the same time optimizing efficiency metrics.

This work takes inspiration from Randomized Load Control proposed in [4]. We substantially extend this work to the case of multiple resources and introduce new start times generators and dispatching policies.

In this work we present a Distributed And Randomized DIspatching and Scheduling (DARDIS) approach that is:

- **Distributed** to scale to an ultra-large system. The scheduler and dispatcher basically leave the dispatching choice to the task and each resource then schedules its own tasks.
- **Supporting variable resources' utilization profile**. Each resource, besides its capacity, exhibits a (variable) desired utilization profile.
- **Randomized**. The scheduler can choose the proper probability distribution for selecting resources and start times to optimize different objective functions.
- **Deadlines aware**. Each activity can specify a time window in which it should start.

Tests against classical commercial scheduler, on standard job traces, show an improvement of 2.6% in makespan and 18% in the total job waiting while having a scheduler 42 times faster in term of computational overhead.

2 Approach

The workload dispatching and scheduling problem can be modeled by a set of resources res_r, with $r \in R$ and a set of activities a_i with $i \in A$. Each resource has a capacity c_r, a desired profile $dp_r(t)$ and a utilization profile $up_r(t)$, with $t \in [0, .., Eoh]$. The desired profile is a profile decided by the administrator that shows how the resource should be used (in term of number of amount of resource used by activities) in time. The utilization profile is the amount of resource already used and reserved to scheduled activities. This profile is a periodic profile repeated in time. As in example for HPC and grid computing this could be a daily utilization profile.

Each activity is submitted to the system in a time instant q_i. At the submission it specifies its earliest start time est_i, the latest start time lst_i, its duration wt_i, and the amount of resource required req_i.

The scheduling problem consists of allocating each activity to a given resource and assigning it a start time st_i and a resource ur_i such that

$$
\begin{aligned}
& st_i :: [est_i..lst_i] \; \forall i \in A \\
& ur_i \in res_r \; \forall i \in A, r \in R \\
& \sum_i req_i \leq c_r \; \forall i \in A | st_i \leq t \wedge st_i + wt_i > t, \forall t \in [0, .., Eoh] \\
& \sum_i req_i \leq dp_r(t) \; \forall i \in A | st_i \leq t \wedge st_i + wt_i > t, \forall t \in [0, .., Eoh]
\end{aligned}
\tag{1}
$$

The main idea is to partition the decision process in two main phases performed by two separate software entities: the task agent and the resource manager. The *task agent* is responsible for the activity submission and the dispatching. This agent resides into the user-space. The *resource manager* is responsible for the scheduling. This agent resides into the resources host.

[1] DISI, University of Bologna, Viale Risorgimento 2, 40123, Bologna, Italy. {thomas.bridi,michele.lombardi2,michela.milano}@unibo.it
[2] DEI, University of Bologna, Viale Risorgimento 2, 40123, Bologna, Italy. {a.bartolini,luca.benini}@unibo.it
[3] Integrated Systems Laboratory, ETH Zurich, Switzerland.

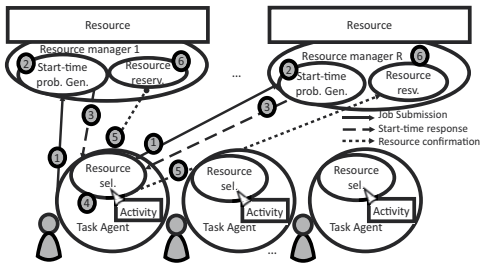


Figure 1: DARDIS architecture and phases (number ordering corresponds to time progression)

Figure 1 shows the different phases of our approach. These phases are subdivided in:

Job Submission (1) - Our approach starts with a task agent submitting an activity to all the resource managers of the system. After the submission to all the resource managers, the task agent waits for the responses.

Start time probability generation (2) - Each resource manager receives the submitted activity and starts the start time probability generation phase in which the manager generates a start time for the activity according to an internal rule (Section 2.1).

Start time response (3) - After the start time generation, the resource manager sends a generated start time to the task agent.

Resource selection (4) - The task agent, after receiving the responses from all the resources, applies a policy (Section 2.2) to select the resources for the activity execution.

Resource confirmation (5) - The task agent sends the result to all the resource managers involved in the submission, namely, the one selected and those not selected.

Resource reservation (6) - The resource managers in which the activity has to execute, reserves the proper capacity for the execution, by modifying the utilization profile.

2.1 Start time probability generation

The start time generation process for the resource j starts by computing a fitting index for the submitted activity i. This index indicates how many parallel runs of the same activity could be executed in a given start time s while satisfying the desired utilization profile for the resource. Due to the variability over time of the desired profile, we have to check for each time instant $t \in \{s, .., s + wt_i\}$ how many times the activity's resource requirement req_i can fit the space left between the utilization profile and the desired profile (equation 2).

$$I'(s) = min_t(\frac{dp_j(t) - up_j(t)}{req_i}) \; \forall t \in \{s, .., s + wt_i\} \qquad (2)$$

Note that $I'(s) = 1$ means that the activity perfectly fits into the resource without exceeding the desired profile. $I'(s) > 1$ means that the activity fits the desired profile and leaves some resource for other activities. If $I'(s) < 1$, it means that the activity exceeds the desired profile. The capacity instead cannot be exceeded by definition. To handle this case, we use equation 3. Where $\setminus$ represents integer division.

$$I(s) = min(I'(s), min_t((c_j - up_j(t)) \setminus req_i)) \forall t \in \{s, .., s + wt_i\} \qquad (3)$$

The index distribution I is calculated for each possible start time between the earliest start time est_i and the latest start time lst_i of the activity: $I = \{I(est_i), .., I(lst_i)\}$. In this way, we obtain the fitting profile for the activity.

We defined three generators for the start time selection:

- **First**: the goal of this star-time generation procedure is to maximize the throughput of the entire system. This deterministic selection works by picking up the first feasible start time.
- **Uniform**: the goal of this generator is to produce a scheduler that allocates resources following the shape of the desired profile for its entire window. This is a probabilistic selection that chooses the start times randomly.
- **Exponential**: this generator has been designed to reach a trade-off between throughput and profile chase. This is a probabilistic generator that chooses a start time following an exponential distribution.

2.2 Resource selection

After the start time generation process, the task agent receives the responses from all the resource managers involved in the submission. This algorithm selects the resources for the activity execution.

The designed policies are:

- **MIN_START**: it selects the resource that will execute the activity first. This approach goes in the direction of optimizing the activity throughput.
- **MAX_PROB**: it selects the resource that gives the highest fitting index. This means that it selects the most unloaded resource. This approach is designed to minimize the standard deviation from the desired profile.
- **MIN_PROB**: it selects the resource that gives the lowest fitting index. This policy is designed to ensure the best fitting for the desired profile. This is useful when we have to prefer solutions that saturate one resource before starting filling another one.
- **RANDOM**: it selects randomly the resources using a uniform. This policy is designed to enforce each resource to have the same probability of hosting an activity.

ACKNOWLEDGEMENTS

This work was partially supported by the FP7 ERC Advance project MULTITHERMAN (g.a. 291125), by the YINS RTD project (no. 20NA21 150939), evaluated by the Swiss NSF and funded by NanoTera.ch with Swiss Confederation financing and by CINECA.

REFERENCES

[1] Andrea Bartolini, Andrea Borghesi, Thomas Bridi, Michele Lombardi, and Michela Milano, 'Proactive workload dispatching on the eurora supercomputer', in *Principles and Practice of Constraint Programming*, ed., Barry OSullivan, Lecture Notes in Computer Science, Springer International Publishing, (2014).

[2] Thomas Bridi, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini, 'A constraint programming scheduler for heterogeneous high-performance computing machines', *IEEE Transactions on Parallel & Distributed Systems*, (2016).

[3] JF Lavignon et al. Etp4hpc strategic research agenda achieving hpc leadership in europe. `http://www.etp4hpc.eu/wp-content/uploads/2013/06/ETP4HPC_book_singlePage.pdf`, 2013.

[4] Menkes Van Den Briel, Paul Scott, and Sylvie Thiébaux, 'Randomized load control: A simple distributed approach for scheduling smart appliances', in *Proceedings of the 23th international joint conference on Artificial Intelligence*. AAAI Press, (2013).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1600

Reasoning About Belief and Evidence with Extended Justification Logic

Tuan-Fang Fan[1] and **Churn-Jung Liau**[2]

1 Introduction

While modal logic has been a standard approach for reasoning about knowledge and belief of intelligent agents [7, 11] since the seminal work by Hintikka [10], the notion of justification, which is an essential component in Plato's tripartite definition of knowledge as *justified true belief*, was largely ignored in the formalisms. Because the formula $\Box\varphi$ is interpreted as "φ is believable" or "φ is knowable" in the epistemic/doxastic reading of modal logics[3], explicit justifications are not represented in the logic. By contrast, justification logics (JL) supply the missing component by adding justification terms to epistemic formulas [5, 2, 4, 8]. The first member of the JL family is the logic of proofs (LP) proposed in [1]. Although the original purpose of LP is to formalize the Brouwer-Heyting-Kolmogorov semantics for intuitionistic logic and establish the completeness of intuitionistic logic with respect to this semantics, in a more general setting, JL has evolved into a kind of explicit epistemic logic and received much attention in computer science and AI [2, 5].

Currently, the most prominent semantics of JL is based on Fitting models, which are essentially extensions of Kripke models for epistemic logic with an *admissible evidence function*, i.e., a mapping from justification terms and formulas to states that stipulates in what state the evidence is admissible for the formula. However, Fitting semantics suffers the ambiguous interpretation of a justification formula as justified belief or simply the admissibility relation between evidence and belief. The ambiguity arises mainly from the inadequate expressive power of JL. While a justification formula can represent that a piece of evidence is admissible for the belief, it cannot express whether the evidence has been actually observed. Therefore, to facilitate the more fine-grained distinction between admissible and actual evidence, we must enhance the expressive power of the JL language. To address the issue, we propose a JL that can express the informational contents of justification terms; and the fact that a piece of evidence has been actually observed is definable in such logics. Furthermore, the recent development on dynamic epistemic logic (DEL) has shown that modeling dynamics of information plays a crucial role in epistemic reasoning [12, 13]. As a byproduct, we also show that the proposed logics can be extended to accommodate dynamic evidential reasoning and hence we can easily integrate JL and DEL paradigms.

[1] Department of Computer Science and Information Engineering, National Penghu University of Science and Technology, Penghu 880, Taiwan. email: dffan@npu.edu.tw
[2] Institute of Information Science, Academia Sinica, Taipei 115, Taiwan. email: liaucj@iis.sinica.edu.tw
[3] For the purpose of the paper, the difference between belief and knowledge is not important. Hence, hereafter, we use epistemic reasoning to denote reasoning about any kind of informational attitude for an agent.

2 Justification Logic

We assume the basic familiarity with modal (epistemic) logic [6]. To extend modal logic with explicit justifications, JL provides formal terms built up from constants and variables using various operation symbols. Constants represent justifications for commonly accepted truths—-typically axioms, whereas variables denote unspecified justifications. While different variants of JL allow different operation symbols, most of them contain *application* and *sum*. Specifically, the justification terms and formulas of the basic JL are defined as follows:

$$t ::= a \mid x \mid t \cdot t \mid t + t,$$

$$\varphi ::= p \mid \bot \mid \varphi \to \varphi \mid t{:}\varphi,$$

where $p \in \Phi$, a is a justification constant, and x is a justification variable. We use $\mathcal{L}_J$ to denote the basic JL language and Tm to denote the set of all justification terms.

JL furnishes an evidence-based foundation for epistemic logic by using justification formula $t{:}\varphi$ to denote "t is a justification of φ", or more strictly, "t is accepted as a justification of φ" [2]. Semantically, the formula $t:\varphi$ can be regarded as indicating that t is an admissible evidence for φ and based on the evidence, φ is believed. Thus, the model of JL is the Kripke model enriched with an additional evidence component [2, 8]. This kind of model, called *Kripke-Fitting model* or simply *Fitting model*, is formally defined as a quadruple $\mathfrak{M} = \langle W, R, E, \Vdash\rangle$, where W is a set of possible worlds (states), $R \subseteq W \times W$ is a binary *accessibility relation* on W, $\Vdash \subseteq W \times \Phi$ is a *forcing relation* between possible worlds and propositional symbols such that $w \Vdash p$ means that p is satisfied in w, and E is an *admissible evidence function* such that $E(t,\varphi) \subseteq W$ for any justification term t and formula φ. Intuitively, $E(t,\varphi)$ specifies the set of possible worlds in which t is regarded as admissible evidence for φ. In this paper, we consider the basic JL in which it is required that E must satisfy the closure condition with respect to the application and sum operations:

- Application: $E(s, \varphi \to \psi) \cap E(t,\varphi) \subseteq E(s \cdot t, \psi)$;
- Sum: $E(s,\varphi) \cup E(t,\varphi) \subseteq E(s+t,\varphi)$;

The first condition states that an admissible evidence for $\varphi \to \psi$, which can be regarded as a function that transforms a justification of φ to a justification of ψ, can be applied to an admissible evidence for φ to obtain an admissible evidence for ψ. The second condition guarantees that adding a piece of new evidence does not defeat the original evidence. That is, $s + t$ is still an admissible evidence for φ whenever either s or t is an admissible evidence for φ. The forcing relation $\Vdash$ between W and the justification formula $t{:}\varphi$ satisfies the following condition:

- $w \Vdash t:\varphi$ iff $w \in E(t,\varphi)$ and for any u such that $(w,u) \in R$, $u \Vdash \varphi$.

We will use (F1) and (F2) to denote the conditions "$w \in E(t,\varphi)$" and "for any u such that $(w,u) \in R$, $u \Vdash \varphi$" respectively.

According to the semantics, the following formula is valid and included as a main axiom in the axiomatic system for JL.

$$s{:}\varphi \rightarrow s+t{:}\varphi \text{ and } t{:}\varphi \rightarrow s+t{:}\varphi.$$

3 Problem Statement and the Proposed Solution

In [1], it is claimed that the intended semantics of $t:\varphi$ in LP is "t is a proof of φ", which is exactly captured by the condition (F1) in Fitting semantics. When the justification t represents a proof in some mathematical or logical system, (F1) naturally implies condition (F2) of Fitting semantics. The fact that (F1) implies (F2) is called the principle of *justification yielding belief* (JYB) in [3] and formalized in the modular semantics introduce there. However, when LP is evolving into the more general JL for epistemic reasoning, the principle becomes less convincing because of the two ambiguous interpretations of $t:\varphi$ as "t is an evidence of φ":

1. t is a piece of actually observed evidence for φ: then the JYB principle holds and the truth condition of $t:\varphi$ can be formalized in Fitting semantics. However, in this case, the axiom Sum seems doubtful because t being actually observed does not imply that $t+s$ has been also actually observed.
2. t is regarded as admissible or relevant evidence for φ: it means that φ will be believed once t is observed. However, it does not assert that t has been actually observed. Hence, it is possible that φ is not believed currently. In this case, the axiom Sum is valid but the JYB principle fails. Therefore, condition (F2) in Fitting semantics must be dropped.

According to the analysis above, we can see that the interpretation of a justification formula $t:\varphi$ is ambiguous in JL. From the axiomatic viewpoint, it seems that $t:\varphi$ only stipulates the admissibility of the evidence t with respect to φ no matter whether t is actually observed; whereas from the viewpoint of Fitting semantics, the JYB principle implies that $t:\varphi$ represents a justified belief on φ due to the actual observation of t. The ambiguity arises because, during the evolution from LP to JL, the semantic meaning of justification has been extended from mathematical proof to general evidence; however, the syntax of the language remains unchanged and hence the expressive power is no longer adequate for explicit epistemic reasoning. Therefore, to overcome the problem, we need a more fine-grained language that can differentiate these two interpretations of justification formulas.

The key point to clarify the ambiguity is whether a piece of evidence has been actually observed. The basic idea is that evidence has some informational contents and if a piece of evidence has been observed, then its informational contents should have been assimilated into the current belief. Thus, our language must be extended with modal operators $\Box_t$ to represent the informational contents of t for each justification term t. In addition, we have a special constant ϵ that represent the agent's belief based on the accumulation of evidence so far and a corresponding epistemic operator $\Box_\epsilon$. To represent the fusion of the agent's belief and evidence, we employ the Boolean modal logic (BML) encompassing union, intersection and complement of modalities[9]. Moreover, we need a relational symbol to compare the relative strength of informational contents between different justification terms or between ϵ and justification terms.

To implement these ideas, we enrich JL with modalities that can represent informational contents of accumulated evidence. In the enriched languages, we can clarify the ambiguous interpretation of justification formulas. While the clarification is unnecessary when justification terms are regarded as mathematical proofs as in the case of LP, it become crucially important when LP is evolved into general JL for reasoning about explicit belief. The resultant languages are expressive enough to address the issue of the mismatch between the formal semantics and the intuitive explanation of the Sum axiom. Moreover, as a byproduct, we show that the DEL-like dynamic modalities can be easily integrated into the enriched logics and this leads to logics for dynamic evidential reasoning with justifications.

4 Concluding Remark: A Simple-Minded Solution Doesn't Work

A simple-minded reader may think that an obvious solution to the above-mentioned problem is simply to allow the separate representation of conditions (F1) and (F2) in the logical language. Such an oversimplified proposal is to add formulas of the forms $Ev(t,\varphi)$ and $\Box\varphi$ to JL. Then, if t is actual evidence, then $Ev(t,\varphi) \rightarrow \Box\varphi$ is valid; and if t is potential (but not actual) evidence, then $Ev(t,\varphi) \wedge \neg\Box\varphi$ may be true. However, such system is too poor to represent the possible coexistence of potential and actual evidence at the same time[4]. For example, if s is actual evidence and t is only potential evidence for φ. Then, both $Ev(s,\varphi)$ and $Ev(t,\varphi)$ are true and because s is actual, it implies that $\Box\varphi$ is also true. Hence, it is impossible that $Ev(t,\varphi) \wedge \neg\Box\varphi$ is true and consequently, t must be also actual evidence even though t is not actually observed. In addition, t may also serve as potential evidence for another formula ψ although s is not evidence for ψ. However, because the actuality of s forces the actuality of t, this leads to the ridiculous conclusion of $\Box\psi$ which has no actual justification.

REFERENCES

[1] S. Artemov, 'Explicit provability and constructive semantics', *Bulletin of Symbolic Logic*, **7**, 1–36, (2001).
[2] S. Artemov, 'The logic of justification', *The Review of Symbolic Logic*, **1**, 477–513, (2008).
[3] S. Artemov, 'The ontology of justifications in the logical setting', *Studia Logica*, **100**(1–2), 17–30, (2012).
[4] S. Artemov and M. Fitting, 'Justification logic', in *The Stanford Encyclopedia of Philosophy*, ed., E.N. Zalta, fall 2012 edn., (2012).
[5] S. Artemov and E. Nogina, 'Introducing justification into epistemic logic', *Journal of Logic and Computation*, **15**(6), 1059–1073, (2005).
[6] P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*, Cambridge University Press, Cambridge, United Kingdom, 2001.
[7] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi, *Reasoning about Knowledge*, MIT Press, 1996.
[8] M. Fitting, 'The logic of proofs, semantically', *Annals of Pure and Applied Logic*, **132**(1), 1–25, (2005).
[9] G. Gargov and S. Passy, 'A note on boolean modal logic', in *Mathematical Logic*, ed., P. Petrov, 299–309, Springer, (1990).
[10] J. Hintikka, *Knowledge and Belief*, Cornell University Press, 1962.
[11] J.-J. Ch. Meyer and W. van der Hoek, *Epistemic Logic for AI and Computer Science*, Cambridge University Press, 1995.
[12] J. van Benthem, *Logical Dynamics of Information and Interaction*, Cambridge University Press, 2014.
[13] H. van Ditmarsch, W. van der Hoek, and B. Kooi, *Dynamic Epistemic Logic*, Springer, 2008.

[4] In fact, we had considered this solution at the beginning but finally rejected it because of its poor expressive power. Unfortunately, an anonymous reviewer made the same mistake and argued against the necessity of our sophisticated language based on the simple-minded observation.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1602

Scaling Structure Learning of Probabilistic Logic Programs by MapReduce

Fabrizio Riguzzi[1] and **Elena Bellodi**[2] and **Riccardo Zese**[2]
and **Giuseppe Cota**[2] and **Evelina Lamma**[2]

Abstract. Probabilistic Logic Programming is a promising formalism for dealing with uncertainty. Learning probabilistic logic programs has been receiving an increasing attention in Inductive Logic Programming: for instance, the system SLIPCOVER learns high quality theories in a variety of domains. However, SLIPCOVER is computationally expensive, with a running time of the order of hours. In order to apply SLIPCOVER to Big Data, we present SEMPRE, for "Structure lEarning by MaPREduce", that scales SLIPCOVER by following a MapReduce strategy, directly implemented with the Message Passing Interface.

1 Introduction

Probabilistic Logic Programming (PLP) is an interesting language for Inductive Logic Programming (ILP), because it allows algorithms to better deal with uncertain information. The distribution semantics [5] is an approach to PLP that is particularly attractive for its intuitiveness and for the interpretability of the programs. Various algorithms have been proposed for learning the parameters of probabilistic logic programs under the distribution semantics, such as ProbLog2 [3] and EMBLEM [1]. Recently, systems for learning the structure of these programs have started to appear. Among these, SLIPCOVER [2] performs a beam search in the space of clauses using the log-likelihood as the heuristics.

This system was able to learn good quality solutions in a variety of domains [2] but is usually costly in terms of time.

In this paper, we propose the system SEMPRE for "Structure lEarning by MaPREduce", that is a MapReduce version of SLIPCOVER.

We experimentally evaluated SEMPRE by running it on various datasets using 1, 8, 16 and 32 nodes. The results show that SEMPRE significantly reduces SLIPCOVER running time, even if the speedup is often less than linear because of a (sometimes) relevant overhead.

The paper is organized as follows. Section 2 summarises PLP under the distribution semantics. Section 3 discusses SEMPRE and presents the experiments, while Section 4 concludes the paper.

[1] Dipartimento di Matematica e Informatica – University of Ferrara, Via Saragat 1, I-44122, Ferrara, Italy. Email: fabrizio.riguzzi@unife.it
[2] Dipartimento di Ingegneria – University of Ferrara Via Saragat 1, I-44122, Ferrara, Italy Email: [elena.bellodi, riccardo.zese, giuseppe.cota, evelina.lamma]@unife.it

2 Probabilistic Logic Programming

We introduce PLP focusing on the distribution semantics. We consider Logic Programs with Annotated Disjunctions (LPADs) as the language for their general syntax and we do not allow function symbols; for the treatment of function symbols see [4].

An LPAD is a finite set of annotated disjunctive clauses of the form $h_{i1} : \Pi_{i1}; \ldots; h_{in_i} : \Pi_{in_i} \text{ :- } b_{i1}, \ldots, b_{im_i}.$ where $b_{i1}, \ldots, b_{im_i}$ are literals forming the *body*, $h_{i1}, \ldots h_{in_i}$ are atoms whose disjunction forms the *head* and $\Pi_{i1}, \ldots, \Pi_{in_i}$ are real numbers in the interval $[0, 1]$ s.t. $\sum_{k=1}^{n_i} \Pi_{ik} \leq 1$. If $\sum_{k=1}^{n_i} \Pi_{ik} < 1$, the head contains an extra atom *null* absent from the body of every clause annotated with $1 - \sum_{k=1}^{n_i} \Pi_{ik}$

Given an LPAD P, the grounding $ground(P)$ is obtained by replacing variables with terms from the Herbrand universe in all possible ways. If P does not contain function symbols and P is finite, $ground(P)$ is finite as well. $ground(P)$ is still an LPAD from which we can obtain a normal logic program by selecting a head atom for each ground clause. In this way we obtain a so-called *world* to which we can assign a probability by multiplying the probabilities of all the head atoms chosen. We thus get a probability distribution over worlds from which we can define a probability distribution over the truth values of a ground atom: the probability of an atom q being true is the sum of the probabilities of the worlds where q is true[3].

3 Distributed Structure Learning

SEMPRE parallelizes three operations of the structure learning algorithm SLIPCOVER [2] by employing n workers, one master and $n - 1$ slaves.

The first operation is the scoring of the clause refinements [lines 8-14 in Algorithm 1]: when the revisions for a clause are generated, the master process splits them evenly into n subsets and assigns $n-1$ subsets to the slaves. One subset is handled by the master. Then, SEMPRE enters the *Map phase* [lines 15-25], when each worker scores a set of refinements by means of (serial) EMBLEM [1] which is run over a theory containing only one clause. Then, SEMPRE enters the *Reduce phase* [lines 26-31], where the master collects all sets of scored refinements from the workers and updates the beam of promising clauses and the sets of target and background

[3] We assume that the worlds all have a two-valued well-founded model.

Algorithm 1. Function SEMPRE

```
1: function SEMPRE(I, n, NInt, NS, NA, NI, NV, ϵ, δ)
2:     IBs =InitialBeams(I, NInt, NS, NA)                    ▷ Clause search
3:     TC ← [], BC ← []
4:     for all (PredSpec, Beam) ∈ IBs do
5:         Steps ← 1, NewBeam ← []
6:         repeat
7:             while Beam is not empty do
8:                 if MASTER then
9:                     Refs ←ClauseRefinements((Cl, Literals), NV)
10:                    Split evenly Refs into n subsets
11:                    Send Refs_j to worker j
12:                else                                       ▷ the j-th slave
13:                    Receive Refs_j from master
14:                end if
15:                for all (Cl', Literals') ∈ Refs_j do
16:                    (LL'', {Cl''}) ←EMBLEM(I, {Cl'}, ϵ, δ)
17:                    NewBeam_j ←Insert((Cl'', Literals'), LL'')
18:                    if Cl'' is range restricted then
19:                        if Cl'' has a target predicate in the head then
20:                            TC ←Insert((Cl'', Literals'), LL'')
21:                        else
22:                            BC ←Insert((Cl'', Literals'), LL'')
23:                        end if
24:                    end if
25:                end for
26:                if MASTER then
27:                    Collect all the sets NewBeam_j from workers
28:                    Update NewBeam, TC, BC
29:                else                                       ▷ the j-th slave
30:                    Send the set NewBeam_j to master
31:                end if
32:            end while
33:            Beam ← NewBeam, Steps ← Steps + 1
34:        until Steps > NI
35:    end for
36:    if MASTER then
37:        T ← ∅, TLL ← −∞                                    ▷ Theory search
38:        repeat
39:            Remove the first couple (Cl, LL) from TC
40:            (LL', T') ←EMBLEM^MR(I, T ∪ {Cl}, n, ϵ, δ)
41:            if LL' > TLL then
42:                T ← T', TLL ← LL'
43:            end if
44:        until TC is empty
45:        T ← T ⋃_{(Cl,LL)∈BC} {Cl}
46:        (LL, T) ←EMBLEM^MR(I, T, n, ϵ, δ)
47:        return T
48:    end if
49: end function
```

clauses (TC and BC respectively): the scored refinements are inserted in order of LL into these lists.

The second parallelized operation is parameter learning for the theories with the target clauses. In this phase [lines 37-44], each clause from TC is tentatively added to the theory, which is initially empty. In the end, it contains all the clauses that improved its LL (search in the space of theories). In this case, parameter learning may be quite expensive since the theory contains multiple clauses, so a MapReduce version of EMBLEM called EMBLEM$^{\mathrm{MR}}$ is used.

The third parallelized operation is the final parameter optimization for the theory including also the background clauses [lines 45-46]. All the background clauses are added to the theory previously learned and the parameters of the resulting theory are learned by means of EMBLEM$^{\mathrm{MR}}$.

SEMPRE was implemented in Yap Prolog using the `lam_mpi` library for interfacing Prolog with the Message Passing Interface (MPI) framework.

SEMPRE was tested on the following seven real world datasets: Hepatitis, Mutagenesis, UWCSE, Carcinogenesis, IMDB, HIV and WebKB. All experiments were performed on GNU/Linux machines with an Intel Xeon Haswell E5-2630 v3 (2.40GHz) CPU with 8GB of memory allocated to the job.

Figure 1 shows the speedup of SEMPRE as a function of the number of workers. The speedup is always larger than 1 and grows with the number of workers, except for HIV and IMDB, where there is a slight decrease for 16 and 32 workers due to the overhead; however, these two datasets were the smallest and less in need of a parallel solution.

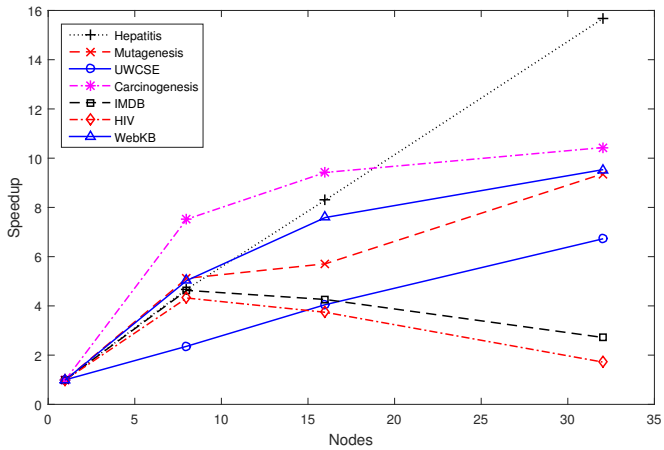


Figure 1. SEMPRE speedup.

4 Conclusions

The paper presents the algorithm SEMPRE for learning the structure of probabilistic logic programs under the distribution semantics. SEMPRE is a MapReduce implementation of SLIPCOVER, exploiting modern computing infrastructures for performing learning in parallel. The results show that parallelization is indeed effective at reducing running time, even if in some cases the overhead may be significant.

Acknowledgement This work was supported by the "GNCS-INdAM".

REFERENCES

[1] Elena Bellodi and Fabrizio Riguzzi, 'Expectation Maximization over Binary Decision Diagrams for probabilistic logic programs', *Intelligent Data Analysis*, **17**(2), 343–363, (2013).

[2] Elena Bellodi and Fabrizio Riguzzi, 'Structure learning of probabilistic logic programs by searching the clause space', *Theory and Practice of Logic Programming*, **15**(2), 169–212, (2015).

[3] Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Sht. Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt, 'Inference and learning in probabilistic logic programs using weighted boolean formulas', *Theory and Practice of Logic Programming*, **15**(3), 358–401, (2015).

[4] Fabrizio Riguzzi and Terrance Swift, 'Well-definedness and efficient inference for probabilistic logic programming under the distribution semantics', *Theory and Practice of Logic Programming*, **13**(Special Issue 02 - 25th Annual GULP Conference), 279–302, (2013).

[5] Taisuke Sato, 'A statistical learning method for logic programs with distribution semantics', in *12th International Conference on Logic Programming, Tokyo, Japan*, ed., Leon Sterling, pp. 715–729, Cambridge, Massachusetts, (1995). MIT Press.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1604

Employing Hypergraphs for Efficient Coalition Formation with Application to the V2G Problem

Filippos Christianos and **Georgios Chalkiadakis**[1]

Abstract. This paper proposes, for the first time in the literature, the use of *hypergraphs* for the efficient formation of effective coalitions. We put forward several formation methods that build on existing hypergraph algorithms, and exploit hypergraph structure to identify agents with desirable characteristics. Our approach allows the near-instantaneous formation of high quality coalitions, while adhering to multiple stated requirements regarding coalition quality. Moreover, our methods are shown to scale to *dozens of thousands* of agents within fractions of a second; with one of them scaling to even *millions* of agents within seconds. We apply our approach to the problem of forming coalitions to provide *(electric) vehicle-to-grid (V2G)* services. Ours is the first approach able to deal with *large-scale*, *real-time* coalition formation for the V2G problem, while taking *multiple criteria* into account for creating electric vehicle coalitions.

1 Introduction

Coalition formation (CF) is a paradigm widely studied in multiagent systems and economics, as means of forming teams of autonomous, rational agents working towards a common goal [1]. One domain where the formation of coalitions comes naturally into play is the so-called *vehicle-to-grid (V2G)* problem. In V2G, battery-equipped *electric vehicles (EVs)* communicate and strike deals with the electricity Grid in order to either lower their power demands, or return power back to the network when there is a peak in the request for power. This helps the Grid to maintain a balanced power load [4].

As such, several recent approaches have called for the formation of EV coalitions in order to tackle the V2G problem [2]. The existing approaches, however, typically exhibit the following characteristics: *(a)* they attempt to form *optimal* coalitions or coalition structures (i.e., partitions of the agents space); and *(b)* they either attempt to form coalitions with respect to a single criterion, or employ lengthy negotiation protocols in order to capture various coalitional requirements while respecting the constraints of individual agents.

The inherent hardness of the optimal coalition structure generation problem [3], however, and the fact that negotiation protocols can be lengthy and thus highly time consuming, severely restricts the practicality and scalability of such algorithms: they can handle at most a few hundred EVs. In reality though, there exist hundreds of thousands of EVs that connect to the Grid, and could potentially offer their services; any formed coalition would be required to possess *a multitude of desirable characteristics* (e.g., high collective storage capacity, and high collective discharge rate); and, if the aim is to balance electricity demand in real time, any such service should be offered by the appropriate coalition almost instantaneously.

[1] Technical University of Crete, School of Electronic & Computer Engineering, Greece, email: {fchristianos, gchalkiadakis}@isc.tuc.gr

In this paper, we overcome the aforementioned difficulties by employing, for the first time in the literature, *hypergraphs* to achieve the timely formation of coalitions that satisfy *multiple criteria*. In our approach, EV agents that share specific characteristics are organised into *hyperedges*. Then, building on the existing hypergraphs literature [5], we propose algorithms for hypergraph *transversal* to identify sets of vertices (agents) that combine several desirable characteristics; and hypegraph *clustering*, that potentially allows the identification of clusters of high quality agents. Moreover, we put forward a heuristic formation algorithm that generates high quality coalitions near-instantaneously, and scales linearly with the number of agents.

In some detail, a *transversal* (or *hitting set*) of a hypergraph H, is a set $T \subseteq V$ with hyperedges X where $X = E$ (i.e., vertices in T belong to *all* hyperedges in E). Our *transversal algorithm* finds hypergraph *minimal transversals*, and uses these to select nodes for the acting coalition. Our *clustering algorithm*, on the other hand, clusters together agents with similar characteristics, and then identifies promising clusters. Finally, the *heuristic algorithm* identifies agents that belong to high quality hyperedges, and creates a coalition using the best among them.

In contrast to existing approaches, we do not attempt to generate an optimal coalition structure, nor do we attempt to compute a single optimal coalition. Instead, we exploit the hypergraph representation of our problem in order to select agents and form highly effective coalitions, while being able to scale to *dozens of thousands* of agents within fractions of a second; and, in the case of our heuristic method, even to *millions* of EV agents in seconds.

Though here we apply it to the V2G problem, our approach is generic and can be used in *any* coalition formation setting. It is perhaps surprising that a powerful model like hypergraphs has not been so far exploited for devising efficient coalition formation methods, despite its intuitive connections to the concept of coalitions. Regardless, we are not aware of any work to date that has exploited hypergraphs and related algorithms in order to perform *real-time*, *large-scale*, *multi-criteria* coalition formation, as we do in this paper.

2 Our Approach

In order to develop multi-criteria coalition formation algorithms that generate coalitions efficiently, we employ the concept of *a hypergraph*. A hypergraph $H = (V, E)$ is a generalization of a graph, where each *hyperedge* $e \in E$ can contain any number of *vertices (or nodes)* in the set V. Vertices in H correspond to agents; a hyperedge corresponds to some particular *attribute* or *characteristic* possessed by the agents in the hyperedge. In the V2G setting, an EV agent is represented by a node in our hypergraph; while the hyperedges correspond to vehicle characteristics. More specifically, a hyperedge

corresponds to a "quality level" of some EV attribute.

In order to represent the different *quality* of the various hyperedges, and utilize it in our algorithms, we mark each hyperedge with a weight. Intuitively, *a high degree node is a high quality one*. This fact is exploited in our algorithms.

Organizing the information relating to specific agent attributes using hyperedges, enables us to both access this information efficiently, and keep it organized. Moreover, in many settings, agent characteristics captured by hyperedges, naturally correspond to criteria according to which we can form coalitions. Our approach of using hypergraphs is even more generic than what implied so far, since we can easily define hyperedges that contain agents which are or are not *permitted* to connect with each other, for various reasons; and since we can exploit the hypergraph to allow the formation of coalitions according to a multitude of criteria.

2.1 Criteria for Forming Coalitions

The algorithms presented in this work can be employed by any entity or enterprise (such as the Grid, utility companies or Smart Grid cooperatives) that wants to form EV coalitions for the V2G problem, using any set of criteria of its choosing. We identify three such natural criteria, namely *reliability, capacity* and *discharge rate*. These criteria are consistently mentioned in the related literature, though perhaps explicitly identified as such (see, e.g., [2]).

First of all, a coalition has to be consistently *reliable*. A reliable coalition will be able to serve the power that has been requested without any disruptions.

In addition, a coalition must fulfill a *capacity* requirement. The *capacity* of a coalition is the amount of electricity (measured in *kWh*) the coalition will be offering to the Grid. In fact, gathering enough EV capacity to cover the Grid needs, during high demand periods, is the main objective of any V2G solution.

Another factor in the V2G problem is the *discharge rate* of a coalition (or, of a single EV)—the rate by which the coalition (resp., the EV) is able to provide (electrical) energy to the Grid over a specified time period. Discharge rate is measured in *kW* (=*kWh / h*). Naturally, a coalition has a high discharge rate if its members' discharge rates are high—assuming that the discharge rate is additive.

Now, the hypergraph used in our current implementation was designed so that it could easily satisfy requests pertaining to these particular criteria. In our model, we assume that, at any time step that this is required—due to a consumption peak, an unplanned event, or the need to regulate frequency and voltage—the Grid (or some other entity) advertises its demand for a V2G coalition with several desirable characteristics. As noted in [2], individual EVs are well-suited for providing services at short notice. Our experiments confirm that we can select *coalitions* from a huge pool of EVs that provide large amounts of power at short notice, and with high reliability.

3 Experiments and Results

In this section we present the evaluation of our algorithms. We generated 20 hypergraphs with $20,000$ EVs each, and then ran each algorithm on every hypergraph 10 times, and took the averages.

Our evaluation examined *(a)* how fast and *(b)* by selecting how many EVs the set requirements can be met. Naturally, the faster an algorithm forms a coalition that meets all the requirements, the better. Moreover, coalitions with fewer vehicles are preferable, since this allows for a more efficient allocation of resources, and means that fewer EVs will share the coalitional payoff.

Algorithm	Heuristic	Clustering	Transversal
Mean Coalition Size (# EVs)	58.5	98	64
Mean Running Time (ms)	25	709	120
Mean Generat.+Run. Time (ms)	1041	1725	1136

Table 1. Summarizing the performance results

Table 1 shows the performance of our methods when meeting the set goals of 1,000 *kW* discharge rate and 10,000 *kWh* capacity. Then, Tables 2 and 3 show how our algorithms scale against an increasing capacity goal and an increasing EVs population. Against an increasing capacity goal, the heuristic algorithm scales exponentially. The transversal algorithm scales with steps, since sets of minimal transversals are generated and used (and if the requirements are not met, more such sets are generated). Finally the clustering algorithm is largely independent of the size of the capacity goal.

Goal (kWh)	Heuristic (sec)	Clustering (sec)	Transversal (sec)
10,000	0.03	0.049	0.20
70,000	0.06	0.049	0.34
130,000	0.19	0.049	0.33
190,000	0.39	0.049	0.34
250,000	0.64	0.049	0.32

Table 2. Scaling against an increasing "capacity" goal

EVs	Heuristic (sec)	Clustering (sec)	Transversal (sec)
10,000	0.012	0.14	0.03
12,000	0.015	0.22	0.05
14,000	0.017	0.31	0.07
16,000	0.019	0.36	0.08
18,000	0.021	0.50	0.09
20,000	0.024	0.69	0.12

Table 3. Scaling against an increasing EV population

Against an increasing agent population, the heuristic algorithm *scales linearly up to a million EVs within* 1.1 *second* on a Sandy Bridge i7-2600K at 4.2 GHz. The transversal and clustering algorithms scale exponentially, but are within an acceptable time limit (less than a second) for the tested population sizes.

REFERENCES

[1] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge, 'Computational aspects of cooperative game theory', *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **5**(6), 1–168, (2011).

[2] Sachin Kamboj, Willett Kempton, and Keith S Decker, 'Deploying power grid-integrated electric vehicles as a multi-agent system', in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 13–20. International Foundation for Autonomous Agents and Multiagent Systems, (2011).

[3] Talal Rahwan, Sarvapali D Ramchurn, Nicholas R Jennings, and Andrea Giovannucci, 'An anytime algorithm for optimal coalition structure generation', *Journal of Artificial Intelligence Research (JAIR)*, **34**(1), 521–567, (2009).

[4] Sarvapali D Ramchurn, Perukrishnen Vytelingum, Alex Rogers, and Nicholas R Jennings, 'Putting the'smarts' into the smart grid: a grand challenge for artificial intelligence', *Communications of the ACM*, **55**(4), 86–97, (2012).

[5] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf, 'Learning with hypergraphs: Clustering, classification, and embedding', in *Advances in Neural Information Processing Systems*, pp. 1601–1608, (2006).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1606

Increasing Coalition Stability in Large-Scale Coalition Formation with Self-Interested Agents

Pavel Janovsky[1] and **Scott A. DeLoach**[1]

Abstract. In coalition formation with self-interested agents both social welfare of the multi-agent system and stability of individual coalitions must be taken into account. However, in large-scale systems with thousands of agents, finding an optimal solution with respect to both metrics is infeasible.

In this paper we propose an approach for finding coalition structures with suboptimal social welfare and coalition stability in large-scale multi-agent systems. Our approach uses multi-agent simulation to model a dynamic coalition formation process. Agents are allowed to deviate from unstable coalitions, thus increasing the coalition stability. Furthermore we present an approach for estimating coalition stability, which alleviates exponential complexity of coalition stability computation. This approach is used for estimating stability of multiple coalition structures generated by the multi-agent simulation, which enables us to select a solution with high values of both social welfare and coalition stability. We experimentally show that our algorithms cause a major increase in coalition stability compared to a baseline social welfare-maximizing algorithm, while maintaining a very small decrease in social welfare.

1 Introduction

Coalition formation is a process of grouping of agents into *coalitions* in order to increase the agents' cooperation. A goal of coalition formation is often to increase social welfare of the multi-agent system, which can generate unrealistic solutions if the agents prefer their own profit to the global social welfare. These self-interested agents would deviate from the computed social welfare-maximizing coalitions.

In coalition formation with self-interested agents, *stability* of the coalitions, which measures the coalition's ability to de-incentivize any sub-coalition of agents from leaving the coalition, must be addressed as a concept that along with the social welfare influences the coalition formation algorithms and solutions. Coalition stability is addressed in game theoretical literature mainly through the concept of a *core*, which is a set of allocations to the agents in a coalition, such that these allocations cannot be improved upon by allocations to a subset of these agents. While the *core* is a strong concept, its computation in a setting where coalition values are generated by general polynomial-time functions requires an evaluation of all $2^{|C|}$ possible sub-coalitions of each coalition C containing $|C|$ agents. In this setting even determining whether the *core* is non-empty is $\Delta_2^P - complete$ [3]. This complexity makes the use of the *core* in large-scale systems infeasible. Therefore instead of the *core* we approach coalition stability using multi-agent simulation. Instead of looking for stable distribution of the coalition value to the agents, we specify an allocation scheme beforehand and let the agents utilize this information to choose more stable coalitions.

Specifically, the contributions of this paper are the following:

1. *An algorithm for large-scale coalition formation that increases coalition stability by allowing agents to deviate.*
2. *An approach for selecting sub-optimal solutions based on their social welfare and coalition stability.*

2 Problem Statement

We study the coalition formation problem, in which agents $a_1, a_2, \ldots, a_n \in A$ form coalitions C_i such that each agent belongs to exactly one coalition. We assume that the agents have full information about each others' states. A coalition structure CS is a set of all coalitions C_i that the agents formed. The task is to find a coalition structure that maximizes its social welfare as well as its stability.

We represent the social welfare by a gain metric defined in [4] as $g(CS) = \frac{1}{n} \cdot (v(CS) - v(CS_0))$, where $v(CS)$ is a value of a coalition structure CS, which is a sum of coalition values $v(C)$ assigned by a polynomial-time function, and CS_0 denotes the coalition structure of singleton coalitions. The gain shows an average benefit of an agent in coalition formation.

Self-interested agents maximize their own profit, which we define as marginal contribution of an agent to a coalition at the time of entry. [1] describes games that use this profit as Labor Union games.

Since finding coalition stability is computationally expensive, we approximate it by $stability_\alpha$. We say that a coalition C is $\alpha-$stable if no sub-coalition D with $\langle 1, \alpha \rangle$ members can be formed in which some agents would benefit more and no agent would benefit less than in C. We define $stability_\alpha$ of a coalition structure in terms of α as

$$stability_\alpha(CS) = \frac{|\alpha\text{-stable coalitions in CS}|}{|CS|}, \qquad (1)$$

where $|CS|$ denotes the number of coalitions in CS. With increasing α $stability_\alpha$ approaches the true stability of CS, which we define as the ratio of stable coalitions in CS. Since $stability_\alpha$ is non-increasing w.r.t. α, it is an upper estimate of this true stability of CS.

Finally, we use the price of stability $PoS(CS_{sw}, CS_{sa}) = g(CS_{sw})/g(CS_{sa})$ to show the ratio between the gain of social welfare maximizing solutions CS_{sw} and the gain of solutions reached by behavior of self-interested agents CS_{sa}.

3 Methodology

We find solutions to coalition formation using multi-agent simulation. We extend a multi-agent simulation framework for large-scale coalition formation proposed in [4], in which the agents maximize

[1] Department of Computer Science, Kansas State University, USA, email: {janovsky, sdeloach}@ksu.edu

the social welfare. In that framework the agents use strategies to decide about leaving their coalitions and joining new coalitions. The coalitions are evaluated by a polynomial-time valuation function. This process repeats in an iterative fashion, resulting in an agent-driven search of the coalition structures state-space. While [4] shows almost-optimal performance in small-scale scenarios and stable gain in large-scale scenarios, it does not consider stability of the solutions.

In order to increase stability of coalition structures we extend the algorithm from [4] by allowing the agents to create more stable sub-coalitions within their coalition by the process of deviation, and by selecting the best solution out of the pool of solutions generated by the simulation with respect to both social welfare and stability.

3.1 Deviation

Deviation guides the search towards more stable coalition structures by allowing agents to leave their current coalition along with other agents from the same coalition. A sub-coalition of agents can deviate from its coalition if no agent loses profit by deviation and at least one agent gains profit. Considering all $2^{|C|-1}$ possible sub-coalitions that an agent can be part of is infeasible, therefore the agents use a heuristic to guide their search. Some possible heuristics are adding agents to the sub-coalition in order of increasing and decreasing profit, and in random order. Our experiments showed that most stable coalitions were found using the increasing profit heuristic.

Deviation is performed in our model after the agents decide on leaving and joining coalitions. Each iteration of the simulation therefore consists of two steps: social welfare maximization by leaving and joining coalitions, and stability maximization by deviation.

3.2 Solution selection

An advantage of using multi-agent simulation for coalition formation is the fact that it creates a pool of solutions encountered during the search. At the end of the simulation, [4] selects from this pool a solution that maximizes the gain. We propose to select a solution based on both gain and stability metrics. However, computing stability of a coalition structure is computationally expensive, therefore we use $stability_\alpha$ to estimate the true stability of the solutions.

We compute $stability_\alpha$ in an iterative fashion for increasing $\alpha \in \langle 1, \alpha_{max} \rangle$. We only have to determine whether a coalition is α-stable if it is $(\alpha - 1)$-stable. We mark a coalition C α-stable if in all permutations of all combinations of α agents from C some agents lose or no agent gains profit[2]. We then calculate $stability_\alpha$ using Eq. 1.

After $stability_\alpha$ of all coalition structures is computed, a multi-criteria optimization is used to select a best coalition structure based on its gain and $stability_\alpha$. Common approaches of multi-criteria optimization are finding Pareto optimal solutions and designing a fitness function. We use a simple fitness function that assigns a same weight to both social welfare and coalition stability.

4 Experimental Analysis

We tested our algorithm in a collective energy purchasing scenario from [6], which models agents as households that buy electricity based on their requested daily energy profiles. Electricity can be bought at spot markets based on current demand, and at cheaper forward markets based on demand prediction. Agents form coalitions in order to make their aggregate energy profiles more predictable so they could exploit the reduced prices of the forward market. We used a dataset of daily energy profiles of households in Portugal [5]. To prevent the trivial grand coalition from being the optimal solution, we use a coalition size penalty $\kappa = min(-|C|+\mu, 0)^\gamma$ with $\mu = 10$ and $\gamma = 1.1$. Agents move to new coalitions using *local search strategy* [4], which is a best response strategy with a random element. Results are averaged over 10 random runs of our algorithm.

We compared results of our algorithms with the baseline algorithm [4] using the $stability_\alpha$ and *price of stability* metrics, as shown in Table 1, in which the first line represents the baseline algorithm. Table 1 shows that our algorithms increase significantly the coalition stability, while introducing a necessary, but very low, price of stability. Combination of deviation and solution selection yields the highest coalition stability.

Table 1: Trade-off between average stability and average price of stability achieved by our algorithms with $\alpha = 4$.

Algorithm		Results	
Deviation	**Solution selection**	**Average** $stability_\alpha$	**Average** PoS
No	No	0.5538	-
Yes	No	0.7857	1.0193
No	Yes	0.7121	1.0029
Yes	Yes	0.8363	1.0347

5 Conclusion

Algorithms that find stable coalition structures are often proposed for problems that restrict the valuation functions or scale, such as [1, 2]. Practical aspects of the high complexity of finding stable coalitions for large-scale multi-agent systems are often not considered.

In this work we proposed an approach for increasing coalition stability in large-scale coalition formation with self-interested agents and arbitrary valuation functions. We modeled agent behavior using multi-agent simulation, in which we allowed the agents to choose profitable coalitions and deviate from unstable coalitions. At the end of the simulation, we selected a solution out of a pool of generated coalition structures based on its social welfare and stability. We experimentally showed that our approach is able to increase the stability of the solutions in a real-world scenario. We also showed that the necessary price for this increase in stability that our algorithm incurs to the social welfare is very low.

REFERENCES

[1] John Augustine, Ning Chen, Edith Elkind, Angelo Fanelli, Nick Gravin, and Dmitry Shiryaev, 'Dynamics of profit-sharing games', *IJCAI'11: 21st International Joint Conference on Artificial Intelligence*, (2011).

[2] Filippo Bistaffa and Alessandro Farinelli, 'A fast approach to form core-stable coalitions based on a dynamic model', *2013 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2013*, (2013).

[3] Gianluigi Greco, Enrico Malizia, Luigi Palopoli, and Francesco Scarcello, 'On the complexity of the core over coalition structures', *Twenty-Second International Joint Conference on Artificial Intelligence*, (2011).

[4] Pavel Janovsky and Scott A. DeLoach, 'Multi-agent simulation framework for large-scale coalition formation', *2016 IEEE/WIC/ACM International Conference on Web Intelligence*, (2016).

[5] M. Lichman. UCI machine learning repository, 2013. Available at https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014.

[6] Meritxell Vinyals, Filippo Bistaffa, Alessandro Farinelli, and Alex Rogers, 'Coalitional energy purchasing in the smart grid', *2012 IEEE International Energy Conference and Exhibition, ENERGYCON 2012*, (2012).

[2] All permutations must be considered because the order in which agents join coalitions determines their profit

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1608

Strategies for Privacy Negotiation in Online Social Networks

Dilara Keküllüoğlu[1] and **Nadin Kökciyan**[2] and **Pınar Yolum**[3]

1 Introduction

Online social networks are changing the way information is shared among individuals. Contrary to traditional Web systems, such as e-commerce Web sites, where information about a user is managed solely by the user herself, in online social networks other users can contribute to the content that is shared about an individual. The shared content may reveal information about the user, which the user might not wanted to share herself. This creates a privacy breach on the user's side. In current online social networks, a common way to deal with this is for the user to complain to the social network administration and ask the content to be removed. However, by the time the content is removed (if at all), many people might have seen it already. Ideally, it would be best if such a content was not shared in the first place.

Recent work on privacy management has focused on applying agreement technologies to solve privacy problems before they take place. Two important works in this line are that of Mester *et al.* [3] and Such and Rovatsos [4]. These approaches apply negotiation techniques [2] to resolve privacy conflicts among users. They both consider negotiation before a content is being shared. Both approaches assume that negotiation is being performed on a single content and cannot account for ongoing interactions. However, it has been observed that users build reciprocal trust in online social networks and respect others as much as others respect them. Hence, it is of utmost importance to consider repeated interactions, as opposed to single interactions, to study privacy leakages.

This paper proposes a multiagent management of privacy in online social networks, where each user is represented by an agent that helps its user preserve its privacy. The privacy of users is preserved by a hybrid negotiation architecture where privacy domain and rules are represented semantically but the decision making is done by the agents using utility functions. The paper develops various negotiation strategies including one that exploits reciprocity. The key idea is that each agent keeps track of whether a certain other user has been helpful before in preserving privacy using a credit system. When agents help others in preserving their privacy, their credit increases so that later they can ask others to help them. Hence, helping others to preserve privacy serves as an incentive. Using these strategies, agents can negotiate on the content and agree on how it will be shared before the post goes online.

[1] Bogazici University, Turkey, email: dilara.kekulluoglu@boun.edu.tr
[2] Bogazici University, Turkey, email: nadin.kokciyan@boun.edu.tr
[3] Bogazici University, Turkey, email: pinar.yolum@boun.edu.tr

2 Negotiation Architecture

Our proposed negotiation architecture is based on semantic representation of negotiation concepts and privacy rules, but enables each agent to use its own utility functions to evaluate negotiation offers. We use PriNego [3] as the basis for the semantic aspects of negotiation. PriNego proposes a negotiation framework for privacy where each agent represents a user in the social network. Each agent is aware of the privacy concerns of its user but also has information about the social network, such as the friends of the user. This information is captured in an ontology that is represented in Web Ontology Language (OWL).

Privacy Concerns: Each agent captures its user's privacy concerns as semantic rules (privacy rules) represented with a Semantic Web Rule Language (SWRL) [1]. A privacy rule describes a situation wherein an agent would reject a particular negotiation offer. Consider a user Alice who does not want her colleagues to see her pictures within a leisure context. If Alice herself was sharing a picture in a leisure context, she or her agent can enforce that the audience of a post is set so that colleagues are not included. However, if Bob, a friend of Alice is about to share such a content, then it is difficult to enforce Alice's privacy constraint on Bob's content.

Negotiation: In such a setting, Alice and Bob's agents can negotiate among each other to decide if the content should be shared and if so, under which constraints. Following the above example, if Bob asks Alice to share a post in leisure context, then Alice's agent would reject this regarding Alice's privacy rule. Our proposed system enables agents to provide a rejection reason as well (e.g., rejected because of context). If a negotiation offer does not violate any of the privacy rules of the user, then the agent accepts this offer. For example, if Bob would ask Alice to share a post in work context, then Alice's agent would accept this offer. A user might have various privacy constraints but these might not be equally important. To capture the fact that a rule is more important than a second rule, we associate a weight with each rule.

Decision Making: When an agent creates an offer, the evaluations done to decide whether to accept an offer as well as to create a new counter-offer constitute the *negotiation strategy* of an agent. Here, we require each agent to have a utility function, which is based on the privacy concerns of the user. The utility value of a post request (negotiation offer) considers the threshold value set by the user agent as well. An agent makes a decision about a post request regarding its utility function.

The agent that initiates the negotiation (i.e., initiator) will

have a different utility function than an agent that negotiates for her privacy (i.e., negotiator) since both agents have different responsibilities during negotiation.

- A *negotiator agent* is responsible for evaluating a post request and making a decision about this post request based on its utility. In case where it wants to reject it, it may also provide rejection reason(s) depending on the strategy that it follows.
- An *initiator agent* is responsible for initializing the negotiation with other agents (i.e., agents that are relevant to a post request). Then, it collects responses from other agents. If all agents agree on sharing the post request, then it shares the post. Otherwise, it will try to update the post request according to its utility and rejection reasons of others. As a result of this, it can choose to share the post, continue or terminate the ongoing negotiation.

3 Strategies for Privacy Negotiation

We have developed three negotiation strategies that agents can use.

Good-Enough-Privacy (GEP): In this strategy, the initiator agent sends a post request to relevant agents. At each iteration, each agent provides a rejection reason if it rejects the post request. For this, each agent evaluates a post request by computing a utility. If this utility is above the agent's utility threshold, then the agent accepts the post request as it is. Otherwise, the agent finds its most important rule then it rejects the post request and provides the corresponding rejection reason.

Maximal-Privacy (MP): GEP strategy sends only one rule per iteration. However, the initiator agent may be willing to revise the post request by considering multiple rejection reasons. Hence, the negotiation could terminate in fewer iterations. For example, the initiator agent might want the negotiation to be over in two rounds, and an agent relevant to the post request might have three rules that are violated. The initiator agent may be actually ready to prevent all these violations. If the negotiator agent uses GEP strategy, then at most two rejection reasons can be considered. In MP, an agent will send all rejections reasons to the initiator agent. If the initiator agent rejects the post request, then the negotiator agent will start narrowing the set of rejection reasons by removing rejection reasons that are less important than others.

For GEP and MP, the outcome of the negotiation is only determined by considering the current situation and ignoring the previous interactions.

Reciprocal Strategy (RP): The outcome of a negotiation is beneficial for all the negotiating agents; however one party is usually better than the others. This difference might be insignificant for many negotiations. The difference may get disadvantageous for the others if one party is favored most of the times. To prevent this, we propose a new strategy based on reciprocity called *Reciprocal Strategy* (RP). In this strategy, agents negotiate regarding the previous behaviors and negotiations. If one party is favored more in previous negotiations, then this strategy tries to favor the other party. To keep track of the previous negotiations, we use a point-based system where both parties have the same amount of points in the initial state (e.g., each 5pts). For every negotiation, agents exchange points depending on who is the initiator and how much benefit they get from that negotiation.

At every negotiation iteration, the initiator agent sends the post request together with a point offer to the negotiator agent. In the previous strategies, the negotiator agent was calculating a utility per post request, and if this utility was below its utility threshold, it would send a rejection reason. In this strategy, agents also consider point offers of each other while computing their utilities. Hence, they try to compensate the utility shortage by the points that they get from others. If the computed utility is below the threshold, the negotiator agent asks the initiator agent for sufficient points to accept the post request. Otherwise, the negotiator agent accepts the post request as it is.

Since the negotiations in this strategy change depending on the previous interactions, the effects of RP should be captured observing continuous posting. We have tested our system with consecutive postings. In order to understand how posting habits of the people affect the outcome of the negotiation, we have tried different cases. To see how these posting habits affect the resulting utilities, we have considered various cases for two users: (i) one user shares a post, (ii) both users share posts regularly.

4 Future Directions

An important first step is to evaluate these strategies in comparison to each other. This comparison should take into account various factors such as overall points, time constraints, number of posts to be shared and so on.

As a second step, it is worthwhile to incorporate trust relations into the utility functions such that agents are more willing to cooperate with those that they trust. This would reflect real life relations more closely. Another important point is to enable the negotiation framework to be updated such that privacy rule weights can be learned over time.

ACKNOWLEDGEMENTS

This work has been supported by TUBITAK.

REFERENCES

[1] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, et al., 'SWRL: A semantic web rule language combining OWL and RuleML', *World Wide Web Consortium Member submission*, **21**, 79, (2004).

[2] N. R. Jennings, A. R. Lomuscio P. Faratin, S. Parsons, C. Sierra, and M. Wooldridge, 'Automated negotiation: Prospects, methods and challenges', *International Journal of Group Decision and Negotiation*, **10**(2), 199–215, (2001).

[3] Yavuz Mester, Nadin Kökciyan, and Pınar Yolum, 'Negotiating privacy constraints in online social networks', in *Advances in Social Computing and Multiagent Systems*, eds., Fernando Koch, Christian Guttmann, and Didac Busquets, volume 541 of *Communications in Computer and Information Science*, 112–129, Springer International Publishing, (2015).

[4] Jose M. Such and Michael Rovatsos, 'Privacy policy negotiation in social media', *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, **11**(1), 4:1–4:29, (2016).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1610

DA-BSP: Towards Data Association Aware Belief Space Planning for Robust Active Perception

Shashank Pathak and **Antony Thomas** and **Asaf Feniger** and **Vadim Indelman**[1]

Belief space planning (BSP) and decision-making under uncertainty are fundamental problems in robotics and artificial intelligence, with applications including autonomous navigation, object grasping and manipulation, active SLAM, and robotic surgery. In the presence of uncertainty, such as in robot motion and sensing, the true state of variables of interest (e.g. robot poses), is unknown and can only be represented by a probability distribution over possible states, given available data. This distribution, the belief space, is inferred using probabilistic approaches based on incoming sensor observations and prior knowledge. The corresponding BSP problem is an instantiation of a partially observable Markov decision problem (POMDP) [4].

Existing BSP approaches (e.g. [2,5,9,11]) typically assume data association to be given and perfect, i.e. the robot is assumed to correctly perceive the environment to be observed by its sensors, given a candidate action. However, this assumption, denoted for brevity as DAS, can be harder to justify while operating in ambiguous and perceptually aliased environments (see Figure 1), and in the presence of different sources of uncertainty (uncertainty due to stochastic control and imperfect sensing).

Indeed, in the presence of ambiguity, DAS may lead to incorrect posterior beliefs and as a result, to sub-optimal actions. More advanced approaches are thus required to enable reliable operation in ambiguous conditions, approaches often referred to as (active) robust perception. Yet, existing robust perception approaches (e.g. [1,3,6,10]) focus on the passive case, where robot actions are externally determined and given.

In this work we develop a general data association aware belief space planning (DA-BSP) framework capable of better handling complexities arising in a real world, possibly perceptually aliased, scenarios. We rigorously incorporate reasoning about data association within belief space planning (and inference), while also considering other sources of uncertainty (motion, sensing and environment). In particular, we show that due to perceptual aliasing, the posterior belief becomes a mixture of probability distribution functions, and design cost functions that measure the expected level of ambiguity and posterior uncertainty. Using these and standard costs (e.g. control penalty, distance to goal) within the objective function, yields a general framework that reliably

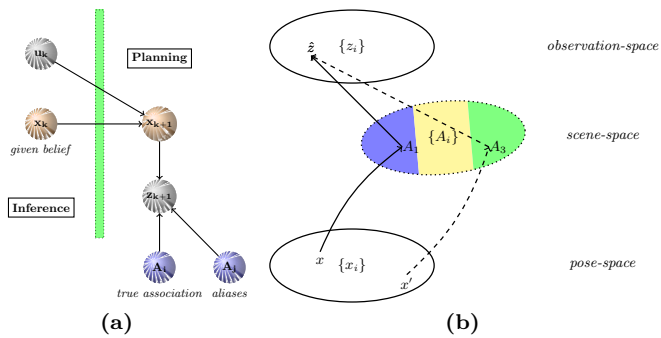


Figure 1: (a) Generative graphical model. Standard BSP approaches assume data association (DA) is given and perfect (DAS). We incorporate data association aspects within BSP and thus can reason about ambiguity (e.g. perceptual aliasing) at a decision-making level. (b) Schematic representation of pose, scene and observation spaces. Scenes A_1 and A_3 when viewed from perspective x and x' respectively, produce the same nominal observation $\hat{z}$, giving rise to *perceptual aliasing*.

represents action impact, and in particular, capable of active disambiguation. Our approach is thus applicable to robust active perception and autonomous navigation in perceptually aliased environments. In this short paper, we provide a concise overview of the DA-BSP approach, referring the interested reader to [7,8] for full details.

Concept and Approach Overview

Given some candidate action u_k and the belief at planning time k, we can reason about a future observation z_{k+1} (e.g. an image) to be obtained once this action is executed; its actual value is unknown. All the possible values such an observation can assume should be thus taken into account while evaluating the objective function, which can be written as:

$$J(u_k) \doteq \int_{z_{k+1}} \overbrace{\mathbb{P}(z_{k+1} \mid \mathcal{H}_{k+1}^-)}^{(a)} \; c \left(\overbrace{\mathbb{P}(X_{k+1} | \mathcal{H}_{k+1}^-, z_{k+1})}^{(b)} \right), \quad (1)$$

where X_{k+1} denotes the past and current robot poses $X_k \doteq \{x_0, \ldots, x_k\}$, and $\mathcal{H}_{k+1}^- \doteq \{u_{0:k}, Z_{0:k}\}$.

The two terms (a) and (b) in Eq. (1) have intuitive meaning: for each considered value of z_{k+1}, (a) represents how likely is it to get such an observation when both the history $\mathcal{H}$ and control u_k are known, while (b) corresponds to the posterior belief *given* this specific z_{k+1}.

Existing BSP approaches typically consider data association is solved (DAS), i.e. given and perfect. In other words, DAS

[1]Department of Aerospace Engineering, Technion - Israel Institute of Technology, Israel, emails: {shashank, antony.thomas, asaff, vadim.indelman}@technion.ac.il. This work was partially supported by the Israel Science Foundation (ISF) and the Technion Autonomous Systems Program (TASP).

means we can correctly associate each possible measurement z_{k+1} with the corresponding scene A_i it captures and write the corresponding measurement likelihood term $\mathbb{P}(z_{k+1}|x_{k+1}, A_i)$. Yet, it is unknown from what future robot pose x_{k+1} the actual observation z_{k+1} will be acquired, since the *actual* robot pose x_k at time k is unknown, the control is stochastic and sensing is imperfect. In inference, we have a similar situation with the key difference that the observation z has been acquired.

Rather than assuming DAS, in this work we incorporate within BSP (and similarly within inference) reasoning about possible scenes or objects that the future observation z_{k+1} could be generated from, see Figure 1. While this may seem computationally expensive, realistic scenarios typically exhibit *parsimonious data association*: If the environment has only distinct scenes or objects, then for each specific value of z_{k+1}, there will be only one scene A_i that can generate such an observation. In the case of perceptually aliased environments, there could be several other scenes (or objects) that are either completely identical or have a similar visual appearance when observed from appropriate viewpoints. They could equally well explain the considered observation z_{k+1}. Thus, there are several possible associations $\{A_i\}$ and due to localization uncertainty determining which association is the correct one is not trivial. As we show in [7], in these cases the posterior (term (b) in Eq. (1)) becomes a Gaussian mixture with appropriate weights that we rigorously compute. Additionally, the weight updates are capable of discriminating against unlikely data-associations, during the planning steps.

We now briefly summarize how terms (a) and (b) in Eq. (1) are calculated while reasoning about data association, referring the reader to [7] for full details.

Computing the term (a): $\mathbb{P}(z_{k+1}|\mathcal{H}^-_{k+1})$: Applying total probability over non-overlapping scene space $\{A_\mathbb{N}\}$ and marginalizing over all possible robot poses, yields

$$\mathbb{P}(z_{k+1}|\mathcal{H}^-_{k+1}) \equiv \sum_i^{|A_\mathbb{N}|} \int_x \mathbb{P}(z_{k+1}, x, A_i \,|\, \mathcal{H}^-_{k+1}) \doteq \sum_i^{|A_\mathbb{N}|} w^i_{k+1}. \quad (2)$$

As seen from the above equation, to calculate the likelihood of obtaining some observation z_{k+1}, we consider separately, for each scene $A_i \in \{A_\mathbb{N}\}$, the likelihood that this observation was generated by scene A_i. This probability is captured for each scene A_i by a corresponding weight w^i_{k+1}; these weights are then summed to get the actual likelihood of observation z_{k+1}. As shown in [7], these weights naturally account for perceptual aliasing aspects for each considered z_{k+1}.

In practice, instead of considering the entire scene space $\{A_\mathbb{N}\}$ that could be computationally costly, the availability of the belief from the previous time step, $b[X^-_{k+1}]$, enables us to consider only those scenes that could be actually observed from the viewpoints with non-negligible probability according to $b[X^-_{k+1}]$.

Computing the term (b) : $\mathbb{P}(X_{k+1}|\mathcal{H}^-_{k+1}, z_{k+1})$: The term ($b$), $\mathbb{P}(X_{k+1}|\mathcal{H}^-_{k+1}, z_{k+1})$, represents the posterior probability conditioned on observation z_{k+1}. This term can be similarly calculated, with a key difference: since the observation z_{k+1} is given, it must have been generated by *one* specific (but unknown) scene A_i according to an appropriate measurement model. Hence, also here, we consider all possible such scenes and weight them accordingly, with weights $\tilde{w}^i_{k+1}$ representing the probability of each scene A_i to have generated the observation z_{k+1}.

As shown in [7], the term (b) in Eq. (1) is a GMM with M_{k+1} components, $\mathbb{P}(X_{k+1}|\mathcal{H}^-_{k+1}, z_{k+1}) = \sum_{r=1}^{M_{k+1}} \xi^r_{k+1} b[X^{r+}_{k+1}]$, where $b[X^{r+}_{k+1}]$ represents the rth component of the belief, and the weights ξ^r_{k+1} are defined recursively (see full details in [7]). Interestingly, the number of components can not only go down $M_{k+1} \leq M_k$ (as a result of a partially or fully disambiguating action), but could also go up, i.e. $M_{k+1} > M_k$.

To summarize the discussion thus far, we have shown that for the myopic case, the objective function (1) can be rewritten as

$$J(u_k) = \int_{z_{k+1}} \left(\sum_i^{|A_\mathbb{N}|} w^i_{k+1}\right) \cdot c \left(\sum_r^{M_{k+1}} \xi^r_{k+1} b[X^{r+}_{k+1}]\right). \quad (3)$$

In [7], we present the other ingredients of our approach, including sampling-based simulation of future observations $\{z_{k+1}\}$ given $b[X^-_{k+1}]$, and the design of suitable cost functions to quantify ambiguity level. We also show that DA-BSP considers data-association parsimoniously and a simple thresholding is enough for a scalable application of data-association aware belief space planning, and demonstrate key aspects basic and realistic simulations. Potential directions for future research include extension to non-myopic planning as well as proving the general theoretical properties of DA-BSP.

References

[1] L. Carlone, A. Censi, and F. Dellaert. Selecting good measurements via l1 relaxation: A convex approach for robust estimation over graphs. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2667–2674, 2014.

[2] V. Indelman, L. Carlone, and F. Dellaert. Planning in the continuous domain: a generalized belief space approach for autonomous navigation in unknown environments. *Intl. J. of Robotics Research*, 34(7):849–882, 2015.

[3] V. Indelman, E. Nelson, J. Dong, N. Michael, and F. Dellaert. Incremental distributed inference from arbitrary poses and unknown data association: Using collaborating robots to establish a common reference. *IEEE Control Systems Magazine (CSM), Special Issue on Distributed Control and Estimation for Robotic Vehicle Networks*, 36(2):41–74, 2016.

[4] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[5] A. Kim and R. M. Eustice. Active visual slam for robotic area coverage: Theory and experiment. *Intl. J. of Robotics Research*, 2014.

[6] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *Intl. J. of Robotics Research*, 32(7):826–840, 2013.

[7] S. Pathak, A. Thomas, A. Feniger, and V. Indelman. Robust active perception via data-association aware belief space planning. *arXiv preprint: 1606.05124*, 2016.

[8] S. Pathak, A. Thomas, A. Feniger, and V. Indelman. Towards data association aware belief space planning for robust active perception. In *AI for Long-term Autonomy, workshop in conjunction with IEEE International Conference on Robotics and Automation (ICRA)*, May 2016.

[9] S. Prentice and N. Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *Intl. J. of Robotics Research*, 2009.

[10] N. Sunderhauf and P. Protzel. Towards a robust back-end for pose graph slam. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1254–1261. IEEE, 2012.

[11] J. Van Den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *Intl. J. of Robotics Research*, 31(11):1263–1278, 2012.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1612

Bagged Boosted Trees for Classification of Ecological Momentary Assessment Data

Gerasimos Spanakis and **Gerhard Weiss** [1] and **Anne Roefs**[2]

Abstract. Ecological Momentary Assessment (EMA) data is organized in multiple levels (per-subject, per-day, etc.) and this particular structure should be taken into account in machine learning algorithms used in EMA like decision trees and its variants. We propose a new algorithm called BBT (standing for Bagged Boosted Trees) that is enhanced by a over/under sampling method and can provide better estimates for the conditional class probability function. Experimental results on a real-world dataset show that BBT can benefit EMA data classification and performance.

1 Background & Motivation

This work focuses on classification trees and how their ensembles can be utilized in order to set up a prediction environment using Ecological Momentary Assessment (EMA) data from a real-world study. EMA [8] refers to a collection of methods used in many different disciplines by which a research subject repeatedly reports on specific variables measured close in time to experience and in the subject's natural environment (e.g. experiencing food craving is measured again and again on the same subject). EMA aims to minimize recall bias, maximize ecological validity and allow microscopic analysis of influence behavior in real-world contexts. EMA data has a different structure than normal data and account for several dependencies between them, since e.g. many samples belong to the same subject so they are expected to be correlated. However, most decision trees that deal with EMA data do not take these specificities into account.

Bagging involves having each tree in the ensemble vote with equal weight while boosting involves incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models mis-classified. Major differences between bagging and boosting are that (a) boosting changes the distribution of training data based on the performance of classifiers created up to that point (bagging acts stochastically) and (b) bagging uses equal weight voting while boosting uses a function of the performance of a classifier as a weight for voting.

There are limited studies on combining bagging and boosting ([10], [3], [6] and [11]), however, none of these approaches have been applied to longitudinal data or take into account the EMA structure. Efforts to apply decision trees to EMA data have been attempted but they are mostly focusing on regression tasks ([7], [4], [2]) and on the other hand they do not use bagging or boosting for improving performance. Work in current paper aims at bridging this gap by combining bagging and boosting with the longitudinal data structure.

[1] Department of Data Science and Knowledge Engineering, Maastricht University, email:{jerry.spanakis,gerhard.weiss}@maastrichtuniversity.nl
[2] Faculty of Psychology and Neuroscience, Maastricht University, email: a.roefs@maastrichtuniversity.nl

2 BBT: The proposed algorithm

Let the training data be $x_1, ..., x_n$ and $y_1, ..., y_n$ where each x_i is a d-dimensional vector and $y_i \in \{-1, 1\}$ is the associated observed class label. To justify generalization, it is usually assumed that training data as well as any test data are *iid* samples from some population of (x, y) pairs. Our goal is to as accurately predict y_i given x_i.

The first step to fit a BBT is to select the loss function, which in the case of a classification problem is based on the logistic regression loss. After some initial parameter selection (number of trees to be grown in sequence, shrinkage (or learning) rate, size of individual trees and fraction of the training data sampled) we grow BBT (say using M trees) on the training data using the following process and by growing single Boosted Trees (BT):

- Divide the data into B (typically $5 - 10$) subsets and construct B training data sets each of which omits one of the B subsets (the 'out-of-bag' data). Each one of the B subsets is created by bootstrap sampling data points from the set of subjects ($p = 1, ..., P$). To create the learning set we introduce the strategy $\mathcal{S}$ according to which one observation is drawn per subject. This strategy is based on a simple rationale: When only one observation per subject is selected, the probability that different observations are used for the training of different trees is increased, although the same subjects might be selected which further reduces similarity between trees. By this way, we manage to incorporate advantages of subject-based bootstrapping and observation-based bootstrapping into the final BBT ensemble. Also, this approach can be applied to unbalanced data points per subject.
- Grow B BT; one for each of the B training sets, based on the AdaBoost algorithm [1]: First let $F_0(x_i) = 0$ for all x_i and initialize weights $w_i = 1/d$ for $i = 1, ..., d$. Then repeat the following for $m = 1, ..., M$ for each one of the B BT:
  - ⋆ Fit the decision tree g_m to the training data sample using weights w_i where g_m maps each x_i to -1 or 1.
  - ⋆ Compute:
    - the weighted error rate $\epsilon_m = \sum_{i=1}^{n} w_i I\{y_i \neq g_m(x_i)\}$
    - half its log-odds and derive $\alpha_m = \frac{1}{2} \log \frac{1-\epsilon_m}{\epsilon_m}$
  - ⋆ Let $F_m = F_{m-1} + \alpha_m g_m$
  - ⋆ Replace the weights w_i with $w_i = w_i e^{-\alpha_m g_m(x_i) y_i}$ and then renormalize by replacing each wi by $wi/(\sum w_i)$.
- Calculate the PE for each BT for tree sizes 1 to M from the corresponding out-of-bag data and pool across the B boosted trees. Predictions for new data are computed by first predicting each of

the component trees and then aggregate the predictions (e.g., by averaging), like in bagging.

- The minimum PE estimates the optimum number of trees $m*$ for the BT. The estimated PE of the single BT obtained by cross-validation can thus also be used to estimate PE for the BBT. BBT thus require minimal additional computation beyond estimation of $m*$.
- Reduce the number of trees for each BT to $m*$.

For a classification problem, we use an estimate $p_m(x)$ of the Conditional Class Probability Function (CCPF) $p(x)$ that can be obtained from F_m through a logistic link function:

$$p_m(x) = p_m(y = 1|x) = \frac{1}{1 + \exp(-2F_m(x))} \quad (1)$$

Classifying at the 1/2 quantile of the CCPF works well for binary classification problems but in the case of EMA data, sometimes classification with unequal costs or, equivalently, classification at quantiles other than 1/2 is needed. Strategies about correctly computing the CCPF are considered [5] by over/under-sampling which convert a median classifier into a q-classifier.

3 Experiments

In order to illustrate the effect of BBT, we now apply this method to an EMA dataset obtained by a study designed by the authors [9]. The EMA study followed 100 participants over the course of 14 days using experience & event sampling questionnaires ending up with over 5000 data points containing information about participants' eating events, emotions, circumstances, locations, etc. (in total there are 9 variables) for several time moments during each day that they participated in the study. Each data point is used to predict whether the next data point (provided that they both occur on the same day) will be a healthy or an unhealthy eating moment. Figure 1 shows an example of how data points (belonging to user "pp5") are converted and combined in order to enable early prediction using a classification algorithm (class can be either "healthy" or "unhealthy"). Then the BBT algorithm can be applied.

user	Date/time	crv	negE	posE	sp_cr	time	week	circ	loc	sp_eat
pp5	26/01/15 23:57	LOW	NO	LOW	N	evening	NO	LowLevel	Home	N
pp5	27/01/15 09:32	LOW	NO	HIGH	N	morning	NO	LowLevel	Home	N
pp5	27/01/15 12:17	MID	NO	HIGH	H	noon-after	NO	ComputerRelated	Work	U
pp5	27/01/15 14:43	LOW	YES	MID	N	noon-after	NO	Work	Work	H

user	crv	negE	posE	sp_cr	time	week	circ	loc	sp_eat	NextEating (y class)
pp5	LOW	NO	HIGH	N	morning	NO	LowLevel	Home	N	U
pp5	MID	NO	HIGH	H	noon-after	NO	ComputerRelated	Work	U	H

Figure 1: Data conversion example for early prediction

In the comparison between methods, BBT gave a PE of 23.3%, whereas the single classification tree (37.3%), bagged trees (28.9%), boosted trees (adaboost) (25.9%), random forests (26.8%) and B&B combine method [11] (26.2%) have higher PE than the BBT. Table 1 summarizes these results and also presents a series of experiments made to demonstrate the effectiveness of BBT when the number of different subjects (P) involved in the dataset increases. For relatively small numbers of subjects (10 or 20) performance of BBT and AdaBoost is comparable (although variance increases and the number of data samples is not large enough) but as P increases the performance of BBT is clearly better. Larger P means that there are more subjects in the dataset, thus the complexity of longitudinal structure increases and it is imperative to take this into account when classifying longitudinal data. This is the reason that BBT performs better than all other algorithms as P increases. However, for small P the effect of different subjects is smaller and this is the reason that Adaboost performs slightly better than all other algorithms.

Table 1: Prediction Error (Variance) % for different algorithms and different numbers of subjects (P)

	P=10	P=20	P=50	P=100
SCT	25.1 (0.10)	27.4 (0.08)	30.9 (0.10)	37.3 (0.06)
Bagging	24.4 (0.22)	23.8 (0.08)	30.1 (0.08)	28.9 (0.06)
Boosting	**22.0 (0.12)**	22.7 (0.10)	27.0 (0.06)	25.9 (0.04)
Random Forest	23.7 (0.16)	24.5 (0.14)	27.2 (0.04)	26.8 (0.04)
B&B Combine	23.2 (0.08)	25.1 (0.06)	26.8 (0.08)	26.2 (0.02)
BBT	22.4 (0.14)	**21.9 (0.06)**	**24.2 (0.04)**	**23.3 (0.02)**

4 Discussion

In this paper a combination of bagging and boosting was presented: Bagged Boosted Trees (BBT). BBT have the advantage of being able to deal with multiple categorical data which raises a scalability issue when dealing with classic models (like generalized linear models) that are widely used in EMA studies. Moreover, BBT can tackle potential nonlinearities and interactions in the data, since these issues are handled through the combination of many different trees of different sizes. Experimental results of BBT on a real-world EMA dataset clearly show improvement with respect to accuracy in prediction compared to other decision tree algorithms. Further work involves the evaluation of the conditional class probability function (based on over/under sampling of data), as well as the application to other EMA datasets. Finally, adjustment of boosting in order to implement weights based on subjects (and not individual observations) is a direction with promising results.

REFERENCES

[1] Yoav Freund, Robert E Schapire, et al., 'Experiments with a new boosting algorithm', in *ICML*, volume 96, pp. 148–156, (1996).

[2] Wei Fu and Jeffrey S Simonoff, 'Unbiased regression trees for longitudinal and clustered data', *Computational Statistics & Data Analysis*, **88**, 53–74, (2015).

[3] S Kotsiantis and P Pintelas, 'Combining bagging and boosting', *International Journal of Computational Intelligence*, **1**(4), 324–333, (2004).

[4] Wei-Yin Loh, Wei Zheng, et al., 'Regression trees for longitudinal and multiresponse data', *The Annals of Applied Statistics*, **7**(1), 495–522, (2013).

[5] David Mease, Abraham J. Wyner, and Andreas Buja, 'Boosted classification trees and class probability/quantile estimation', *J. Mach. Learn. Res.*, **8**, 409–439, (May 2007).

[6] J. Ross Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[7] Rebecca J Sela and Jeffrey S Simonoff, 'RE-EM trees: a data mining approach for longitudinal and clustered data', *Machine learning*, **86**(2), 169–207, (2012).

[8] Saul Shiffman, Arthur A Stone, and Michael R Hufford, 'Ecological momentary assessment', *Annu. Rev. Clin. Psychol.*, **4**, 1–32, (2008).

[9] Gerasimos Spanakis, Gerhard Weiss, Bastiaan Boh, and Anne Roefs, 'Network analysis of ecological momentary assessment data for monitoring and understanding eating behavior', in *Smart Health - International Conference, ICSH 2015, Phoenix, AZ, USA, November 17-18, 2015*, pp. 43–54, (2015).

[10] Yuk Lai Suen, Prem Melville, and Raymond J Mooney, 'Combining bias and variance reduction techniques for regression trees', in *Machine Learning: ECML 2005*, 741–749, Springer, (2005).

[11] Jianjun Xie, Viktoria Rojkova, Siddharth Pal, and Stephen Coggeshall, 'A combination of boosting and bagging for kdd cup 2009-fast scoring on a large database.', in *KDD Cup*, pp. 35–43, (2009).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1614

Reputation in the Academic World

Nardine Osman and **Carles Sierra** [1]

Abstract. This paper proposes a computational model based on peer reviews for assessing the reputation of researchers and research work. We argue that by relying on peer opinions, we address some of the pitfalls of current approaches for calculating the reputation of authors and papers. We also introduce a much needed feature for review management: calculating the reputation of reviews and reviewers.

1 MOTIVATION

With open access gaining momentum, **open reviews** becomes a more persistent issue. Current institutional and multidisciplinary open access repositories lack the quantitative assessment of the hosted research items that will facilitate the process of selecting the most relevant and distinguished content. Common currently available metrics, such as number of visits and downloads, do not reflect the quality of a research product, which can only be assessed directly by peers offering their expert opinion together with quantitative ratings based on specific criteria. The articles published in the Frontiers book [1] highlight the need for open reviews.

To address this issue we develop an open peer review module, the Academic Reputation Model (ARM), as an overlay service to existing institutional or other repositories. The model calculates the reputation of authors, reviewers, papers, and reviews, by relying on peer opinions. We argue that this addresses some of the pitfalls of current approaches for calculating the reputation of authors and papers. It also introduces a much needed feature for review management, and that is calculating the reputation of reviews and reviewers.

2 ARM: ACADEMIC REPUTATION MODEL

2.1 Data and Notation

To compute its reputation measures, ARM requires a *reputation data set* (which should be extracted from existing repositories) specified as the tuple $\langle P, R, E, D, a, o, v\rangle$, where

- $P = \{p_i\}_{i\in\mathcal{P}}$ is a set of papers (e.g. DOIs).
- $R = \{r_j\}_{j\in\mathcal{R}}$ is a set of researcher names or identifiers (e.g. the ORCHID identifier).
- $E = \{e_i\}_{i\in\mathcal{E}} \cup \{\bot\}$ is a totally ordered evaluation space, where $e_i \in \mathbb{N} \setminus \{0\}$ and $e_i < e_j$ iff $i < j$ and $\bot$ stands for the absence of evaluation. We suggest the range [0,100], although any other range may be used, and the choice of range will not affect the performance.
- $D = \{d_k\}_{k\in\mathcal{K}}$ is a set of evaluation dimensions, such as *originality*, *technical soundness*, etc.
- $a : P \to 2^R$ is a function that gives the authors of a paper.
- $o : R \times P \times D \times Time \to E$, where $o(r, p, d, t) \in E$ is a function that gives the opinion of a reviewer, as a value in E, on a dimension d of a paper p at a given instant of time t.
- $v : R \times R \times P \times Time \to E$, where $v(r, r', p, t) = e$ is a function that gives the judgement of researcher r over the opinion of researcher r', on paper p as a value $e \in E$. Therefore, a judgement is a reviewer's opinion about another reviewer's opinion.

[1] Artificial Intelligence Research Institute (IIIA-CSIC), Barcelona, Spain, email: {nardine, sierra}@iiia.csic.es

2.2 Reputation of a Paper

The reputation of a paper is a weighted aggregation of its reviews, where the weight is the reputation of the reviewer (Section 2.4).

$$R_P(p) = \begin{cases} \dfrac{\sum_{\forall r\in rev(p)} R_R(r)\cdot o(r,p)}{\sum_{\forall r\in rev(p)} R_R(r)} & \text{if } |rev(p)| \geq k \\ \bot & \text{otherwise} \end{cases} \quad (1)$$

where $rev(p) = \{r \in R \mid o(r, p) \neq \bot\}$ denotes the reviewers of a given paper, and k is a parameter specifying the minimum number of reviews required for reputation to be calculates, and $\bot$ represents ignorance, that is, the reputation is not known.

2.3 Reputation of an Author

A researcher's author reputation is an aggregation of the reputation of her papers. The aggregation is based on the concept that *the impact of a paper's reputation on its authors' reputation is inversely proportional to the total number of its authors*. That is, if one researcher is the sole author of a paper, then this author is the only person responsible for this paper, and any (positive or negative) feedback about this paper is propagated as is to its sole author. However, if the researcher has co-authored the paper with several other researchers, then the impact (whether positive or negative) that this paper has on the researcher decreases with the increasing number of co-authors.

$$R_A(r) = \begin{cases} \dfrac{\sum_{\forall p\in pap(r)} \gamma(p)^\gamma \times R_P(p) + (1-\gamma(p)^\gamma)\times 50}{|pap(r)|} & \text{if } pap(r) \neq \emptyset \\ \bot & \text{otherwise} \end{cases} \quad (2)$$

where $pap(r) = \{p \in P \mid r \in a(p) \wedge R_P(p) \neq \bot\}$ denotes the papers authored by a given researcher r, $\bot$ describes ignorance, $\gamma(p) = \dfrac{1}{|a(p)|}$ is the coefficient that takes into consideration the number of authors of a paper (recall that $a(p)$ denotes the authors of a paper p), and γ is a tuning factor that controls the rate of decrease of the $\gamma(p)$ coefficient. Also note the multiplication by 50, which describes ignorance, as 50 is the median of the chosen range $[0, 100]$.

2.4 Reputation of a Reviewer

The reputation of a reviewer is an aggregation of the opinions over her reviews. We assume such opinions can be obtained, in a first instance, by other reviewers that *also reviewed the same paper*. However, as this is a new feature to be introduced in open access repositories and conference and journal paper management systems, we also provide an alternative: we take 'similarity' between reviews as a measure of the reviewers opinions about reviews. For instance, we assume that 'if my review is similar to yours then I may assume your judgement of my review would be good.' We note $v^*(r_i, r_j, p) \in E$ for the 'extended judgement' of r_i over r_j's opinion on paper p, and define it as an aggregation of opinions and similarities as follows:

$$v^*(r_i, r_j, p) = \begin{cases} v(r_i, r_j, p) & \text{if } v(r_i, r_j, p) \neq \perp \\ Sim(\bar{o}(r_i, p), \bar{o}(r_j, p)) & \text{If } \bar{o}(r_i, p) \neq \perp \text{ and } \bar{o}(r_j, p) \neq \perp \\ \perp & \text{Otherwise} \end{cases} \quad (3)$$

where Sim stands for an appropriate similarity measure. We say the similarity between two opinions is the difference between the two: $Sim(\bar{o}(r_i, p), \bar{o}(r_j, p)) = 100 - |\bar{o}(r_i, p) - \bar{o}(r_j, p)|$.

Now consider the set of judgements of r_i over reviews made by r_j as: $V^*(r_i, r_j) = \{v^*(r_i, r_j, p) \mid v(r_i, r_j, p) \neq \perp \text{ and } p \in P\}$. This set might be empty. We define the judgement of a reviewer over another one as a simple average:

$$R_R(r_i, r_j) = \begin{cases} \dfrac{\sum_{\forall v \in V^*(r_i, r_j)} v}{|V^*(r_i, r_j)|} & \text{if } V^*(r_i, r_j) \neq \emptyset \\ \perp & \text{otherwise} \end{cases} \quad (4)$$

Finally, the reputation of a reviewer r, $R_R(r)$, is an aggregation of judgements that her colleagues make about her reviews. We weight this with the reputation of the colleagues as a reviewer:

$$R_R(r) = \begin{cases} \dfrac{\sum_{\forall r_i \in R^*} R_R(r_i) \cdot R_R(r_i, r)}{\sum_{\forall r_i \in R^*} R_R(r_i)} & R^* \neq \emptyset \\ 50 & \text{otherwise} \end{cases} \quad (5)$$

where $R^* = \{r_i \in R \mid V^*(r_i, r) \neq \emptyset\}$. The default, representing ignorance, is 50 (as 50 is the median of the chosen range $[0, 100]$).

2.5 Reputation of a Review

The reputation of a review is an aggregation of its judgements, weighted by the reputation of their reviewers (Section 2.4).

$$R_O(r', p) = \begin{cases} \dfrac{\sum_{\forall r \in jud(r', p)} R_R(r) \cdot v^*(r, r', p)}{\sum_{\forall r \in jud(r', p)} R_R(r)} & \text{if } |jud(r', p)| \geq k \\ R_R(r') & \text{otherwise} \end{cases} \quad (6)$$

where $jud(r', p) = \{r \in R \mid v^*(r, r', p) \neq \perp\}$ denotes the set of judges of a review written by r' on paper p, and k is a parameter specifying the minimum number of judgements needed to calculate the reputation. The default is the reputation of the author of the review (her reputation as a reviewer).

3 EVALUATION

To evaluate the effectiveness of the proposed model, we have simulated a community of researchers, using NetLogo [3]. We simulated the creation of papers, reviews, and played with the parameters that tune the true quality of researchers (both as authors and reviewers), which impacts the true quality of their papers and reviews. The aim of the evaluation was to investigate how close are the calculated reputation values to the *true* values.

The results (Table 1) illustrate how the error of the reviewers' reputation increases as the number of bad reviewers increases in the community (where the reviewer's true quality is sampled from a beta distribution specified by the parameters α_R and β_R). The results also show how the error in the papers' reputation increases with the error in the reviewers' reputation, though at a smaller rate. One curious thing about these results is the constant error in the reputation of authors. To investigate this further, we played with the number of co-authors ($\#_{CA}$). Results (Table 2 show how an increasing number of co-authors increases the error in authors' reputation.

	Error in Reviewers' Reputation	**Error in Papers' Reputation**	**Error in Authors' Reputation**
α_R=5 & β_R=1	∼ 11 %	∼ 2 %	∼ 22 %
α_R=2 & β_R=1	∼ 23 %	∼ 5 %	∼ 23 %
α_R=1 & β_R=1	∼ 30 %	∼ 7 %	∼ 23 %
α_R=0.1 & β_R=0.1	∼ 34 %	∼ 5 %	∼ 22 %
α_R=1 & β_R=2	∼ 44 %	∼ 8 %	∼ 23 %
α_R=1 & β_R=2	∼ 60 %	∼ 9 %	∼ 20 %

Table 1: Impact of reviewers' quality on ARM's performance

	Error in Reviewers' Rep.		**Error in Papers' Rep.**		**Error in Authors' Rep.**	
	α_R=5; β_R=1	α_R=1; β_R=5	α_R=5; β_R=1	α_R=1; β_R=5	α_R=5; β_R=1	α_R=1; β_R=5
$\#_{CA}$=0	∼13%	∼54%	∼3%	∼9%	∼2%	∼7%
$\#_{CA}$=1	∼13%	∼57%	∼3%	∼9%	∼12%	∼15%
$\#_{CA}$=2	∼11%	∼60%	∼2%	∼9%	∼22%	∼20%

Table 2: Impact of number of co-authors on authors' reputation

4 CONCLUSION

We have presented the ARM reputation model for the academic world that calculates the reputation of researchers, both as authors and reviewers, and their research work, as well as the reputation of reviews. The model is based on peer reviews. For further details we refer the interested reader to [2].

ACKNOWLEDGEMENTS

This work has been supported by the projects CollectiveMind (funded by MINECO, grant # TEC2013-49430-EXP), and Open Peer Review Module for Repositories (funded by OpenAIRE).

REFERENCES

[1] Nikolaus Kriegeskorte and Diana Deca, eds. *Beyond open access: visions for open evaluation of scientific papers by post-publication peer review*, Front. Comput. Neurosci., 2012.

[2] Nardine Osman and Carles Sierra, 'Reputation in the academic world', in *Proc. of Trust in Agent Societies*, eds., J. Zhang, R. Cohen, and M. Sensoy, volume 1578, pp. 1–17. CEUR-WS.org, (2016).

[3] Seth Tisue and Uri Wilensky, 'Netlogo: Design and implementation of a multi-agent modeling environment', in *Proc. of Agent2004*, pp. 161–184, (2004).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1616

Collective Future Orientation and Stock Markets

Mohammed Hasanuzzaman[1] and **Wai Leung Sze**[2] and **Mahammad Parvez Salim**[3] and **Gaël Dias**[4]

Abstract. Web search query logs can be used to track and, in some cases, anticipate the dynamics of individual behavior which is the smallest building block of the economy. We study AOL query logs and introduce a collective future intent index to measure the degree to which Internet users seek more information about the future than the past and the present. We have asked the question whether there is link between the collective future intent index and financial market fluctuations on a weekly time scale, and found a clear indication that the weekly transaction volume of S&P 500 index is correlated with the collective intent of the public to look forward.

1 INTRODUCTION

Everyday, huge amounts of data are generated through the society's extensive interactions with technological systems, automatically documenting collective human behaviour on the Internet. The enormity and high variance of information that flow through the web opened new avenues for harnessing that data and associating online human activities with offline outcomes (real world social phenomena).

Analysis of web search queries, as logged by search engines such as Google, Bing, Yahoo and AOL has received increased attention in recent years. For example, health-seeking behavior in the form of online web search queries, which are submitted to Google search by millions of users around the world are used to improve early detection of seasonal influenza [2]. A strong correlation was found between the current level of economic activity in given industries and the search volume data of industry-based query terms [1]. However, it remains unexplored whether trends in financial markets can be anticipated by the collective temporal orientation of online users.

Temporal orientation refers to differences in the relative emphasis individuals place on the past, present and future, and is a predictive indicator of many human factors such as occupational and educational success, engagement in risky behavior, financial stability, depression and health [8].

In this study, we look into temporal orientation of search queries of online users. First, we developed a model on the basis of several linguistic features to automatically detect the temporal intent (oriented towards *past, present* and *future*) of search engine queries. We use this model to classify millions of web search queries as *past, present, future* or *atemporal* . Afterwards, we calculate the ratio of the volume of searches oriented towards the *future* to the volume of searches oriented towards the past, aggregate the ratios on a weekly time scale and call this value the *collective future intent index*. We find evidence that weekly trading volumes of stocks traded in S&P 500 are correlated with the *collective future intent index* of queries. Our results provide quantitative support for the recommendation that stock market predictions can be improved by the inclusion of the specific temporal dimension of online users.

[1] ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland
[2] Lamplight Analytics Limited, email: stephen.sze@lamplight.me
[3] Techno India Group, email: parvez.salim@gmail.com
[4] Normandie University, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

2 EXPERIMENTS

2.1 Data Analysed

We use weekly aggregated transaction volumes of the S&P 500 and NASDAQ Composite stock index from 1st March, 2006 to 31st May, 2006 from Yahoo Finance[5]. AOL Query Logs distributed for non-commercial research purposes only by the AOL search engine [6] are used to compute the *collective future intent index*. This collection consists of approximately 20 million web search queries by 650,000 users over the three months from 1st March, 2006 to 31st May, 2006. The data set includes various information related to each query. We used only the query text issued by the user and the time at which the query was submitted for search.

2.2 Query Temporal Intent Classification

The idea behind query temporal intent classification is to determine whether there is a temporal dimension for users' information need [5]. This idea is pushed further in [3] and aims to determine whether the user is interested in information about the past, present or future when issuing a query or if his query has no temporal dimension. The task is to predict the temporal class $c \in$ { *past, present, future, atemporal* } of a web search query with reference to its issuing date.

For the classification task, we use the gold standard data set composed of 400 queries (300 queries for training and 100 for test) released in the context of NTCIR-11 Temporalia shared task [4]. Examples of the form $< q, d, c >$ are as: q= who was martin luther, d= Jan 1, 2013, c= past; q= amazon deal of the day, d= Feb 28, 2013, c= present; q= stock market forecast tomorrow, d= Jan 1, 2013, c= future; q=number of neck muscles, d= Feb 28, 2013, c= atemporal.

We developed a methodology based on supervised learning technique exploring several features to label the temporal class of a query. The following features are used for the classification:

- **n-grams**: 1-2 token sequences. Features are encoded simply as binary indicators of whether the n-grams has appeared in the query.
- **timexes**: The mean difference between the resolved year date of time expressions and the issue year date. Time expressions are labeled and resolved via Standford's SUTime[6]. Specific features computed include the temporal difference itself, its absolute value and binary variables indicating any temporal expressions appearing in the query text that refer to the *future*, *past* or *present*.

[5] `http://finance.yahoo.com/`
[6] `http://nlp.stanford.edu/software/sutime.html`

- **Lexica**: The relative frequency of temporal categories in the query text based on a freely available temporal resource namely TempoWordNet[7]. The features are encoded as the frequency with which a word from a temporal category has (*past, present, future*) in the text divided by the total number of tokens in the text.
- **Time-gap**: If there is no mention of time expression inside q (timely implicit query), we consider the most confident year date obtained from freely available web service GTE[8]. Finally, the features recorded include the difference between the most confident returned year date and the issue year date, its absolute value, and confidence value returned by GTE. Some examples of extracted[9] year dates and confidence values are given in Table 1.

Query	Most confident Year	Confidence value
who was martin luther	1929	0.944
amazon deal of the day	2015	0.760
release date for ios7	2013	0.893
number of neck muscles	2014	0.708

Table 1: Examples of extracted year dates and confidence values

As for classifier, we used Support Vector Machines (SVM) from the Weka platform. We consider both a linear kernel (lSVM) and a radial basis function kernel (rSVM). From cross validation over the training data, we choose L1 penalization for lSVM and L2 ($\alpha||\beta||^2$) for rSVM. Our models are evaluated over the test data provided by the organizers. Overall results are presented in Table 2.

System	Atemporal	Past	Present	Future	All
lSVM	69.2	87.5	77.5	89.5	**80.0**
rSVM	68.5	84.7	74.5	86.4	78.0

Table 2: Classification accuracy for TQIC Task

Finally, the query temporal intent classification model is applied to time-tag AOL queries according to its underlying temporal intent. Afterwards, we calculate the ratio of total number of *future* oriented queries to the aggregated number of queries tagged as *past, present* for each week of our study period and consider these values as *collective future intent index* scores. For example, the *collective future intent index* for a given week (e.g. first week of April) is calculated as the ratio of the total number of searches with *future* intent to the total number of searches with *present* plus *past* intent for the same week (i.e. first week of April).

2.3 Time Series Correlation

After calculating the *collective future intent index* scores, we are concerned with the question whether it correlates with financial market fluctuations, in particular S&P 500 index traded volumes. To answer this question, we use the cross-correlation coefficient (CCF) in a similar fashion to [7]. The CCF estimates how variables are related at different time lags. The CCF value at time lag Δt between two time-series X and Y measures the correlation of the first time-series with respect to the second time-series in dependence of a time lag parameter Δt. The CCF is calculated as:

$$R_{xy}(t,\Delta t) = \frac{E[Y_{t+\Delta t}]E[X_t]}{\sigma[Y_{t+\Delta t}]\sigma[X_t]} \quad (1)$$

In this experiment, two time-series (X, Y) are change of aggregated traded volume of stocks and change of *collective future intent index* respectively on time with week granularity. The CCF values in dependence of different time lags are presented in Table 3.

	Time Lag (weeks)						
	-3	-2	-1	0	+1	+2	+3
S&P 500 index	-0.13	0.09	0.17	**0.41**	0.16	-0.12	-0.09
NASDAQ index	-0.17	-0.06	0.14	**0.34**	0.11	-0.16	-0.18

Table 3: Time lag-dependent cross correlation between weekly changes of transaction volumes of S&P 500 and NASDAQ Composite stock index and weekly collective future intent index changes.

It is evident from the results that the *collective future intent index* scores are correlated with the the aggregated volume of transactions for all stocks in S&P 500 constituents for a time lag of week (Δt = 0 week), i.e. the present week *collective future intent index* value is significantly correlated with present week transaction volumes of the S&P 500.

Moreover, we investigate the robustness of the *collective future intent index* by considering weekly transaction volume of NASDAQ Composite stock market index for the same period. Results in Table 3 illustrate the cross correlation between weekly aggregated volume of transactions and weekly *collective future intent index* changes. The same effects can be found for NASDAQ Composite, even if the CCF scores at time lag Δt = 0 week is smaller than for the S&P 500 constituents.

3 CONCLUSION

In this study, we have introduced the *collective future intent index* and its relation on the weekly transaction volumes of stock markets. One explanation for this relationship could be the *focus on the future supports economic success*. As future work, we are in the process of collecting long-term data, and will validate the results using long-term data. Afterwards, a Fuzzy Neural Network will be used to examine the predictive power of temporal orientation on the stock market.

REFERENCES

[1] Hyunyoung Choi and Hal Varian, 'Predicting the present with google trends', *Economic Record*, **88**(s1), 2–9, (2012).

[2] 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).

[3] Hideo Joho, Adam Jatowt, and Roi Blanco, 'Ntcir temporalia: a test collection for temporal information access research', in *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion (WWW)*, pp. 845–850, (2014).

[4] Hideo Joho, Adam Jatowt, Roi Blanco, Hajime Naka, and Shuhei Yamamoto, 'Overview of ntcir-11 temporal information access (temporalia) task', in *NTCIR-11 Conference (NTCIR)*, pp. 429–437, (2014).

[5] Rosie Jones and Fernando Diaz, 'Temporal profiles of queries', *ACM Transactions on Information Systems*, **25**(3), (2007).

[6] Greg Pass, Abdur Chowdhury, and Cayley Torgeson, 'A picture of search.'.

[7] Eduardo J. Ruiz, Vagelis Hristidis, Carlos Castillo, Aristides Gionis, and Alejandro Jaimes, 'Correlating financial time series with micro-blogging activity', in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pp. 513–522, New York, NY, USA, (2012). ACM.

[8] Philip G Zimbardo and John N Boyd, 'Putting time in perspective: A valid, reliable individual-differences metric', in *Time Perspective Theory; Review, Research and Application*, 17–55, Springer, (2015).

[7] https://tempowordnet.greyc.fr/

[8] http://wia.info.unicaen.fr/GTEAspNetFlatTempCluster_Server/

[9] Extraction was processed 16th April, 2015 for illustration.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1618

Learning of Classification Models from Noisy Soft-Labels

Yanbing Xue [1] and **Milos Hauskrecht** [2]

Abstract. We develop and test a new classification model learning algorithm that relies on the soft-label information and that is able to learn classification models more rapidly and with a smaller number of labeled instances than existing approaches.

1 Introduction

While huge amounts of data in various areas of science, engineering, and every day life are available nowadays, these data alone may not be sufficient for all the learning tasks we are ultimately interested in, and additional data collection is necessary to address them. These learning tasks include classification problems in which class labels are based on subjective human assessment. Examples include various text annotation problems, annotation of images or videos, or annotation of patient cases with diseases, and many others. For all these problems annotation effort is needed to supplement the data. However, the annotation effort may often be too costly limiting the number of instances one may feasibly label. The challenge is to develop methods that can reduce the number of the labeled instances but at the same time preserve the quality of the learned models.

Here we study the sample labeling problem in binary classification settings. Our solution advances a relatively new approach to address the problem: learning with soft label information [7, 8], in which each instance is associated with a soft-label further refining the class label. Soft labels reflect the certainty of human annotators in the specific class label, such as, the probability the patient suffers from a specific disease. The benefit of soft labels is that they distinguish data instances that are strong, weak or marginal representatives of a class, and when properly used in the training phase they can help us learn better models with a smaller number of labeled samples.

In this work we assume that soft-labels given to learners by humans are probabilistic. The caveat of learning models from such labels is that humans are often unable to give consistent probabilistic assessments; a phenomenon well documented in psychology and decision making literature [6, 3]. In such a case, learning methods that are robust to 'noisy' soft-label assessments are necessary. [7, 8, 9] address the problem by using probabilistic soft-labels to first determine the relative order of examples in the training data and then build the final classification model by considering all pairwise orderings among them [5, 4]. They showed this approach is more robust to the soft-label noise than regression methods trying to directly fit probabilities. However, the limitations of their approach is that (1) the number of pairwise orderings one aims to satisfy is quadratic in the number of data points in the training data, and (2) all orderings (with both small and large soft-label differences) are treated equally.

Our objective is to develop a more efficient approach for learning models from noisy soft-label information. Our solution relies on soft-label binning. Briefly, we modify the all-pair problem formulation through binning where constraints within each bin are ignored and only constraints among data points in the different bins are enforced. This leads to a smaller number of pairwise constraints to satisfy and exclusion of constraints that are more likely corrupted by the noise. Second, we reformulate the problem of satisfying constraints among data points in different bins as an ordinal regression problem and solve it using ranking-SVM [5, 4] defined on these bins [1]. This reformulation reduces the number of constraints one has to satisfy leading to a more efficient solutions where the number constraints to satisfy is linear in the number of data instances.

[1] University of Pittsburgh, United States, E-Mail: yax14@pitt.edu
[2] University of Pittsburgh, United States, E-Mail:milos@cs.pitt.edu

2 Methodology

Our binning approach divides data instances into multiple non-overlapping bins according to their soft label information. The idea is to satisfy constraints only among entries placed in the different bins. Optimally we would like to have data entries that are in the same bin according to its probability label fall into the same bin also after the projection. We can use this to reformulate the optimization problem as an ordinal regression problem [1]. Briefly we want to find the function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ that puts the data points into bins according to their soft label. We can achieve this by having every example $\mathbf{x}$ project on the correct side of each bin boundary. For example, if the example $\mathbf{x}$ is located in ith bin, then after the projection, $f(\mathbf{x})$ should be smaller than the lower margin (boundary) of bin j in the projected space, whenever $i < j$. In general, assuming m bins labeled from 1 to m, bin boundaries $b_1, b_2, \ldots b_{m-1}$ separating them in the projected space, and bin function $bin(p_i)$ that maps the probability to the bin number (lowest probability maps to lowest number), then, after the projection, the example x_i with soft label p_i should project to value smaller than b_j whenever $bin(p_i) \leq j$, otherwise its value should be larger than b_j. Overall, for N data entries and m boundaries there are $(m-1)N$ constraints, one for each data entry/boundary pair.

In general, because of the soft label noise, we cannot expect that all the constraints will be always satisfied. We allow violations of constraints but penalize them via bin-constraint loss function. This leads to the following optimization problem:

$$\min_{\mathbf{w},w_0,\mathbf{b},\eta,\xi} \quad \frac{\mathbf{w}^T\mathbf{w}}{2} + B\sum_{i=1}^{N}\eta_i + C\sum_{j=1}^{m-1}\sum_{i=1}^{N}\xi_{j,i}$$

$$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1 - \eta_i \qquad \forall i$$

$$\mathbf{w}^T\mathbf{x}_i - b_j \leq \xi_{j,i} - 1 \qquad \forall i, j(bin(p_i) \leq j)$$

$$\mathbf{w}^T\mathbf{x}_i - b_j \geq 1 - \xi_{j,i} \qquad \forall i, j(bin(p_i) > j)$$

where $j = 1, 2, ..., m-1$ indexes bin boundaries in $\mathbf{b}$, and $i = 1, 2, ..., N$ indexes data entries. The first term in the objective function is the regularization term, the second term (single sum) defines the hinge loss with respect to binary labels, and the third term (double sum) defines the bin-constraint loss function. η_i and $\xi_{j,i}$ are non-negative slack variables permitting violations of binary class and soft-label bins respectively. B and C are constants weighting

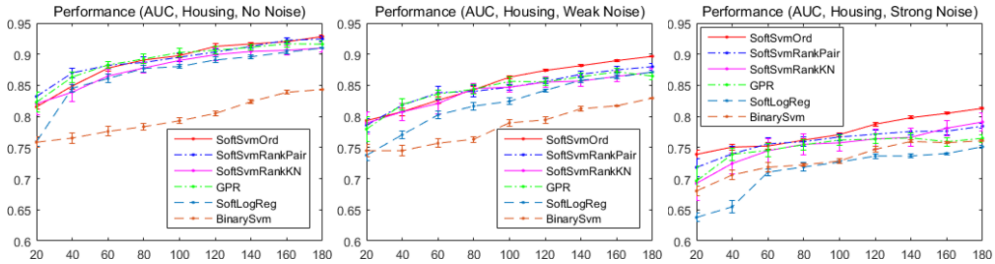


Figure 1. Performance on UCI Housing data set (no, weak, strong noise)

the objective function terms. This optimization yields a discriminant function $f(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i + w_0$ that tries to minimize the number of violated constraints, but the number of constrains is reduced to $O(mN)$ as compared to $O(N^2)$ for the pairwise-ordering methods.

One important open question is how to define the bins and how to choose their number. In our work, we use equal size binning, that is, the bin boundaries are built such that each bin covers approximately the same number of examples. The challenge, however, is to choose the number of bins. The caveat is that the number of bins may affect the quality of the result. Briefly, choosing the number of bins to be equal to N (singleton bins) reduces to $(N-1)$ bin boundaries and the total of $O(N^2)$ constraints in the optimization problem which basically mimics all pairwise orderings. On the other side, having just two bins means we are trying to separate two groups of data points, which is equivalent to binary classification. The optimal bin choice is somewhere in between these two extremes. One approach to select the number of bins is to use a heuristic. Our heuristic is inspired by the results on the optimal binning for discretization of continuous values [2] who determined that the number of bins for N examples should follow $floor(\sqrt[3]{N})$ trend.

3 Experiments and Results

We use UCI Housing data set to test our method. We normalize the real-valued outputs and reinterpreted them as probabilistic scores. We also defined a binary class threshold over the probabilistic scores to distinguish class 0 from class 1. The outputs in Housing data set represents the attractiveness of houses to the consumers. In this case, we define two classes: houses with high attractiveness (class 1) and houses with low attractiveness (class 0). We use 30% of data entries with top score to define class 1, the rest are assigned to class 0. Our experiments compare the following methods:

BinarySVM: The standard linear SVM with the hinge loss and quadratic regularization trained on examples with binary labels.

SoftLogReg: The logistic-regression-based model based on [7] that directly fits the soft-label information to the model.

GPR: The Gaussian process regression approach [10] for learning with soft-label information.

SoftSVMRankPair: The soft-label method proposed in [7] that relies on all pairwise ordering of data instances.

SoftSVMOrd: Our SVM-based ordinal regression model that splits the data into m bins based on the soft labels and enforces the bin-entry constraints. The bin size m is $floor(\sqrt[3]{N})$.

SoftSVMRankKN: A version of SoftSVMRankPair that uses a random subset of KN pairwise constraints. The value of K is selected to assure the SoftSVMOrd and SoftSVMRankKN methods always use the same number of constraints.

We evaluated the performance of the different methods by calculating the Area under the ROC (AUC) the learned classification model would achieve on the test data. Hence, each data set prior to the learning was split into the training and test set (using $\frac{2}{3}$ and $\frac{1}{3}$ of all data entries respectively). The learning considered training data only, the AUC was always calculated on the test set. To avoid potential train/test split biases, we repeated the training process (splitting) and learning steps 24 times. We report the average AUC. To test the impact of soft label information on the number of data entries, we trace the performance of all models for the different sizes N of labeled data. Figure 1(left) shows the performance of methods when simulated soft-labels are not corrupted by additional noise. The results show that all methods that rely on soft-label information outperform the SVM method trained on binary labels only. This demonstrates the sample-size benefit of soft-labels for learning classification models and basically reiterates the point made in [7]. Figure 1(left) assumes the soft labels are accurate. However, in practice, probabilistic information (when collected from humans) may be imprecise and subject to noise. In order to generate soft-label with the noise (p') we modify a soft label p derived from the UCI data by injecting a Gaussian noise of different strength. The noise injection levels indicate the average proportion of noise. Figures 1 (middle, right) show results for the noise signal at weak (10%) and strong (30%) levels respectively. The figures demonstrates that the performance of a model may drop when noise is injected. One of the methods, SoftLogReg that directly fits probabilities is particularly sensitive to the noise and its performance drops significantly for both noise levels and across all data sets. Other soft-label models that use constraints or bins are more robust and do not suffer from such a performance drop. Our new method, SoftSVMOrd, is the most consistent and tends to outperform other SVM-based models. These experiments demonstrate the robustness of our method on the soft-label learning tasks.

Acknowledgement This work was supported in part by grants R01GM088224 and R01LM010019 from the NIH. The content of the paper is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

REFERENCES

[1] W Chu et al, 'New approaches to support vector ordinal regression', in *Proc. of 22nd Int. Conf. on Machine learning*, 145-152, (2005).
[2] D Freedman et al, 'On the histogram as a density estimator', *Probability Theory and Related Fields*, **57**(4), 453-476, (1981).
[3] D Griffin et al, 'The weighing of evidence and the determinants of confidence', *Cognitive Psychology*, **24**(3), 411-435, (1992).
[4] R Herbrich et al, 'Support vector learning for ordinal regression', in *Int. Conf. on Artificial Neural Networks*, 97-102, (1999).
[5] T Joachims, 'Optimizing search engines using clickthrough data', in *Proc. of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 133-142, (2002).
[6] P Juslin et al, 'The calibration issue: Theoretical comments on suantak, bolger, and ferrell', *Organizational Behavior and Human Decision Processes*, **73**(1), 3-26, (1998).
[7] Q Nguyen et al, 'Learning classification with auxiliary probabilistic information', in *IEEE Int. Conf. on Data Mining*, 477-486, (2011).
[8] Q Nguyen et al, 'Sample-efficient learning with auxiliary class-label information', in *Proc. of Annu. American Medical Informatics Assoc. Symp.*, 1004-1012, (2011).
[9] Q Nguyen et al, 'Learning classification models with soft-label information', *J. of American Medical Informatics Assoc.*, (2013).
[10] P Peng et al, 'Learning on probabilistic labels', in *SIAM Int. Conf. on Data Mining*, 307-315, (2014).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1620

Distributed Learning in Expert Referral Networks

Ashiqur R. KhudaBukhsh and **Peter J. Jansen** and **Jaime G. Carbonell**[1]

Abstract. Human experts or autonomous agents in a referral network must decide whether to accept a task or refer to a more appropriate expert, and if so to whom. In order for the referral network to improve over time, the experts must learn to estimate the topical expertise of other experts. This paper extends concepts from Reinforcement Learning and Active Learning to referral networks, to learn how to refer at the network level, based on the proposed distributed interval estimation learning (DIEL) algorithm. Diverse Monte Carlo simulations reveal that DIEL improves network performance significantly over both greedy and Q-learning baselines [3], approaching optimal given enough data.

1 INTRODUCTION

Consider a network of experts with differing expertise, where any expert may receive a problem (aka a task or a query) and must decide whether to work on it or to refer the problem, and if so to which other expert. How can a network, or its individual experts, learn how to refer tasks effectively?

This paper proposes a new Distributed Active Learning approach in referral networks. Our referral model assumes an initial sparse topology of a referral graph where each expert knows a handful of colleagues so that $E \sim O(V)$ (E and V denote the number of edges and vertexes in the network, respectively). Learning consists of each expert improving its estimates of the ability of colleagues to solve different classes of problems. We address learning to refer comparing overall network performance contrasting an exploitation-centered (greedy optimization) with a balanced exploration-exploitation tradeoff (amortized optimization), showing that the former outperforms at first, and the latter overtakes as the network learns more effectively over time.

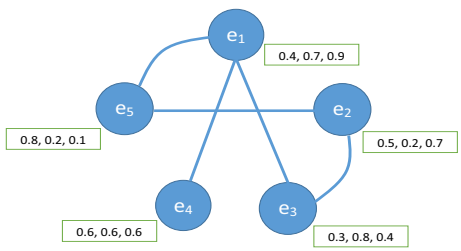


Figure 1. A referral network with five experts.

To illustrate the problem, consider the extremely simple graph, representing a five-expert network, shown in Figure 1. The nodes of the graph are the experts, and the edges indicate that the experts 'know' each other, that is, they can send or receive referrals and communicate results. We assume 3 different topics (subdomains) can be distinguished – call them t_1, t_2, and t_3 – and the figures in brackets indicate an expert's expertise in each of these.

In the example, with a query belonging to t_2, if there was no referral, the client may consult first e_2 and then possibly e_5, leading to a probability of getting the correct answer of $0.2 + (1 - 0.2) \times 0.2 = 0.36$. With referrals, an expert handles a problem she knows how to answer, and otherwise if she had knowledge of all the other experts' expertise she could ask e_2 who would refer to e_3 for the best skill in t_2, leading to a solution probability of $0.2 + (1 - 0.2) \times 0.8 = 0.84$.

Our referral mechanism consists of the following steps: 1) A user issues an *initial query* to an *initial expert*. 2) If the initial expert is able to solve it, she returns the solution; 3) if not, she selects a *referred expert* within her subnetwork, who solves or refers in turn. *Learning-to-refer* means improving the estimate of who is most likely to solve the problem.

Our primary contribution is the distributed learning-to-refer framework and a distributed learning algorithm. To learn referrals, we borrowed ideas from Reinforcement Learning [5] and Active Learning [1, 7], up till now rarely applied to referral networks, and compared performance under various conditions and learning algorithms. We extended Interval Estimation learning [4, 6] to DIEL, a distributed interval learning methods, and compare DIEL with well-known algorithms: ϵ-greedy Q-learning and double Q-learning, and with an upper-bound where every expert has access to an oracle that knows the true topic-mean and thus can refer optimally.

2 REFERRAL NETWORK

We first present our notation and assumptions.

- A set of m instances $(q_1, q_2, \ldots, q_m)$ belonging to n topics $(topic_1, topic_2, \ldots, topic_n)$ are to be addressed by k experts
- The experts are connected through a *referral network*, a graph (V, E) where each vertex v_i denotes an expert e_i and each edge $\langle v_i, v_j \rangle$ indicates a *referral link*. The probability of an edge is: $P(ReferralLink(v_i, v_j)) = \tau + c\ Sim(e_i, e_j)$. We used cosine similarity of topic means for *Sim*. The *subnetwork* of each expert e_i is the set of of all experts $\{e_j\}$ that e_i can refer to.
- The *expertise* for an expert-instance pair, $\langle e_i, q_j \rangle$, is the probability that she can successfully solve the problem, i.e., $Expertise(e_i, q_j) = P(solve(e_i, q_j))$.
  Topic-wise distributional assumption: We take the expertise distribution for a given topic t to be a mixture of two truncated Gaussians (with parameters $\lambda = \{w_i^t, \mu_i^t, \sigma_i^t\}\ i = 1, 2.$). One of them ($\mathcal{N}(\mu_2^t, \sigma_2^t)$) has higher mean ($\mu_2^t > \mu_1^t$), smaller variance ($\sigma_2^t < \sigma_1^t$) and lower mixture weight ($w_2^t << w_1^t$).
  Instance-wise distributional assumption: We model the expertise of a given expert on instances under a topic by a truncated Gaussian distribution with small variance. i.e.,
  $Expertise(e_i, q_j) \sim \mathcal{N}(\mu_{topic_p, e_i}, \sigma_{topic_p, e_i})$,
  $\forall q_j \in topic_p, \forall p, i : \sigma_{topic_p, e_i} \leq 0.2$.

[1] Carnegie Mellon University, USA, email:{akhudabu, pjj, jgc}@cs.cmu.edu

3 DISTRIBUTED REFERRAL LEARNING

Action selection using Interval Estimation Learning [2] (IEL) estimates a the upper confidence interval for the mean reward by

$$UI(a) = m(a) + t_{\frac{\alpha}{2}}^{(n-1)} \frac{s(a)}{\sqrt{n}} \quad (1)$$

where m(a) is the mean observed reward for a, $s(a)$ is the sample standard deviation of the reward, n is the number of observed samples from a, and $t_{\frac{\alpha}{2}}^{(n-1)}$ is is the critical value for the Student's t-distribution ($n-1$ degrees of freedom, $\frac{\alpha}{2}$ confidence level) (in our case, the action is the selection of a referred expert among possible choices in the subnetwork). Next, IEL selects the action with the highest upper confidence interval.

Algorithm 1 performs a single referral (per-task query budget Q = 2). The function $expR_h(e', topic)$ estimates e''s topical expertise. `DIEL` (Distributed Interval Estimation Learning), and `DMT` (Distributed Mean-Tracking) differ in h, `DIEL` estimating reward by equation (1) and `DMT` by using the sample-mean.

Input: A set of k experts e_1, e_2, ..., e_k. A set of n topics $topic_1$, $topic_2$, ..., $topic_n$. A $k \times k$ referral network.
Initialize rewards.
for *iter* $\leftarrow$ 1 ***to*** *maxIter* **do**
  Assign instance q to an initial expert e randomly
  if e fails to solve q **then**
    topic $\leftarrow$ *getTopic*(q)
    expectedReward $\leftarrow$ 0
    bestExpert $\leftarrow$ 0
    for each expert e' in the subnetwork of e **do**
      if $expR_h(e', topic) \geq$ *expectedReward* **then**
        bestExpert $\leftarrow e'$
        expectedReward $\leftarrow expR_h(e', topic)$
      end
    end
  end
  referredExpert $\leftarrow$ *bestExpert*
  if *referredExpert* solves q **then**
    update(*reward*(*e*,*topic*,*referredExpert*),1)
  else
    update(*reward*(*e*,*topic*,*referredExpert*),0)
  end
end

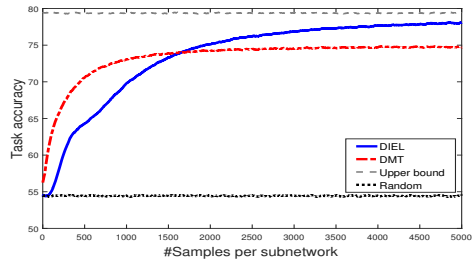
Algorithm 1: DISTRIBUTED REFERRAL LEARNING, Q = 2

4 EXPERIMENTAL SETUP AND RESULTS

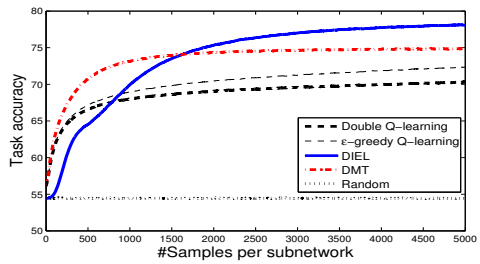
Parameter	Description	Distribution
τ	$P(ReferralLink(v_i, v_j)) = \tau + c\, Sim(e_i, e_j)$.	Uniform(0.01, 0.1)
c		Uniform(0.1,0.2)
μ_1	Truncated mixture of two Gaussians for topics	Uniform(0,b)
μ_2		Uniform(b,1) $b \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$
σ_1		Uniform(0.2,0.4)
σ_2		Uniform(0.05,0.15)
w_2		$\mathcal{N}(0.03, 0.01)$, $w_2 \geq 0$

Table 1. Parameters for our synthetic data set.

We evaluated the performance of our learning algorithms on on 1000 scenarios, each with 100 experts, 10 topics and a referral network, whose parameters are in Table 1. Our measure of performance is the overall task accuracy of the multi-expert network. As an upper bound we considered the performance achieved by a network where every expert has access to an oracle that knows the true topic-mean (i.e., $mean(Expertise(e_i, q) : q \in topic_p)\ \forall i, p$) of every expert-topic pair. Our baseline is a strategy where a task can be queried to maximum two randomly chosen experts.



(a) Comparison with baselines and upper bound.



(b) Comparison with Q-learning and DQ-learning.

Figure 2. Performance comparison of referral algorithms.

On every simulation, `DIEL` and `DMT` outperformed the baseline by a substantial margin. Over time `DIEL` clearly outperforms `DMT` and approaches the topical (oracle) upper bound. Figure 2(b) shows that both `DIEL` and `DMT` outperform Double Q-learning and ϵ-greedy Q-learning (optimized using a rough parameter sweep) for the duration of the experiment, although unlike `DMT`, Double Q-learning and ϵ-greedy Q-learning continue to improve.

REFERENCES

[1] P. Donmez, J. G. Carbonell, and P. N. Bennett, 'Dual Strategy Active Learning', *Machine Learning ECML 2007*, 116–127, (2007).
[2] P. Donmez, J. G. Carbonell, and J. Schneider, 'Efficiently learning the accuracy of labeling sources for selective sampling', *Proc. of KDD 2009*, 259, (2009).
[3] H. V. Hasselt, 'Double q-learning', in *Advances in Neural Information Processing Systems*, pp. 2613–2621, (2010).
[4] L. P. Kaelbling, *Learning in embedded systems*, MIT press, 1993.
[5] L. P. Kaelbling, M. L. Littman, and A. P. Moore, 'Reinforcement Learning: A Survey', *Journal of Artificial Intelligence Research*, **4**, 237–285, (1996).
[6] Andrew W Moore, Jeff Schneider, et al., 'Memory-based stochastic optimization', *Advances in Neural Information Processing Systems*, 1066–1072, (1996).
[7] M. Saar-Tsechansky and F. Provost, 'Active learning for class probability estimation and ranking', in *Proc. of IJCAI-01*, pp. 911–920, (2001).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1622

Transfer Learning for Automatic Short Answer Grading

Shourya Roy[1] and **Himanshu S. Bhatt**[1] and **Y. Narahari**[2]

Abstract. Automatic short answer grading (ASAG) is the task of automatically grading students answers which are a few words to a few sentences long. While supervised machine learning techniques (classification, regression) have been successfully applied for ASAG, they suffer from the constant need of instructor graded answers as labelled data. In this paper, we propose a transfer learning based technique for ASAG built on an ensemble of text classifier of student answers and a classifier using numeric features derived from various similarity measures with respect to instructor provided model answers. We present preliminary empirical results to demonstrate efficacy of the proposed technique.

1 Introduction

Assessment of student answers constructed in natural language has remained predominantly a manual job owing to multiple reasons. These include *linguistic variations* (same answer could be articulated in different ways); *subjective nature of assessment* (multiple possible correct answers or no correct answer); *lack of consistency in human rating*; etc. This paper dwells on a computational technique for automatically grading constructed student answers in natural language by focusing on *short answers*: a few words to a few sentences long [6]) and refer to the task as *Automatic Short Answer Grading (ASAG)*. Data for an example ASAG task is shown in Table 1 where the task is to automatically score the student answers.

In the next section, we intuitively describe the approach taken in this work towards developing a transfer learning based ASAG technique. In Section 3, we will describe the solution in greater detail followed by preliminary empirical evidence of benefits of the proposed approach.

Question	How are overloaded functions differentiated by the compiler? (5)
Model Ans	Based on the function signature. When an overloaded function is called, the compiler will find the function whose signature is closest to the given function call.
Stud#1	it looks at the number, types, and order of arguments in the function call.
Stud#2	By the number, and the types and order of the parameters.

Table 1. Example of question, model answer, and student answers from an undergraduate computer science course [3].

2 Approach

A large fraction of prior work in ASAG has been based on supervised learning techniques viz. classification and regression [4]. These techniques utilize features extracted from student and model answers using natural language processing (NLP) techniques reflecting *similarity* (synonymously, *overlap*, *correspondence*, *entailment* etc.) between them. These features are then fed to various classification or regression techniques to train models which can subsequently be applied to score new student answers automatically. In this work, we propose a novel supervised ASAG technique based on an ensemble of two classifiers. The first of the two is a text classifier trained using the classical TFIDF representation of bag-of-words features of student answers. It is independent of model answers and learns good textual features (words and n-grams) from graded student answers to discriminate between student answers belonging to different scores. On the other hand, features of the second classifier are a bunch of real numbers indicating similarity of student answers with the corresponding model answer. While both classifiers are trained for the same task of predicting scores for new student answers, they leverage different sets of features. Additionally, they are complimentary in nature owing to their independence and dependence on model answers respectively.

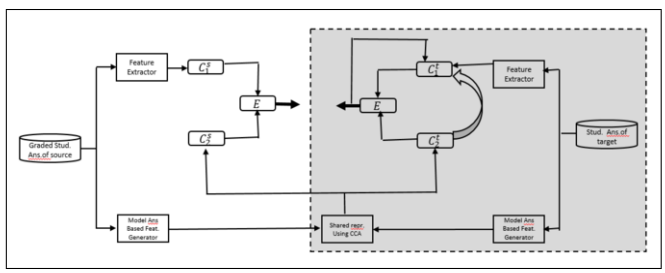


Figure 1. Block diagram of the proposed algorithm. The shaded part can be replicated for target questions for which no labelled data is available.

Supervised ASAG techniques require ongoing instructor involvement to create labelled data (by grading $\frac{1}{2}$ to $\frac{2}{3}$ of student answers as per typical train-test split) for every question and assessment task. Requirement of such continuous involvement of instructor limits the benefit of automation and thereby poses a hindrance to practical adoption. Towards addressing this limitation, we bring in the concept of *transfer learning*. Transfer learning techniques work, in contrast to traditional supervised techniques, on the principle of transferring learned knowledge across *domains*. These techniques learn a supervised model on a *source* domain with plenty of labelled data and apply to another *target* domain without (or minimal) labelled data. We formulate ASAG as a transfer learning task by considering answers to different questions as different domains as they have different marginal probability distributions of features and possibly different features too. Towards that, we propose a feature transformation based transfer learning technique using canonical correlation analysis (CCA) between the source and target questions. It transfers the trained model of the second classifier of the source question ensemble by learning a common shared representation of features which minimizes domain divergence and classification error.

[1] Xerox Research Centre India, Bangalore, India email: firstname.lastname@xerox.com
[2] Indian Institute of Science, Bangalore, India email: hari@csa.iisc.ernet.in

3 Technique

Figure 1 shows the key components of the proposed system. In this section we describe the same highlighting two main themes:

Ensemble of classifiers: We model ASAG as a supervised learning task where we employ an ensemble of two classifiers to predict student scores. The first classifier (C_1) uses the popular TFIDF vectorization on bag-of-word representations of student answers and convert to TFIDF vectors with corresponding grades as class labels. Prior to vectorization, we perform basic NLP pre-processing of stemming and stopword removal. We also perform question word demoting (i.e. considering words appearing in the question as stopwords while vectoring student answers) to avoid giving importance to parrot answering. The second classifier (C_2) is based on real-valued features capturing similarity of student answers with respect to model answer. In our endeavor towards generalizability of the proposed technique, we employ multiple generic state of the art measures to compute similarity between two pieces of short text (model and student answers) covering *lexical* (BLEU [5]), *semantic* (Wordnet based measures [3]) and *vector-space* measures (latent semantic analysis and word vectors [2]). Additionally, we would like readers to note that model of the first classifier is question specific (i.e. a word which is a good feature for a question is not necessarily a good feature for another question), whereas features for the second classifier are more question agnostic (i.e. high similarity with *respective* model answer is indicative of high scores irrespective of question). Finally, these two classifiers are combined in a weighted manner to form an ensemble (E) which is used for enhanced automatic short answer grading.

Transfer based on common representation: The ensemble of classifiers can be developed as described above for the source question based on instructor graded answers. The question is how do we do the same for target questions in absence of graded answers? It is done in two steps - (i) obtaining the second classifier through a common feature space based transfer of model from source to target followed by (ii) iteratively building the first classifier and the ensemble using pseudo labeled data.

Learning a common representation for ASAG task is based on finding a shared projection of the question agnostic features (used in the second classifier) from source and target questions. For numeric features, we used the classical canonical correlation analysis (CCA) [1] which extracts features from source and target questions such that the projected features from the two becomes maximally correlated. It learns multiple projection vectors to transform the real valued features from the source and target questions respectively to have maximum correlation. The source labeled instances are then projected onto a subspace (with the learnt projection vectors as bases) to learn a model which is subsequently used to predict labels of target instances in this subspace.

The newly trained classifier on CCA-based transformed features is the second classifier of target question. It is applied to all student answers to target question and *confidently* predicted answers are chosen as pseudo-labeled data to train the first version of the first classifier of the target question. We call this training data pool as pseudo as these are not labeled by the instructor rather based on (confident) predictions of the second classifier. This, along with the transferred second classifier are combined as an ensemble (as described above) and tested on the remaining student answers (i.e. which were not in pseudo labeled training data). Confidently predicted instances from the ensemble are subsequently iteratively used to re-train the text classifier and boost up the overall prediction accuracy of the ensemble. The iteration continues till all the examples are correctly predicted or a specified number of iterations are performed.

4 Evaluation

We empirically evaluated the proposed technique on a dataset from an undergraduate computer science course (CSD) [3] and one of its extended version (X-CSD). They consist of 21 and 87 questions respectively from introductory assignments in the course with answers provided by a class of abut 30 undergraduate students. We followed the convention in transfer learning literature of comparing against a skyline and a baseline:

- **Baseline (Sup-BL)**: Supervised models are built using labeled data from a source question and applied *as-it-is* to a target question.
- **Skyline (Sup-SL)**: Supervised models are built assuming labeled data is available for all questions (including target). Performance is measured by training a model on every question and applied on the same.

Performances of transfer learning techniques should be in between the baseline and skyline - closer to the skyline, better it is.

We use mean absolute error (MAE) as the metric for quantitative evaluation. MAE for a question is the absolute difference between groundtruth and predicted scores averaged over all students ($\frac{1}{n}\sum_{i=1}^{n}|t_i - y_i|$), where t_i and y_i are respectively the groundtruth and predicted scores of the i^{th} student's answer. For reporting one number for the dataset, the values are averaged for all questions.

Aggregated performances of ASAG techniques for the three datasets are shown in Table 3.

	CSD	X-CSD
Sup-BL	2.46	4.52
Sup-SL	0.64	0.92
Proposed	0.81	1.41

Table 2. Overall performance (MAE) of the proposed algorithm grading along with the baseline and skyline (lower the better).

The proposed method beats the baseline for both datasets handsomely (differences being 1.65 and 3.11) whereas coming much closer to the skyline (differences being 0.17 and 0.49). This demonstrates benefit of the proposed technique over supervised learning based ASAG techniques.

REFERENCES

[1] C.H. Huang, Y. R. Yeh, and Y. C. F. Wang, 'Recognizing actions across cameras by exploring the correlated subspace', in *Proceedings of International Conference on Computer Vision - Volume Part I*, pp. 342–351, (2012).

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, 'Efficient estimation of word representations in vector space', *arXiv preprint arXiv:1301.3781*, (2013).

[3] M. Mohler and R. Mihalcea, 'Text-to-text semantic similarity for automatic short answer grading', in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 567–575, (2009).

[4] Michael Mohler, Razvan C. Bunescu, and Rada Mihalcea, 'Learning to grade short answer questions using semantic similarity measures and dependency graph alignments.', in *ACL*, pp. 752–762, (2011).

[5] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, 'Bleu: a method for automatic evaluation of machine translation', Technical report, IBM Research Report, (2001).

[6] Shourya Roy, Y Narahari, and Om D Deshmukh, 'A perspective on computer assisted assessment techniques for short free-text answers', in *Computer Assisted Assessment. Research into E-Assessment*, 96–109, Springer, (2015).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1624

Long-Time Sensor Data Analysis for Estimation of Physical Capacity

Yoshikuni Sato and **Yoshihide Sawada** and **Toru Nakada**[1] and **Tadayoshi Nonoyama** and **Masafumi Kubota** and **Yusuke Koie** and **Masaki Yasutake**[2] and **Osamu Yamamura**[3]

Abstract. In this paper, we present a feature learning method for long-time sensor data. Although feature learning methods have been successfully used in many applications, they cannot extract features efficiently when the dimension of training data is quite large. To address this problem, we propose a method to search effective features from long-time sensor data. The important characteristic of our method is that it searches the features based on the gradient of input vectors to minimize the objective function of the learning algorithm. We apply our method to the estimation of physical capacity from wearable sensor data. The experimental results show that our method can estimate leg muscle strength more accurately than conventional methods using a feature learning method and current clinical index.

1 INTRODUCTION

Recently, machine learning has been successfully applied in a wide range of artificial intelligence fields, such as computer vision [5], speech recognition [4], and games [7]. State-of-the-art machine learning methods incorporate feature learning [2]. Such methods can learn classifier features from training data without any heuristics of the target application. These methods are quite effective when the dimension of training data is small. However, they cannot accept large-dimensional raw data such as sensor data measured over a long time. In such case, they need heuristic pre-processing to determine the viewpoint of the long data. This negates the advantage of the methods that they do not need heuristics of the target applications.

To address this problem, we propose a feature learning method for long-time sensor data analysis. Our method searches the effective features from sensor data efficiently by using the gradient of input vector. In this research, we will apply our method to the estimation of physical capacity from wearable sensor data. Evaluating physical capacity has become increasingly important as the demographic trends towards an older population. Our approach will enable easy and highly accurate estimation of the physical capacity of the elderly.

2 RELATED WORK

2.1 Sensor data recognition

There are many studies on activity recognition from wearable sensor data [1]. The conventional methods estimate daily activities, such as walking, standing, and running, from the acceleration and angular velocity measured by wearable sensors. Traditional methods extract statistical features from the sensor data, and predict the activity with more than approximately 80% accuracy. Recently, feature learning methods such as RBM and PCA have been applied to activity recognition [6]. These methods achieved higher accuracy compared to the methods using statistical features. However, the conventional methods are effective for "short time" activity recognition, and a method for the analysis of long-time input data has not yet been proposed.

2.2 Evaluation of Physical Capacity

Evaluation of physical capacity, particularly leg muscle strength, is important for the elderly to prevent them from falls and a bedridden state. Conventionally, leg muscle strength is measured using an isokinetic dynamometer under the supervision of a physical therapist. However, this method requires expensive devices and heavy physical load, and it is therefore not suitable for the elderly. In clinical practice, a 6-minute walk distance (6MWD) [3] is generally used for the evaluation of physical capacity. This method is quite simple but has a drawback in terms of its accuracy. In this paper, we estimate leg muscle strength from wearable sensor data obtained during a 6-minute walk test. Our approach aims at realizing easy and highly accurate estimation of the leg muscle strength, especially for the elderly.

3 METHOD

Figure 1 shows an overview of the proposed method. Our method repeats weight updating in the common training process and feature updating based on the gradient of input vectors.

In the common training phase, where such a method as logistic regression or SVM is used, we minimize the objective function based on the gradient of the weights, $w_j = w_j - \partial f(\boldsymbol{w}, \boldsymbol{x})/\partial w_j$. where $\boldsymbol{x}$ and $\boldsymbol{w}$ are input variables and weights, respectively, and $f(\boldsymbol{w}, \boldsymbol{x})$ denotes an objective function of the learning algorithm.

The characteristic of our method is that it adds the phase of calculating the gradient of input vectors, $\partial f(\boldsymbol{w}, \boldsymbol{x})/\partial x_{ij}$. Certainly, we cannot change the input vector $\boldsymbol{x}$ arbitrarily; thus, we search features to extract $\boldsymbol{x}$ that minimize the objective function of learning algorithm. In this study, we introduce two parameters t_j and p_j to extract j-th feature ($1 \leq j \leq M$). t_j and p_j denotes the time range where the feature is extracted and component number of PCA, respectively. Our method tunes these feature parameters by following equation.

$$(t_j, p_j) = \underset{(t_j, p_j)}{\operatorname{argmax}} \sum_{i}^{N} \left((x_{ij} - X(t_j, p_j)) \cdot \operatorname{sgn} \left[\frac{\partial f(\boldsymbol{w}, \boldsymbol{x})}{\partial x_{ij}} \right] \right) \quad (1)$$

[1] Advanced Research Division, Panasonic Corporation
[2] Division of Physical Therapy and Rehabilitation, University of Fukui Hospital
[3] Department of Community Medicine, Faculty of Medical Sciences, University of Fukui

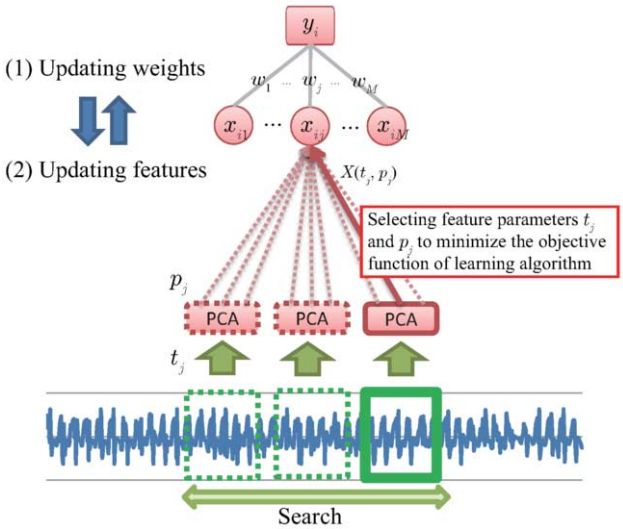


Figure 1. Procedure of proposed method.

where N, x_{ij} and $X(t_j, p_j)$ represent the number of training data, the current input variable, and the feature variable using feature parameters t_j and p_j. This procedure means that our method updates the features to change the input variables for the direction of the gradient to minimize the objective function of the learning algorithm.

The size of time window to extract a feature is set to 30sec. We slide the window by 15sec and search the best parameter of t_j and p_j. By the above procedure, our method enables the search of effective viewpoint for prediction from long time sensor data without any heuristics.

4 RESULTS

We tested our method by estimating the knee extensor strength of 102 subjects. In this experiment, we measured 3-axis accelerator, 3-axis angular velocity, heart rate, and temperature. Accelerometers and gyroscopes are attached to the waist and right ankle.

We compared the results obtained with our method and conventional methods. In this experiment, we use 6MWD [3] and PCA as baselines. 6MWD is a common index in clinical practice, and PCA is a basic feature learning method that has been successfully used in activity recognition [6]. PCA cannot accept the 6-minute sensor data directly because of the quite large dimension, and hence, we introduce the pre-processing phase that extracts averaged step wave form from the sensor data.

The number of features extracted by PCA and our method is set to 64. We use the neural network regression as a predictor in PCA and our method. In this study, we use shallow network architecture because of the number of training data. The number of hidden layers and hidden nodes is set to 1 and 32, respectively. All experiments were performed by 6-fold cross validation.

Table 1 and Figure 2 show the errors and correlations between the estimated values and the accurate values measured by an isokinetic dynamometer. The results show that the estimates using our method are much closer to the accurate values measured by an isokinetic dynamometer. The accuracy of our method is improved by more than 6 percentage points compared to the current clinical index, 6MWD, although the load to the subjects is the same.

Table 1. Errors between estimated and measured values.

Method	Error (%)
6MWD	19.9
PCA(average) + 6MWD	16.7
Proposed Method + 6MWD	**13.7**

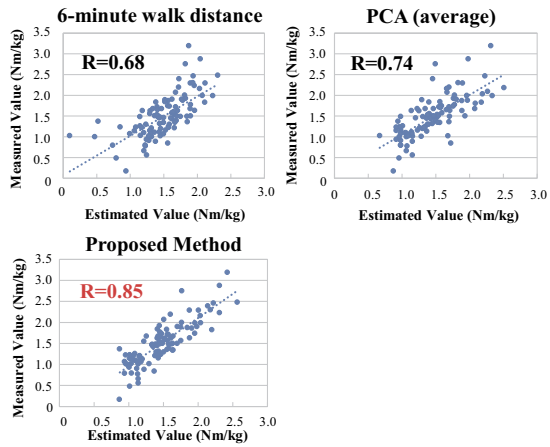


Figure 2. Correlations between estimated and measured values.

5 CONCLUSION

In this paper, we presented a feature learning method for long-time sensor data analysis. Our method searches effective features by using the gradient of input vector to minimize the objective function of the learning algorithm. We applied our method to the estimation of physical capacity from wearable sensor data. Experimental results show that our method can estimate the knee extensor strength, one of the most important indicators of physical capacity, more accurately than conventional methods. In future work, we will apply our method to the analysis of longer time of sensor data, which includes various daily activities. Our method is considered to be effective for situations that include large and complex data.

REFERENCES

[1] L. Bao and S. S. Intille, 'Activity recognition from user-annotated acceleration data.', in *Proceeedings of the 2nd International Conference on Pervasive Computing*, pp. 1–17, (2004).

[2] Y. Bengio, A. Courville, and P. Vincent, 'Representation learning: A review and new perspectives', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(8), 1798–1828, (2013).

[3] G. H. Guyatt, M. J. Sullivan, P. J. Thompson, et al., 'The 6-minute walk: a new measure of exercise capacity in patients with chronic heart failure', *Canadian Medical Association Journal*, **132**(8), 919, (1985).

[4] G. E. Hinton, L. Deng, D. Yu, et al., 'Deep neural networks for acoustic modeling in speech recognition', *IEEE Signal Processing Magazine*, **29**(6), 82–97, (2012).

[5] A. Krizhevsky, I. Sutskever, and G. Hinton, 'Imagenet classification with deep convolutional neural networks', in *Advances in Neural Information Processing Systems 25*, 1106–1114, (2012).

[6] T. Plötz, N. Y. Hammerla, and P. Olivier, 'Feature learning for activity recognition in ubiquitous computing', in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 1729–1734, (2011).

[7] D. Silver, A. Huang, C. J. Maddison, et al., 'Mastering the game of go with deep neural networks and tree search', *Nature*, **529**, 484–503, (2016).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1626

Enhancing Sketch-Based Image Retrieval via Deep Discriminative Representation

Fei Huang, Yong Cheng, Cheng Jin, Yuejie Zhang[1] and **Tao Zhang**[2]

Abstract. In this paper we aim to employ deep learning to enhance SBIR via deep discriminative representation. Our main contributions focus on: 1) The deep discriminative representation is established to bridge both the visual appearance gap and the semantic gap between sketches and images; 2) The deep learning pattern is applied to our SBIR model through training on our transformed sketch-like images to overcome the rarity of training sketches. Our experiments on a large number of public sketch and image data have obtained very positive results.

1 INTRODUCTION

Recently, Sketch-based Image Retrieval (SBIR) is an important research topic with the popularity of touch screen devices. Sketches only contain main strokes or lines of target images and lack rich texture attributes and luminance, which results in the ambiguity for the sketch query among different users. To achieve more robust SBIR, it is significant to bridge both the visual appearance gap and the semantic gap between sparse sketches and colorful images. In this paper we aim to employ deep learning to enhance SBIR via deep discriminative representation. We adopt the low-level visual feature and Convolutional Kernel Network (CKN) to capture the local visual attributes, the Convolutional Neural Network (CNN) model to capture semantic information of sketches and images, and finally the deep discriminative representation is obtained by multimodal feature fusion. Since the existing CNN models trained on the benchmark dataset of ImageNet ineffective to the sketch representation, the major barrier for SBIR is the lack of training samples. We solve this issue by implementing the training on sketch-like images transformed from real natural images. Our experiments on a large number of public data have obtained very positive results.

2 DEEP DISCRIMINATIVE REPRESENTATION

We aim at constructing the deep discriminative representation based on the specific consideration of both low-level visual features and high-level semantic features, which can be achieved by the multimodal fusion of CNN features and local features. Before that, the sketch-like images are generated by the SE detector [1] for original images to bridge the visual domain gap.

Local Feature Descriptor

To construct the local feature descriptor with the stronger descriptive ability, we choose the BoVW framework but adopt Convolutional Kernel Networks (CKNs) [2] to learn the local convolutional features as visual words, called as BCKN.

CKN takes an image or a patch as an input. The feature representation is based on a positive-definite kernel function which can be approximated as $<\xi, \xi'>$, where $<,>$ represents the Euclidean inner-product function. Let Ω be the coordinates from the input and $z \in \Omega$, Ω_1 be a subset of Ω and $|\Omega_1|<|\Omega|$, and ξ is computed for all $u \in \Omega_1$, shown as follows:

$$\xi = \sqrt{\frac{2}{\pi}} \sum_{z \in \Omega} e^{-\frac{\|u-z\|^2}{\beta^2}} g(z) \quad (1)$$

$$g(z) = \| \varphi(z) \| \left[\sqrt{\eta_l} e^{-\frac{1}{\alpha^2}\|\tilde{\varphi}(z)-w_l\|^2} \right]_{l=1}^{p} \quad (2)$$

where $\varphi(z)$ represents a fixed sub-patch centered at z ; $\tilde{\varphi}(z)$ is a normalized version of $\varphi(z)$; α and β are two smoothing parameters of the Gaussian kernel; and p is the number of filters. ξ' is computed in the same way as ξ. The stochastic gradient descent is used to optimize the parameters $[w_l]_{l=1}^{p}$ and $[\eta_l]_{l=1}^{p}$ as follows:

$$\min_{w,\eta} \frac{1}{n} \sum_{i=1}^{n} \left[e^{-\frac{1}{2\alpha^2}\|x_i-y_i\|^2} - \sum_{j=1}^{p} \eta_j e^{-\frac{1}{\alpha^2}\|x_i-w_j\|^2} e^{-\frac{1}{\alpha^2}\|y_i-w_j\|^2} \right]^2 \quad (3)$$

where $\{(x_i, y_i)\}_{i=1, \ldots, n}$ is n pairs of training patches.

In practice, we construct the single-layer CKN using Formula (1) and (2). Such the single-layer CKN can be superimposed on each other as the multi-layer CKN to acquire the better representation. The above strategy can provide a robust local representation for our low-level features. The BoVW framework is applied to encode the local features to the global representation. First, with the sketch or sketch-like images, a set of CKN features are extracted with the (16 ×16) windows centered at interest (contour) points to learn a visual codebook using *k*-means. Then, each sketch or sketch-like image can be quantized to a feature vector representation with BCKN.

CNN-based Feature Descriptor

Semantic information is very important for SBIR, especially when the distortion of freehand sketches is difficult to overcome only by using low-level local features. To fully capture the discriminative semantic information, we utilize the famous Alexnet [3] and GoogLeNet [4], which recently achieve the impressive performance for CBIR, and are extended to learn the deep semantic representation for both sketches and sketch-like images in SBIR.

Since sketches and images are two domains of visual exhibitions, the *Alexnet*/*GoogLeNet* model trained on images cannot be directly applied to SBIR. In particular, we need to solve the following two crucial issues: 1) The existing SBIR datasets are especially uneven with a majority of color images, thus there is a lack of sketch training

[1] School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China, email: {15210240036;13110240027; 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

samples; and 2) With the absence of training samples, how to extend the classification-based CNN models to SBIR without losing the discrimination? For the first issue, we can rationally assume that the obtained sketch-like images retain the main outlines of initial images, which can be approximately regarded as hand-drawn sketches. Thus such sketch-like images can be used as training samples to retrain a CNN model for SBIR. Furthermore, a lot of images on the sharing websites are annotated with tags, such as *Flickr*, which can provide relatively affluent labels for training images. Here, we only consider an image with only one category label for simplicity. For the second issue, we extract the prediction scores of the last layer in both the retrained *Alexnet* model and *GoogLeNet* model as high-level semantic features, since they can capture the discriminative semantic information from both sketches and sketch-like images.

Multimodal Feature Fusion

We combine the two cosine distances of the visual feature D_{vis} and semantic feature D_{sem} through the multimodal feature fusion to form the final ranking list as follows.

$$D_{final} = \alpha \cdot D_{vis} + (1-\alpha) \cdot D_{sem} \quad (4)$$

where $\alpha \in [0,1]$, and we usually set $\alpha = 0.5$ to balance the effect of two kinds of features in practice.

3 EXPERIMENT AND ANALYSIS

To evaluate the effectiveness of our approach, we use one benchmark dataset *Flickr15k* created by Hu *et al*. [5] (including 330 sketch queries and 14,660 images), and another dataset for a specific application scenario *MECD* (including 30 sketch queries and 900 images) from a large Chinese museum collection image set. The official criteria of Mean Rank Precision (MRP) and Average Precision (AP) are introduced to evaluate the whole retrieval performance. *MRP* aims at measuring the precision of the top-*k* returned relevant images, which is defined as:

$$MRP(k) = \sum_{i=1}^{N} Precision_i(k) / N$$
$$Precision_i(k) = n(T) / k \quad (5)$$

where $n(T)$ is the number of relevant images in the top-*k* returned results; and N is the total number of sketch queries. *AP* considers the order that the returned images are presented, which is defined as:

$$AP(k) = \frac{1}{k}\sum_{r=1}^{k} rel(r) \times MRP(r) \quad (6)$$

where $rel(r) \in \{0, 1\}$ is an indicator function, if the image at Rank r is relevant to the sketch query $rel(r)=1$, otherwise $rel(r)=0$.

Our approach is a new exploration for taking full advantage of semantic information for images, and a new deep discriminative representation framework is proposed. We introduce three popular *CNN* models, that is, *Alexnet*, *GoogLeNet* and *Siamese Network* [6], and retrain them on our sketch-like images. For *Siamese Network*, we adopt the same architecture in [6]. The *Baseline*[*BCKN*] only uses our local *BCKN* features, while *DDR*[*GoogLeNet*] and *DDR*[*Alexnet*] implement the complete framework. The related comparison results are presented in Table 1. An instantiation of four sketch queries and their relevant images is shown in Figure 1.

It can be found from Table 1 that the supervised CNN outperform the unsupervised BCKN with absolute advantage on both datasets. The best performance can be acquired by our complete approach on *Flickr15k* and *MECD*. Comparing the results based on *Alexnet*, *GoogLeNet* and *Siamese Network*, *Siamese Network* obtains the worst results because it exploits the pair-wise image similarity or dissimilarity as the supervised information, which is much weaker than the direct labeling information. By comparing the performance of our *DDR*-based approach with *CNN* models and *Baseline*[*BCKN*], there is something indicating that it is effective to jointly use both low-level visual feature and high-level semantic feature for SBIR.

Table 1. The comparison results between our and the other approaches.

Dataset	Approach		Feature Dimension	Evaluation Metric	
				MRP(20)	*AP*(20)
Flickr15k	Other Approach	*GoogLeNet*	1,000	0.701	0.694
		Alexnet	1,000	0.646	0.605
		Siamese	64	0.492	0.485
	Our Approach	*Baseline*[*BCKN*]	300	0.285	0.274
		DDR[*GoogLeNet*]	-	***0.717***	***0.715***
		DDR[*Alexnet*]	-	***0.686***	***0.671***
MECD	Other Approach	*GoogLeNet*	1,000	0.715	0.707
		Alexnet	1,000	0.662	0.642
		Siamese	64	0.514	0.493
	Our Approach	*Baseline*[*BCKN*]	300	0.335	0.305
		DDR[*GoogLeNet*]	-	***0.730***	***0.738***
		DDR[*Alexnet*]	-	***0.705***	***0.684***

Figure 1. An instantiation of some retrieval results with our approach.

4 CONCLUSIONS AND FUTURE WORK

In this work, we present a novel scheme with deep discriminative representation for SBIR. We apply the deep learning pattern and propose an effective deep discriminative representation to encode both low-level and high-level features for sketches and sketch-like. We believe this is just the beginning to extend deep learning to SBIR, and in the future we will further explore the unsupervised SBIR-based CNN framework to further boost the retrieval performance.

5 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 Zhuoxue Program of Fudan University. Yuejie Zhang is the corresponding author.

REFERENCES

[1] P. Dollár and C.L. Zitnick. 'Structured forests for fast edge detection'. *In Proceedings of ICCV 2013*, 1841-1848, (2013).

[2] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid. 'Convolutional kernel networks'. *In Proceedings of NIPS 2014*, 2627-2635, (2014).

[3] A. Krizhevsky, I. Sutskever, and G.E. Hinton. 'Imagenet classification with deep convolutional neural networks'. *In Proceedings of NIPS 2012*, 1097-1105, (2012).

[4] C. Szegedy, W.Liu, Y.Jia, P.Sermanet, S.Reed, D.Anguelov, D.Erhan, V.Vanhoucke, A.Rabinovich. 'Going deeper with convolutions'. In *Proceedings of CVPR 2015*, 1-9, (2015).

[5] R. Hu and J. Collomosse. 'A performance evaluation of gradient field hog descriptor for sketch based image retrieval'. *Computer Vision and Image Understanding*, 117(7):790-806, (2013).

[6] F.Wang, L.Kang, Y.Li. 'Sketch-based 3d shape retrieval using convolutional neural networks'. In *Proceedings of CVPR 2015,* 1875-1883, (2015).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1628

Structure in the Value Function of Two-Player Zero-Sum Games of Incomplete Information

Auke J. Wiggers[1] and **Frans A. Oliehoek**[2] and **Diederik M. Roijers**[3]

Abstract. In this paper, we introduce a new formulation for the value function of a zero-sum Partially Observable Stochastic Game (zs-POSG) in terms of a 'plan-time sufficient statistic', a distribution over joint sets of information. We prove that this value function exhibits concavity and convexity with respect to appropriately chosen subspaces of the statistic space. We anticipate that this result is a key pre-cursor for developing solution methods that exploit such structure. Finally, we show that the formulation allow us to reduce a finite zs-POSG to a 'centralized' model with shared observations, thereby transferring results for the latter (narrower) class of games to games with individual observations.

1 Introduction

The zero-sum Partially Observable Stochastic Game (zs-POSG) is a model for multi-agent decision making under uncertainty in zero-sum sequential games where the state changes over time, and the agents simultaneously choose actions at every stage based on individual observations. In this work, we prove the existence of structural properties of the zs-POSG value function, which may be exploited to make reasoning about these models more tractable.

We take inspiration from recent work for collaborative settings which has shown that it is possible to summarize the past joint policy using so called plan-time sufficient statistics [6], which can be interpreted as the belief of a special type of Partially Observable Markov Decision Process (POMDP) to which the collaborative Decentralized POMDP (Dec-POMDP) can be reduced [1, 4, 5]. This enabled tackling these problems using solution methods for POMDPs, leading to increases in scalability [1].

We extend these results for Dec-POMDPs to the zs-POSG setting by presenting three contributions. First, a definition of the value function of a zs-POSG in terms of distributions over information called *plan-time sufficient statistics*. Second, a proof that the formulation allows for a generalization over the statistics: on every stage, the value function exhibits concavity and convexity in different *subspaces* of statistic-space. Third, a reduction of the zs-POSG to a *Non-Observable Stochastic Game*, which in turn allows us to show that certain properties previously proven for narrower classes of games generalize to the more general zs-POSG considered here. This is the first work that gives insight in how the value function of a zs-POSG generalizes over the space of plan-time sufficient statistics. We argue that this result may open up the route for new solution methods.

[1] Scyfer B.V., email: auke@scyfer.nl
[2] University of Liverpool , Universiteit van Amsterdam, email: Frans.Oliehoek@liverpool.ac.uk
[3] University of Oxford, email: diederik.roijers@cs.ox.ac.uk

2 Model definition

Definition 1. A **finite zs-POSG** is defined as a tuple $\mathcal{P} = \langle h, I, \mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, R, b^0 \rangle$:

- h is the (finite) horizon,
- $I = \{1, 2\}$ is the set of 2 agents,
- $\mathcal{S}$ is the finite set of states s,
- $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ is the finite set of joint actions $a = \langle a_1, a_2 \rangle$,
- $\mathcal{O} = \mathcal{O}_1 \times \mathcal{O}_2$ is the finite set of joint observations $o = \langle o_1, o_2 \rangle$,
- T is the transition function $\Pr(s^{t+1}|s^t, a^t)$,
- O is the observation function $\Pr(o^{t+1}|s^{t+1}, a^t)$,
- $R : S \times \mathcal{A} \times S \to \mathbb{R}$ is the reward function for agent 1,
- $b^0 \in \Delta(\mathcal{S})$ is the initial probability distribution over states.

In the zs-POSG, we aim to find *rational strategies* (i.e., maxmin-strategies) and the *value* of the game (i.e., the expected sum of rewards when agents follow the said strategies). Let a *pure policy* for agent i be a mapping from individual action-observation histories (AOHs) $\vec{\theta}_i^t = \langle a_i^0, o_i^1, \ldots, a_i^{t-1}, o_i^t \rangle$ to actions. Let a *stochastic policy* for agent i be a mapping from individual AOHs to a probability distribution over actions, denoted as $\pi_i(a_i^t|\vec{\theta}_i^t)$. An individual policy defines action selection of one agent on every stage of the game, and is essentially a sequence of individual *decision rules* (one-stage policies) $\pi_i = \langle \delta_i^0 \ldots \delta_i^{h-1} \rangle$. We define the *past individual policy* as a tuple of decision rules $\varphi_i^t = \langle \delta_i^0, \ldots, \delta_i^{t-1} \rangle$, and define the tuple containing decision rules from stage t to h as the *partial individual policy* $\pi_i^t = \langle \delta_i^t, \ldots, \delta_i^{h-1} \rangle$. Rational policies are denoted π^*.

We assume *perfect recall*, i.e., agents recall their own past actions and observations, and assume that all elements of the game are *common knowledge* among the agents. We will use the term 'value function' for a function that captures the future expected rewards under a rational joint policy.

3 Structure in zs-POSG Value Function

It is theoretically possible to convert a zs-POSG to *normal form* and solve it using standard methods, but this is infeasible in practice. An alternative is to convert the zs-POSG to an extensive form game (EFG) and solve it in sequence form [3]. While this is more efficient than the NFG route, it is often still intractable: the resulting EFG is huge since its size depends on the number of full histories (trajectories of joint actions, joint observations, and states) [8]. Instead, in this section, we give a value function formulation in terms of a so-called plan-time sufficient statistic (originally used in the collaborative Dec-POMDP setting [6]). We show that this value function exhibits a potentially exploitable structure at every stage of the game.

Definition 2. The **plan-time sufficient statistic** for a general past joint policy φ^t, assuming b^0 is known, is a distribution over joint AOHs: $\sigma^t(\vec{\theta}^t) \triangleq \Pr(\vec{\theta}^t | b^0, \varphi^t)$.

The zs-POSG Q-value function at the final stage $h-1$ reduces to the immediate reward function: We define the Q-value for all other stages as:

$$Q_t^*(\sigma^t, \vec{\theta}^t, \delta^t) \triangleq R(\vec{\theta}^t, \delta^t) + \sum_{a^t} \sum_{o^{t+1}} \Pr(\vec{\theta}^{t+1} | \vec{\theta}^t, \delta^t) \\ Q_{t+1}^*(\sigma^{t+1}, \vec{\theta}^{t+1}, \delta^{t+1*}). \quad (1)$$

$$Q_t^*(\sigma^t, \delta^t) \triangleq \sum_{\vec{\theta}^t} \sigma^t(\vec{\theta}^t) Q_t^*(\sigma^t, \vec{\theta}^t, \delta^t). \quad (2)$$

Here, σ^{t+1} is found using the statistic update rule: $\sigma^{t+1}(\vec{\theta}^{t+1}) \triangleq \Pr(o^{t+1} | \vec{\theta}^t, a^t) \delta^t(a^t | \vec{\theta}^t) \sigma^t(\vec{\theta}^t)$.

Lemma 1. *σ^t is a sufficient statistic for the value of the zs-POSG, i.e. $Q_t^*(\sigma^t, \vec{\theta}^t, \delta^t) = Q_t^*(\varphi^t, \vec{\theta}^t, \delta^t), \forall t \in 0 \dots h-1, \forall \vec{\theta}^t \in \vec{\Theta}^t, \forall \delta^t$.*

Proof. See [9]. □

We can now define the value function of the zs-POSG in terms of σ^t:

$$V_t^*(\sigma^t) \triangleq \max_{\delta_1^t \in \Delta_1^S} \min_{\delta_2^t \in \Delta_2^S} Q_t^*(\sigma^t, \langle \delta_1^t, \delta_2^t \rangle). \quad (3)$$

Although we have identified the value at a single stage of the game, implementing a backwards inductive approach directly is still not possible, since the space of statistics is continuous and we do not know how to represent $V_t^*(\sigma^t)$. This paper takes a first step at resolving this problem by investigating the structure of $V_t^*(\sigma^t)$.

We decompose the plan-time sufficient statistic in marginal and conditional terms for agent 1 as $\sigma^t(\langle \vec{\theta}_1^t, \vec{\theta}_2^t \rangle) = \sigma_{m,1}^t(\vec{\theta}_1^t) \sigma_{c,1}^t(\vec{\theta}_2^t | \vec{\theta}_1^t)$ (similar for agent 2), and show that the value function is concave in marginal-space for agent 1, $\Delta(\vec{\Theta}_1^t)$, and convex in marginal-space for agent 2, $\Delta(\vec{\Theta}_2^t)$.

Theorem 1. *V_t^* is concave in $\Delta(\vec{\Theta}_1^t)$ for a given $\sigma_{c,1}^t$, and convex in $\Delta(\vec{\Theta}_2^t)$ for a given $\sigma_{c,2}^t$.*

Proof. See [9]. □

Figure 1 provides intuition on how the concepts are related. Each 'slice' in statistic-space corresponds to a single conditional $\sigma_{c,1}^t$ ($\sigma_{c,2}^t$) and a marginal-space. The value function exhibits a concave (convex) shape on this slice, and is comprised of linear segments that each correspond to a partial policy π_i^t of the opposing agent.

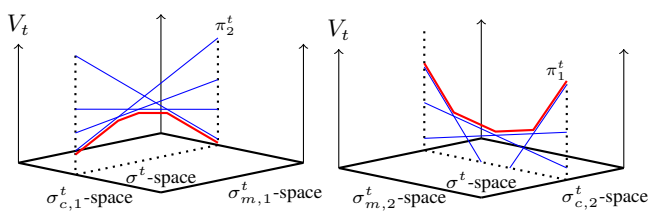


Figure 1: An abstract visualization of the decomposition of statistic-space into marginal-space and conditional-space.

The importance of this theorem is that it suggests ways to (approximately) represent $V_t^*(\sigma^t)$. As our formulation preserves the structure of the game, it allows us to make statements about how *value generalizes as a function of the information distribution*. Thus, it may enable the development of new solution methods for zs-POSGs.

4 Reduction to NOSG

Similar to how the use of sufficient statistics allows a Dec-POMDP to be reduced to a special type of (centralized) POMDP [1, 4, 5, 7], it turns out that, through the use of sufficient statistics, any finite zs-POSG can be reduced to a game to which we refer as a *Non-Observable Stochastic Game* (NOSG): a stochastic game with a single, shared NULL observation, where the joint AOH acts as the state. We give the full NOSG definition in [9].

In the NOSG model, agents condition their choices on the joint belief over augmented states $\dot{b} \in \Delta(\dot{S})$, which corresponds to the belief over joint AOHs captured in the statistic $\sigma^t \in \Delta(\Theta^t)$. As such, a value function formulation for the NOSG can be given in accordance with (3). This indicates that properties of 'zero-sum stochastic games with shared observations' [2] also hold for finite zs-POSGs.

5 Conclusions

We present a structural result on the shape of the value function of two-player zero sum Partially Observable Stochastic Games (zs-POSG). We define the zs-POSG value function in terms of an information distribution called the sufficient plan-time statistic, and prove that this value function exhibits concavity (convexity) in the space of marginal statistics of the maximizing (minimizing) agent. Thus, our formulation enables us to make statements about how value generalizes as a function of the information distribution. Lastly, we showed how the results allow us to reduce our finite zs-POSG to a stochastic game with shared observations, thereby transferring properties of this narrower class of games to the finite zs-POSG case.

Acknowledgments This research is supported by the NWO Innovational Research Incentives Scheme Veni (#639.021.336) and NWO DTC-NCAP (#612.001.109) project.

REFERENCES

[1] Jilles S. Dibangoye, Christopher Amato, Olivier Buffet, and François Charpillet, 'Optimally solving Dec-POMDPs as continuous-state MDPs', in *Proceedings of the International Joint Conference on Artificial Intelligence*, (2013).

[2] MK Ghosh, D McDonald, and S Sinha, 'Zero-sum stochastic games with partial information', *Journal of Optimization Theory and Applications*, **121**(1), 99–118, (2004).

[3] Daphne Koller, Nimrod Megiddo, and Bernhard Von Stengel, 'Fast algorithms for finding randomized strategies in game trees', in *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pp. 750–759. ACM, (1994).

[4] Liam C. MacDermed and Charles Isbell, 'Point based value iteration with optimal belief compression for Dec-POMDPs', in *Advances in Neural Information Processing Systems 26*, pp. 100–108, (2013).

[5] Ashutosh Nayyar, Aditya Mahajan, and Demosthenis Teneketzis, 'Decentralized stochastic control with partial history sharing: A common information approach', *IEEE Transactions on Automatic Control*, **58**, 1644–1658, (July 2013).

[6] Frans A. Oliehoek, 'Sufficient Plan-Time Statistics for Decentralized POMDPs', in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pp. 302–308, (2013).

[7] Frans A. Oliehoek and Christopher Amato, 'Dec-POMDPs as non-observable MDPs', IAS technical report IAS-UVA-14-01, Intelligent Systems Lab, University of Amsterdam, Amsterdam, The Netherlands, (October 2014).

[8] Frans A. Oliehoek and Nikos Vlassis, 'Dec-POMDPs and extensive form games: equivalence of models and algorithms', *Ias technical report IAS-UVA-06-02, University of Amsterdam, Intelligent Systems Lab, Amsterdam, The Netherlands*, (2006).

[9] A. J. Wiggers, F. A. Oliehoek, and D. M. Roijers, 'Structure in the Value Function of Two-Player Zero-Sum Games of Incomplete Information', *ArXiv e-prints*, (June 2016). `http://arxiv.org/abs/1606.06888`.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1630

GDL-III: A Proposal to Extend the Game Description Language to General Epistemic Games

Michael Thielscher[1]

Abstract. We propose an extension of the standard game description language for general game playing to include *epistemic games*, which are characterised by rules that depend on the knowledge of players. A single additional keyword suffices to define *GDL-III*, a general description language for games with *imperfect information* and *introspection*. We present an Answer Set Program for automatically reasoning about GDL-III games. Our extended language along with a suitable basic reasoning system can also be used to formalise and solve general epistemic puzzles.

Introduction. A general game player is a system that can understand the rules of new strategy games at runtime and learn to play these games effectively without human intervention [4]. The game description language GDL has become the standard for describing the rules of games to general game-playing systems [3].

The extension GDL-II has been developed with the aim to include general *imperfect information* games [5]. While these games require general game players to reason about their knowledge of the state of a game, GDL-II does not support the specification of games with *epistemic* goals or, more generally, with rules that depend on the epistemic state of players. As an example, consider the game NUMBER-GUESSING from the GDL-II track at the AI'12 general game playing competition,[2] in which the goal for a single player is to repeatedly ask yes/no questions to determine an initially unknown number. However, the player can win merely by guessing correctly; the game description language does not provide means to specify, as a necessary winning condition, that the player must actually *know* the number. Another example of games beyond the expressiveness of GDL-II are so-called Russian Card problems [2], where the goal of two cooperating players is to inform each other about their hands through public announcements without a third player being able to learn anything from their communication.

The purpose of this paper is to propose a formal language suitable for general game playing that supports the encoding of game rules which depend on the epistemic states of the players. We will show that a single additional keyword suffices to define *GDL-III*, a general description language for games with *Imperfect Information* and *Introspection*. The new keyword can be used to express individual knowledge, which can also be nested (e.g. player A knows that her cards are known to player B) as well as common knowledge (e.g. player C does not know of any card held by another player and everyone knows this—and also everyone knows that everyone knows etc). While the main purpose of our language extension is to allow for the description of epistemic games for the purpose of general game playing, we will furthermore demonstrate how GDL-III can be used to encode, and automatically solve, epistemic puzzles like Cheryl's Birthday, which recently acquired public fame [1].

Game Descriptions. The declarative Game Description Language (GDL) uses a prefix-variant of the syntax of normal logic programs along with the following special keywords, the last two of which have been added in GDL-II for imperfect-information games.

`(role R)`	`R` is a player
`(init F)`	feature `F` holds in the initial position
`(true F)`	feature `F` holds in the current position
`(legal R M)`	`R` has move `M` in the current position
`(does R M)`	player `R` does move `M`
`(next F)`	feature `F` holds in the next position
`terminal`	the current position is terminal
`(goal R V)`	player `R` gets payoff `V`
`(sees R P)`	player `R` is told `P` in the next position
`random`	the random player (aka. Nature)

GDL-III = GDL-II + Introspection. We define GDL-III as an extension of GDL-II by a new keyword for *introspection*,

`(knows R P)`	player `R` knows `P` in the current position
`(knows P)`	`P` is common knowledge

along with the following additional restrictions on syntactically valid game descriptions G:

1. `knows` only occurs in the body of clauses, and neither `role` nor `init` depend on `knows`.
2. There is a total ordering $>$ on all predicate symbols `P` that occur as argument of `knows` in G such that `P` $>$ `Q` whenever `P` itself depends on `(knows R Q)` or `(knows Q)` in G.
3. If `P` occurs as argument of `knows` in G then `P` does not depend on `does` in G.

Note the use of *reification*, whereby a defined predicate, `P`, occurs as an argument of another predicate. *Nested* knowledge can be expressed with the help of auxiliary predicates, for example, `(knows a kbp)` along with `(<= kbp (knows b p))`. The syntactical restrictions then ensure that nested knowledge is hierarchical (condition 2) and confined to state-dependent properties (condition 3). The former simply disallows circular definitions, as in `(<= p (knows q))`, `(<= q (knows p))`, while the latter restriction ensures that knowledge only refers to the current state and not to future actions.

[1] UNSW Australia, email: mit@unsw.edu.au. This research was supported by the Australian Research Council under grant no. DP150101351. The author is also affiliated with the University of Western Sydney.
[2] see `ai2012.web.cse.unsw.edu.au/ggp.html`

```
1  (role player) (role random)
2
3  (number 1)  ...  (number 32)
4  (succ 0 1)  ...  (succ 31 32)
5  (<= (less ?m ?n) (or (succ ?m ?n)
6                       ((succ ?m ?l) (less ?l ?n))))
7  (init (step 0))
8
9  (<= (legal random (choose ?n))
10     (number ?n) (true (step 0)))
11 (<= (legal random noop) (not (true (step 0))))
12
13 (<= (legal player noop) (true (step 0)))
14 (<= (legal player (ask_if_less ?n))
15     (number ?n) (not (true (step 0))))
16 (<= (sees player yes)
17     (does player (ask_if_less ?n))
18     (true (secret ?m)) (less ?m ?n))
19
20 (<= (next (secret ?n)) (does random (choose ?n)))
21 (<= (next (secret ?n)) (true (secret ?n)))
22 (<= (next (step ?n))   (true (step ?m)) (succ ?m ?n))
23
24 (<= (num ?n) (true (secret ?n)))
25 (<= (kows_the_number ?r) (role ?r) (knows ?r (num ?n)))
26
27 (<= (terminal) (or (knows_the_number player)
28                     (true (step 12))))
29 (<= (goal player 100) (knows_the_number player))
30 (<= (goal player   0) (not (knows_the_number player)))
```

Example 1 *The rules on the top of this page formalise a GDL-III variant of* NUMBERGUESSING*. Line 1 introduces the roles. Lines 3–5 define some auxiliary predicates. Line 7 provides the initial game state. The moves are specified by the rules for* `legal`*: In the first round, a number between 1 and 32 is randomly chosen (lines 9–10). The player can then repeatedly ask yes/no questions (lines 14–15). The player's percepts are truthful replies to these questions (lines 16–18). The rules for* `next` *specify the state update (lines 20–22). The objective of the game is formulated as a* knowledge goal *(lines 24–30): the game ends (*`terminal`*) and the player wins (*`goal` *value 100) upon* knowing *the secret number.*

Automated Reasoning for GDL-III. In order to be able to play games specified in the extended game description language, any general game-playing system needs an automated reasoning component for evaluating the rules to determine legal moves and compute state updates. In this section, we build on previous uses of Answer Set Programming (ASP) to develop the foundations for automated reasoners for GDL-III. The game description language and ASP have essentially the same basic semantics given by the unique stable model (aka. answer set) of a stratified set of logic program rules. Since the evaluation of knowledge conditions depends on previous moves and percepts, all state-dependent predicates in a game description are augmented by two additional arguments so that a single ASP can be used to reason about different legal play sequences, `seq(S)`, and different time points, `time(T)`. For example, the ASP-encoding of rule 25 in NUMBERGUESSING is

```
knows_the_number(R,S,T) :-
  seq(S), time(T),
  role(R), knows(R,num(N),S,T).
```

We follow the convention of using natural numbers for time, so that `(init F)` is replaced by `true(F,S,0)` and `(next F)` by `true(F,S,T+1)`. Knowledge conditions are evaluated on the basis of the (in-)distinguishability of legal play sequences, according to which players can distinguish any two sequences in which at least one of their preceding moves or percepts differ. Otherwise, the two sequences are indistinguishable (predicate `ind`):

```
distinguishable(R,S1,S2,N) :- time(N), T<N,
  does(R,M1,S1,T), does(R,M2,S2,T), M1!=M2.
distinguishable(R,S1,S2,N) :- time(T), T<=N,
  sees(R,P,S1,T), not sees(R,P,S2,T).

ind(R,S1,S2,N) :- role(R), seq(S1), seq(S2),
  time(N), not distinguishable(R,S1,S2,N).

indtrans(S1,S1,N) :- seq(S1), time(N).
indtrans(S1,S3,N) :- ind(R,S1,S2,N),
                     indtrans(S2,S3,N).
```

The last two clauses above encode the transitive closure of the indistinguishability relation over all roles. On this basis, conditions `(knows R` $p(\vec{x})$`)` can be evaluated according to the schema

```
knows(R,p(x⃗),S,T) :- p(x⃗,S,T), not np(R,x⃗,S,T).
np(R,x⃗,S,T) :- ind(R,S,S1,T), not p(x⃗,S1,T).
```

Put in words, if `S` is the actual play sequence at time `T`, then player `R` knows that $p(\vec{x})$ just in case $p(\vec{x})$ actually holds and it is *not* the case that (predicate `np`) there is a sequence `S1` that `R` cannot distinguish from `S` and in which $p(\vec{x})$ does not hold. A property $p(\vec{x})$ is *common* knowledge if $p(\vec{x})$ holds in all sequences that are in the transitive closure of the indistinguishability relation across all players:

```
knows(p(x⃗),S,T) :- p(x⃗,S,T), not np(x⃗,S,T).
np(x⃗,S,T) :- indtrans(S,S1,T), not p(x⃗,S1,T).
```

Solving Epistemic Puzzles with GDL-III. Epistemic puzzles are characterised by multiple agents starting off with imperfect, and in many cases asymmetric, knowledge. They draw further conclusions by logical reasoning about each other's (lack of) knowledge in the course of a short sequence of actions, which often merely consists in repeated public announcements of an agent's ignorance until everyone has perfect knowledge. An example is the following puzzle, which recently acquired public fame [1].

Example 2 *Albert and Bernard want to know Cheryl's birthday. She draws a list with possible dates and then tells them separately the correct month and day, respectively. A dialogue follows in which Albert first says that he doesn't know the birthday* and *that he knows that Bernard doesn't know either, then Bernard says that he now knows the date, and after that Albert announces that he does so too.*

Puzzles like this that centre around knowledge of the knowledge of other agents can be formalised using GDL-III in such a way that *every* legal playout corresponds to a solution and vice versa. These puzzles can then be automatically solved by a mere GDL-III legal reasoner like the ASP-based reasoning system from above.

REFERENCES

[1] K. Chang, 'A math problem from Singapore goes viral: When is Cheryl's birthday?', *The New York Times*, (14 Apr 2015).

[2] A. Cordón-Franco, H. van Ditmarsch, D. Duque, and F. Soler-Toscano, 'A colouring protocol for the generalized Russian cards problem', *Journal of Theoretical Computer Science*, **495**, 81–95, (2013).

[3] M. Genesereth, N. Love, and B. Pell, 'General game playing: Overview of the AAAI competition', *AI Magazine*, **26**(2), 62–72, (2005).

[4] M. Genesereth and M. Thielscher, *General Game Playing*, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool, 2014.

[5] S. Schiffel and M. Thielscher, 'Representing and reasoning about the rules of general games with imperfect information', *Journal of Artificial Intelligence Research*, **49**, 171–206, (2014).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1632

Impact of Automated Action Labeling in Classification of Human Actions in RGB-D Videos

David Jardim [1234] and **Luís Nunes** [234] and **Miguel Dias** [124]

Abstract.
For many applications it is important to be able to detect what a human is currently doing. This ability is useful for applications such as surveillance, human computer interfaces, games and healthcare. In order to recognize a human action, the typical approach is to use manually labeled data to perform supervised training. This paper aims to compare the performance of several supervised classifiers trained with manually labeled data versus the same classifiers trained with data automatically labeled. In this paper we propose a framework capable of recognizing human actions using supervised classifiers trained with automatically labeled data in RGB-D videos.

1 Introduction

The goal of human activity recognition is to successfully classify an action performed by an individual or a group of people from a video observation. Although significant progress has been made, HAR remains a challenging area with several problems to solve. Manual analysis of video is labour intensive, fatiguing, and error prone. Solving the problem of recognizing human activities from video can lead to improvements in several application fields like surveillance systems, human computer interfaces, sports video analysis, digital shopping assistants, video retrieval, gaming and healthcare [8, 3, 7, 10, 5]. We are interested in recognizing high-level human activities and interactions between humans and objects, ideally our recognition algorithm should be robust to changes in relative distance between the body and the sensor (Kinect), skeleton orientation, and speed of an action. In order to abstract ourselves from computer vision problems the Kinect sensor will be used to extract 3D skeleton data. Usually manually labeled data is used to perform some kind of training of classifiers that will then recognize the human activities. What if this labeling could be achieved automatically? This paper shows that automating the data labeling process for the type of actions studied results in a minor loss in accuracy.

2 Proposed Pipeline

According to [1] human activity can be categorized into four different levels: gestures, actions, interactions and group activities. This paper will focus on the actions and interactions category. We recorded a dataset containing sequences of actions performed by a 12 different subjects. We used Kinect to record the dataset with sequences of combat movements composed of 8 different actions: *right-punch; left-punch; elbow-strike; back-fist; right-front-kick; left-front-kick; right-side-kick; left-side-kick*. Using combinations of those 8 actions we created 6 distinct sequences (each sequence contains 5 actions). Of the 12 subjects recorded, each subject performed 6 different sequences. A total of 72 sequences, 360 actions was recorded. The dataset [5] is available for public usage. A modular framework was built with several task-oriented modules organized in a work-flow (Fig.1).

2.1 Temporal Segmentation and Action Labeling

In our previous work [4] we proved that given a sequence of contiguous actions it is possible to automatically divide the sequence into what we called temporal segments that correspond to individual actions that would latter be automatically labeled by a clustering algorithm.

2.2 Action classification

At this point, using our temporal segmentation approach and an off-the-shelve algorithm to perform action clustering, we were able to automatically assign a label to an action. In order to verify the accuracy of our automatically labeled training set, the original dataset was manually labeled to be used as our ground truth. Kinect is able to track 20 joints of a subject's skeleton. Of those 20 joints, only four were selected to extract features (wrist-right; wrist-left; ankle-right; ankle-left). The 3D coordinates are with respect to a frame of reference centered at Kinect. Frames from Kinect are converted into feature vectors which are invariant to relative position and orientation of the body and will be used to train the classifiers.

3 Experiments

We experimented with the following classifiers: Multilayer Perceptron (MLP) as in [6]; Support Vector Machines (SVM) using pairwise classification [9] and Random Forests (RF) which are a combination of tree predictors [2]. Eight binary supervised classifiers were trained using manually labeled data for recognizing the eight aggressive actions contained in our dataset. Binary classifiers produced the best results in [7] using SVM classifiers.

Comparing Table 1 with Table 2 shows the difference of using a manually labeled training set versus a training set labeled by our automatic labeling pipeline. As expected the usage of automatic labeling has affected the accuracy of the classifiers. This can be explained by the error that our automatic labeling method introduces. Finally,

[1] Microsoft Language Development Center, Lisbon, Portugal, email: t-dajard, midias@microsoft.com
[2] Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal, email: luis.nunes@iscte.pt
[3] Instituto de Telecomunicações, Lisbon, Portugal
[4] ISTAR-IUL, Lisbon, Portugal

[5] https://github.com/DavidJardim/precog_dataset_16

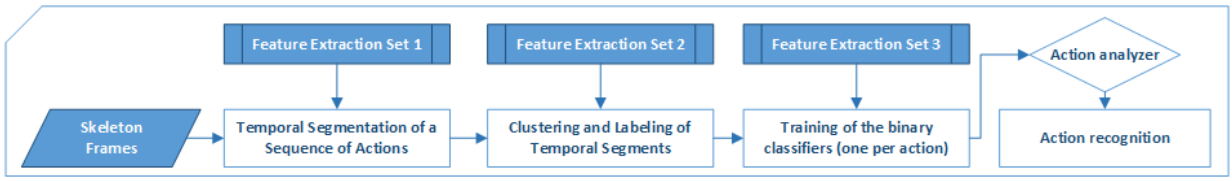


Figure 1. Modular framework for action recognition

Table 1. Classification accuracy (%) of the binary classifiers using manually labeled data and corresponding standard deviation between trials

Action	MLP	SVM	RF
right-punch	94,24 ±0,44%	91,52 ±0,17%	90,08 ±0,37%
left-punch	89,09 ±0,44%	92,50 ±0,26%	92,21 ±0,37%
front-right-kick	88,14 ±0,96%	87,95 ±0,21%	93,20 ±0,53%
front-left-kick	89,96 ±0,79%	90,42 ±0,28%	91,97 ±0,48%
side-right-kick	91,22 ±0,16%	91,92 ±0,07%	94,53 ±0,57%
side-left-kick	83,62 ±0,97%	84,76 ±0,23%	91,74 ±0,51%
backfist	92,55 ±0,32%	92,77 ±0,00%	93,58 ±0,46%
elbow-strike	95,02 ±0,28%	96,66 ±0,00%	96,66 ±0,00%

Table 2. Classification accuracy (%) of the binary classifiers using automatic labeled data and corresponding standard deviation between trials

Action	MLP	SVM	RF
right-punch	83,82 ±0,81%	88,29 ±0,16%	89,40 ±0,48%
left-punch	82,43 ±1,31%	90,20 ±0,00%	90,84 ±0,33%
front-right-kick	81,22 ±0,74%	90,75 ±0,07%	90,00 ±0,49%
front-left-kick	89,99 ±0,76%	87,91 ±0,13%	90,99 ±0,25%
side-right-kick	82,80 ±1,18%	87,88 ±0,07%	89,57 ±0,57%
side-left-kick	84,99 ±0,86%	90,28 ±0,05%	90,56 ±0,68%
backfist	83,09 ±1,44%	87,60 ±0,00%	90,05 ±0,41%
elbow-strike	95,90 ±0,31%	96,83 ±0,00%	96,83 ±0,00%

in Table 3 we calculate the difference in performance for each classifier accuracy using manually labeled data and automatically labeled data.

Table 3. Difference in performance (%) between the two approaches (manual vs automatic) for each binary classifier per action

Action	MLP	SVM	RF
right-punch	-10,42 %	-3,23 %	-0,68 %
left-punch	-6,66 %	-2,30 %	-1,37 %
front-right-kick	-6,92 %	2,80 %	-3,2 %
front-left-kick	0,03 %	-2,51 %	-0,98 %
side-right-kick	-8,42 %	-4,04 %	-4,96 %
side-left-kick	1,37 %	5,52 %	-1,18 %
backfist	-9,46 %	-5,17 %	-3,53 %
elbow-strike	0,88 %	0,17 %	0,17 %
average	-4,95 %	-1,09 %	-1,97 %

4 Conclusion

In summary, our results proved that, for a dataset of simple combat actions, obtained with a standard Kinect camera with no special acquisition conditions, a temporal segmentation and clustering algorithm can be used to label identical actions performed by different users. Also, we have established that this labeling can be used to train supervised classifiers that will be capable of identifying specific actions in a RGB-D video feed without relying on any human resources, with a minor loss of precision relative to training with human labeled data. Although this research area has grown dramatically in the past years, we identified a potentially under explored sub-area: action prediction. In future work, we would like to expand the current vision-based activity analysis to a level where it is possible, at some points, to predict a future action executed by a subject in the context of a sequence of actions.

ACKNOWLEDGEMENTS

This research was sponsored by a joint scholarship between Fundação para a Ciência e a Tecnologia (FCT) and Microsoft Portugal under the grant SFRH/BDE/52125/2013. The project was hosted by Microsoft Language Development Center (MLDC) in Lisbon, Portugal.

REFERENCES

[1] J K Aggarwal and M S Ryoo, 'Human Activity Analysis: A Review', *ACM Computing Surveys*, **43**(3), 1–43, (2011).
[2] Leo Breiman, 'Random forests', *Machine Learning*, **45**(1), 5–32, (2001).
[3] Stephen S Intille and Aaron F Bobick, 'A Framework for Recognizing Multi-agent Action from Visual Evidence', in *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference*, number 489, pp. 518–525. AAAI Press, (1999).
[4] David Jardim, Luís Nunes, and Miguel Sales Dias, 'Automatic human activity segmentation and labeling in rgb-d videos', in *Intelligent Decision Technologies: KES-IDT 2016*, p. in press. Springer International Publishing, (2016).
[5] Christoph G. Keller, Thao Dang, Hans Fritz, Armin Joos, Clemens Rabe, and Dariu M. Gavrila, 'Active pedestrian safety by automatic braking and evasive steering', *IEEE Transactions on Intelligent Transportation Systems*, **12**(4), 1292–1304, (2011).
[6] Miroslav Kubat, 'Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7', in *The Knowledge Engineering Review*, volume 13, pp. 409–412. Cambridge Univ Press, (1999).
[7] Shahriar Nirjon, Chris Greenwood, Carlos Torres, Stefanie Zhou, John a. Stankovic, Hee Jung Yoon, Ho Kyeong Ra, Can Basaran, Taejoon Park, and Sang H. Son, 'Kintense: A robust, accurate, real-time and evolving system for detecting aggressive actions from streaming 3D skeleton data', in *International Conference on Pervasive Computing and Communications*, pp. 2–10. IEEE Press, (2014).
[8] W Niu, J Long, D Han, and Y F Wang, 'Human activity detection and recognition for video surveillance', in *International Conference on Multimedia and Expo*, pp. 719–722. IEEE Press, (2004).
[9] John C. Platt, 'Fast Training of Support Vector Machines Using Sequential Minimal Optimization', in *Advances in Kernel Methods - Support Vector Learning*, pp. 185 – 208. MIT Press, (1998).
[10] Mirela Popa, Alper Kemal Koc, Leon J M Rothkrantz, Caifeng Shan, and Pascal Wiggers, 'Kinect sensing of shopping related actions', in *Communications in Computer and Information Science*, volume 277 CCIS, pp. 91–100. Springer, (2012).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1634

Transductive Learning for the Identification of Word Sense Temporal Orientation

Mohammed Hasanuzzaman [1] and **Gaël Dias** [2] and **Stéphane Ferrari** [3]

Abstract.

The ability to capture the time information conveyed in natural language is essential to many natural language processing applications such as information retrieval, question answering, automatic summarization, targeted marketing, loan repayment forecasting, and understanding economic patterns. In this paper, we propose a graph-based semi-supervised classification strategy that makes use of WordNet definitions or 'glosses', its conceptual-semantic and lexical relations to supplement WordNet entries with information on the temporality of its word senses. Intrinsic evaluation results show that the proposed approach outperforms prior semi-supervised, non-graph classification approaches to the temporality recognition of word senses, and confirm the soundness of the proposed approach.

1 Introduction

There is considerable academic and commercial interest in processing time information in text, where that information is expressed either explicitly, or implicitly, or connotatively. Recognizing such information and exploiting it for Natural Language Processing (NLP) and Information Retrieval (IR) tasks are important features th and at can significantly improve the functionality of NLP/IR applications [5, 1].

Most text applications have been relying on rule-based time taggers such as HeidelTime [9] or SUTime [2] to identify and normalize time mentions in texts. Although interesting levels of performance have been seen, their coverage is limited to the finite number of rules they implement. Such systems would certainly benefit from the existence of a temporal resource enumerating a large set of possible time variants.

However, discovering the temporal orientation of words is a challenging issue even for humans if they intend to formalize them in a knowledge-base, despite the fact that they manage temporal information very naturally and efficiently during their everyday life. There are several explanations for this difficulty: (i) temporal connotations can be conveyed via a wide range of different mechanisms including grammar, aspect, and lexical semantic knowledge [8]. These properties need to be correctly identified, interpreted, and combined to derive the appropriate temporal orientation, (ii) another challenge arises from the fact that time can be expressed in countless manners and is not always expressed explicitly, rather implicitly and require interpretations or inferences derived from world knowledge, (iii) conventional knowledge acquisition approaches are usually driven by humans, which means that they are labor-intensive, time-consuming and troublesome, (iv) the data sparsity problem is aggravated by the fact that dictionary definitions or 'glosses' are very short, typically contains few words.

Whereas most of the prior computational linguistics and text mining temporal studies have focused on temporal expressions and events, there has been a lack of work looking at the temporal orientation of word senses/synsets. In this paper, we put forward a semi-supervised graph-based classification paradigm build on an optimization theory namely the max-flow min-cut theorem [7]. In particular, we propose minimum cut in a connected graph to time-tag each synset of WordNet [6] to one of the two dimensions: *temporal* and *atemporal*. Our methodology was evaluated intrinsically and outperformed prior approaches to the temporality recognition of word senses.

2 Methodology

The s-t mincut algorithm is based on finding minimum cuts in a graph, and uses pairwise relationships among examples in order to learn from both labeled and unlabeled data. In particular, it outputs a classification corresponding to partitioning a graph in a way that minimizes the number of similar pairs of examples that are given different labels.

The formulation of our mincut strategy for temporal classification of synsets involves the following steps.

- **Step I.** We define two vertices s (source) and t (sink), which correspond to the *temporal* and *atemporal* categories, respectively. Vertices s and t are *classification vertices*, and all other vertices (labeled and unlabeled) are *example vertices*.
- **Step II.** The labeled examples are connected to the classification vertices they belong to via edges with high constant non-negative weight. The unlabeled examples are connected to the classification vertices via edges weighted with non-negative scores that indicate the degree of belonging to both the *temporal* and *atemporal* categories. Weights (i.e. individual scores) are calculated based on a supervised classifier learned from labeled examples. For the classification task, each synset from the labeled dataset is represented by its gloss encoded as a vector of word unigrams weighted by their frequency. Then, a two-class SVM classifier is built from the Weka platform.[4] and the SVM membership scores are directly mapped to edge weights.
- **Step III.** For all pairs of example vertices, for which there exists a listed semantic relation in WordNet, an edge is created. This one

[1] ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland
[2] Normandie University, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France
[3] Normandie University, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

[4] `http://www.cs.waikato.ac.nz/ml/weka/`

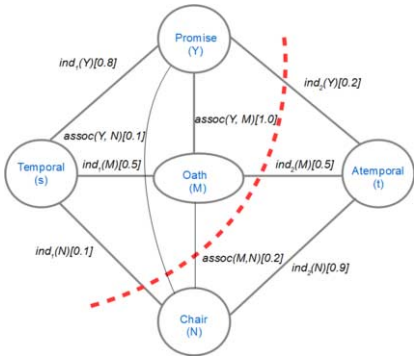


Figure 1: Example graph.

receives a non-negative score that indicates the degree of semantic relationship between both vertices (i.e. association score).

- **Step IV.** The max-flow theorem [7] is applied over the built graph to find the minimum s-t cut.[5]

2.1 Example

Figure 1 illustrates a classification problem with a set of three words {promise (Y), oath (M), chair (N)} belonging either to the *temporal* class (C_1) or the *atemporal* class (C_2) with the s-t mincut algorithm. Square brackets enclose edge weights (here probability scores). Table 1 presents all possible cuts and respective costs. The minimum cut (indicated by the dashed red line) places {promise, oath } in C_1 and {chair} in C_2.

C_1	C_2	$\sum_{x\in C_1} ind_2(x) + \sum_{x\in C_2} ind_1(x)$	$\sum_{x_i\in C_1, x_k\in C_2} assoc(x_i, x_k)$	$cost(S,T)$
Y,M	N	0.2+0.5+0.1	0.1 + 0.2	1.1
none	Y,M,N	0.8+0.5+0.1	0	1.4
Y,M,N	none	0.2+0.5+0.9	0	1.6
Y	M,N	0.2+0.5+0.1	1.0+0.1	1.9
N	Y,M	0.8+0.5+0.9	0.1+0.2	2.5
M	Y,N	0.8+0.5+0.1	1.0+0.2	2.6
Y,N	M	0.2+0.5+0.9	1.0+0.2	2.8
M,N	Y	0.8+0.5+0.9	1.0+0.1	3.3

Table 1: Possible cuts for the illustrative case of Figure 1.

3 Experiments and Evaluation

We used a list that consists of 632 *temporal* synsets and an equal number of *atemporal* synsets provided by Dias et al. [3] as labeled data for our experiments.

Using our formulation in Section 2, we construct a connected graph by importing 1264 training set (632 *temporal* and 632 *atemporal* synsets), and 116394 unlabeled synsets[6]. We construct edge weights to *classification vertices, s (temporal)* and *t (atemporal)* by using the SVM classifier discussed above. WordNet relations for links between *example vertices* are weighted by non-negative constant value of 1.

In order to compare our approach to prior works, we adopted a similar evaluation strategy as proposed in Dias et al. (2014) and Hasanuzzaman et al. [4]. To assess human judgment regarding the temporal parts, inter-rater agreement with multiple raters (i.e. 3 human annotators with the 4th annotator being the classifier) was performed over a set of 398 randomly selected synsets. The free-marginal multirater kappa and the fixed-marginal multirater kappa values are reported in Table 2 and assess moderate agreement for previous versions of TempoWordNet (TWnL, TWnP and TWnH), while good agreement is obtained for the resources constructed by mincut. These agreement values provide a first and promising estimate of the improvement over the previous versions of TempoWordNet. We plan to confirm that in the future by comparing the systems to a true reference instead of observing the agreement between the systems and a multi-reference as we currently do.

Metric	TWnL	TWnP	TWnH	Mincut
Fixed-marginal κ	0.51	0.46	0.54	0.71
Free-marginal κ	0.52	0.55	0.59	0.82

Table 2: Inter-annotator agreement.

4 Conclusions

In this paper, we proposed a semi-supervised minimum cut framework to address the problem of associating word senses with their underlying temporal dimensions. Comparative evaluations are performed to measure the quality of the resource. The results confirm the soundness of the proposed approach.

As part of future work, we plan to investigate the effect of other graph construction methods, such as different weights to different WordNet relations to reflect the degree to which they are temporality preserving instead of using same for all. Another direction of future work is to fine tune the temporal part into *past, present,* and *future*. We would also like to explore the impact of the resource on more applied temporal information extraction task such as temporal relation annotation of TempEval-3 challenge.

REFERENCES

[1] Ricardo Campos, Gaël Dias, Alípio M. Jorge, and Adam Jatowt, 'Survey of temporal information retrieval and related applications', *ACM Computing Survey*, **47**(2), 15:1–15:41, (2014).

[2] Angel X Chang and Christopher D Manning, 'Sutime: A library for recognizing and normalizing time expressions.', in *LREC*, pp. 3735–3740, (2012).

[3] Gaël Harry Dias, Mohammed Hasanuzzaman, Stéphane Ferrari, and Yann Mathet, 'Tempowordnet for sentence time tagging', in *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, WWW Companion '14, pp. 833–838, Republic and Canton of Geneva, Switzerland, (2014). International World Wide Web Conferences Steering Committee.

[4] Mohammed Hasanuzzaman, Gaël Dias, Stéphane Ferrari, and Yann Mathet, 'Propagation strategies for building temporal ontologies', in *14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 6–11, (2014).

[5] Inderjeet Mani, James Pustejovsky, and Robert Gaizauskas, *The language of time: a reader*, volume 126, Oxford University Press, 2005.

[6] Georges A. Miller, 'Wordnet: a lexical database for english', *Communications of the ACM*, **38**(11), 39–41, (1995).

[7] Christos H Papadimitriou and Kenneth Steiglitz, *Combinatorial optimization: algorithms and complexity*, Courier Corporation, 1998.

[8] James Pustejovsky, 'Time and the semantic web', in *Temporal Representation and Reasoning, 2005. TIME 2005. 12th International Symposium on*, pp. 5–8. IEEE, (2005).

[9] Jannik Strötgen and Michael Gertz, 'Heideltime: High quality rule-based extraction and normalization of temporal expressions', in *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pp. 321–324, Stroudsburg, PA, USA, (2010). Association for Computational Linguistics.

[5] Max-flow algorithms show polynomial asymptotic running times and near-linear running times in practice.

[6] All synset of WordNet− labeled data

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1636

Towards a Framework for Detecting Opportunism in Multi-Agent Systems

Jieting Luo[1] and **John-Jules Meyer**[2] and **Max Knobbout**[3]

1 Introduction

Consider a common social scenario. A seller sells a cup to a buyer and it is known by the seller beforehand that the cup is actually broken. The buyer buys the cup without knowing it is broken. Since the buyer's value gets demoted, the behavior performed by the seller is usually forbidden by social norms. Such a social behavior intentionally performed by the seller is first named opportunistic behavior (or opportunism) by economist Williamson [6]. It is a typical social behavior that is motivated by self-interest and takes advantage of knowledge asymmetry about the behavior to achieve own gains, regardless of the principles [3]. This definition implies that, given a social context, opportunistic behavior results in promoting agents' own value while demoting social value. Therefore, it is prohibited by norms in most societies. In the context of multi-agent systems, we constrain such a selfish behavior through setting enforcement norms, in the sense that agents receive a corresponding sanction when they violate the norm. On the one hand, it is important to detect it, as it has undesirable results for the participating agents. On the other hand, as opportunism is always in the form of cheating, deception and betrayal, meaning that the system does not know what the agent performs or even the motivation behind it (for example, in a distributed system), monitors can only observe agents' opportunistic behavior indirectly. Therefore, there has to be a monitoring mechanism that can detect the performance of opportunistic behavior in the system. This paper introduces the idea of using a logical framework based on the specification of actions to verify whether agents in the system perform opportunistic behavior.

2 Framework

Since monitors cannot observe the performance of opportunism directly, the action can only be identified through the information about the context where the action can be performed and the property change in the system, which is called *action specification* [5] or *action description* [2]. Usually an action can be specified through its precondition and its effect (postcondition): the precondition specifies the scenario where the action can be performed whereas the postcondition specifies the scenario resulting from performing the action. For example, the action, dropping a glass to the ground, can be specified as holding a glass as its precondition and the glass getting broken as its effect. Therefore, we assume that every action has a pair of the form $\langle \psi, \psi' \rangle$, where ψ is the precondition of action a and ψ' is the effect of performing action a in the context of ψ.

[1] Utrecht University, the Netherlands, email: J.Luo@uu.nl
[2] Utrecht University, the Netherlands, email: J.J.C.Meyer@uu.nl
[3] Delft University of Technology, the Netherlands, email: M.Knobbout@tudelft.nl

The models that we use are transition systems, which consist of agents, states, actions, transitions between states by actions and a valuation function mapping a state to a set of true propositions. When an action is performed in a certain state s, the system might progress to a different state in which different propositions might hold. Such a system is a very generic way of modelling a multi-agent system (see for example [7]). Since we have already introduced the notion of action specification $\langle \psi, \psi' \rangle$, all the possible state transitions are defined such that they go from a ψ-state to a ψ'-state. We also extend the standard framework with a monitor relation $\mathcal{M}$, which represents the indistinguishability of a monitor over different states. Sometimes we also use $s\langle a \rangle$ to denote the state resulting from the performance of action a in state s. The logical language we use in this paper is propositional logic extended with action modality for reasoning about dynamic worlds. The syntax and the semantics are defined in the same way as propositional logic except the formula $\langle a \rangle \varphi$. This formula holds if and only if φ is true after action a is performed in state s.

Similar to [1], we simply consider a norm as a subset of all the state transitions that is decided by designers of the system. In other words, if a norm is denoted as η, a state transition is an η-violation if and only if it is in the set η. We define norms of the form $\eta(\varphi, a)$, interpreted as it is forbidden to perform action a in a φ-state. This is the most common form in which the action and the context where the action is forbidden are explicitly represented, regardless of the effect that the action brings about. For example, it is forbidden to smoke in a non-smoking area. Of course, it is only a choice in this paper and more forms of norms are described and constructed based on our logical framework in the full paper [4].

3 Defining Opportunism

Before we propose our monitoring approach for opportunism, we should formally define opportunism from the perspective of the system so that the system knows what to detect for monitoring opportunism. In our previous paper [3], we emphasized opportunistic behavior is performed by intent rather than by accident. However, monitors cannot read agents' mental states, so for monitoring we assume that agents violate the norms always by intention from a pragmatic perspective. For example, we always assume that speeding is performed with intention. In this paper we remove all the references to the mental states from the formal definition of opportunism in our previous paper [3], assuming that the system can tell agents' value promotion/demotion causing by an action. In a sentence, from the perspective of the system, opportunistic behavior performed by an agent in a social context can be simply defined as a behavior that causes norm violations and promotes his own value. In this short paper, opportunism is denoted as $Opportunism(\eta, a)$, interpreted as

action a is opportunistic behavior with respect to norm η. Opportunistic behavior results in promoting agents' own value, which can be interpreted as that opportunistic agents prefer the state that results from opportunistic behavior rather than the initial state. For having preferences over different states, we argue that agents always evaluate the truth value of specific propositions in those states based on their value systems. Based on this understanding, we define a function of the form $EvalRef(V_i, s, s')$ mapping a value system and two states to a proposition an agent refers to for specifying his preference over two states.

4 Monitoring Opportunism

In this paper, a monitor is considered as an external observer to evaluate a state transition with respect to a given norm. However, a monitor can only verify state properties instead of observing the performance of actions directly. Our approach to solve this problem is to check how things change in a given state transition and reason about the action taking place in between.

We first define a state monitor $m_{state}(\varphi)$, which can evaluate the validity of a given property in a given state. Because a monitor can be seen as an external observer that can observe agents' activities, we can define state monitors in this paper in a similar way as we define knowledge in epistemic logic, and correspondingly adopt S5 properties.

Definition 1 (State Monitors)**.** *Given a monitoring transition system $\mathcal{I}$, a value system set V, and a propositional formula φ, a state monitor m_{state} for φ over $\mathcal{I}$ is defined as follows: $\mathcal{I}, s \vDash m_{state}(\varphi)$ iff for all s' $s\mathcal{M}s'$ implies $\mathcal{I}, V, s' \vDash \varphi$.*

$m(\varphi)$ is read as φ "is detected" to be φ. As the $\mathcal{M}$-relation is reflexive, we have the validity $\vDash m_{state}(\varphi) \rightarrow \varphi$, meaning that what the state monitor detects is always considered to be true.

State monitors are the basic units in our monitoring mechanism. We can combine state monitors to check how things change in a given state transition and evaluate it with respect to a given set of norms. As we defined in Section 3, opportunistic behavior performed by an agent is a behavior that causes norm violations and promotes his own value. In other words, opportunism is monitored with respect to a norm and a value system of an agent. Based on this definition, we design a monitoring opportunism approach $m_{opp}((\varphi, a), \langle\psi, \psi'\rangle, a')$ with respect to norm $\eta(\varphi, a)$.

Definition 2. *Given a monitoring transition system $\mathcal{I}$, a value system set V, a norm $\eta(\varphi, a)$, and a pair $\langle\psi, \psi'\rangle$ of action a, in order to check action a' performed by agent i in state s is opportunistic behavior, we can combine monitors as follows:*

$$\mathcal{I}, s \vDash m_{opp}((\varphi, a), \langle\psi, \psi'\rangle, a') := m_{state}(\varphi \wedge \psi) \wedge \langle a' \rangle m_{state}(\psi')$$

where $\varphi \wedge \psi$ implies $\neg p$ and ψ' implies p, and $p = EvalRef(V_i, s, s\langle a'\rangle)$.

In order to check whether action a' is opportunistic behavior (violates norm $\eta(\varphi, a)$ and promotes own value), we verify if action a' is performed in φ-state. Besides, we check if action a' is the action that the norm explicitly states. Since the monitors cannot observe the performance of action a' directly, we only can identify action a' to be possibly action a with pair $\langle\psi, \psi'\rangle$ by checking if formulas ψ and ψ' are successively satisfied in the state transition by action a'. With this approach we have a candidate set of states for state s and a candidate set of states for state $s\langle a'\rangle$ and any two states from them satisfy the resulting property of function $EvalRef$, which means that given the partial information the execution of action a' in state s brings about p thus promoting agent i's value.

However, since the monitors can only verify state properties instead of observing the performance of the action directly, we cannot guarantee that an action that is detected to be opportunistic was indeed opportunistic, because there might exist more than one action that can be represented by pair $\langle\psi, \psi'\rangle$. That is, formula $\mathcal{I}, V, s \vDash m_{opp}((\varphi, a), \langle\psi, \psi'\rangle, a') \rightarrow Opportunism((\varphi, a), a')$ might not hold. Given this problem, we need to investigate in which case or with what requirement the action that is detected by the opportunism monitor is indeed opportunistic behavior. In order to guarantee that action a' that is detected to be opportunistic was indeed opportunistic, we should make sure that, within the actions available in φ-state, there exists only one action that can be represented with pair $\langle\psi, \psi'\rangle$. With this condition, action a' is indeed action a as norm $\eta(\varphi, a)$ indicates, so we can guarantee that action a' that is detected to be opportunistic was indeed opportunistic. This approach implies: in order to better monitor opportunistic behavior, we should appropriately find an action pair $\langle\psi, \psi'\rangle$ such that the possible actions in between can be strongly restricted and minimized. Assuming that the action pair we use is $\langle\top, \top\rangle$, the possibility that the opportunism monitor makes an error is extremely high, because every action that is available in a φ-state will be detected to be opportunistic behavior. However, sometimes it is difficult to find a unique pair $\langle\psi, \psi'\rangle$ for the action we monitor, especially when we cannot limit the available actions with the given context (a φ-state). So it is important to have more information not only about the action, but also about the context where the action performed and the system. All these issues will be elaborated and discussed in the full paper [4].

5 CONCLUSION

In this paper, we introduce the idea of verifying opportunism, which is a behavior that causes norm violation and promotes agents' own value. Our logical framework is developed based on the specification of actions. In particular, we investigated how to evaluate agents' actions to be opportunistic with respect to norms when those actions cannot be observed directly. Future work can investigate more formal properties to improve the effectiveness of our monitoring approach for opportunism: whenever an action is detected to be opportunistic, it was indeed opportunistic; whenever an action was opportunistic, it is indeed detected. Monitoring costs can be another interesting topic to be studied based on our monitoring approach.

REFERENCES

[1] Nils Bulling, Mehdi Dastani, and Max Knobbout, 'Monitoring norm violations in multi-agent systems', in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 491–498, (2013).

[2] J Fiadeiro and T Maibaum, 'Temporal reasoning over deontic specifications', *Journal of Logic and Computation*, **1**(3), 357–395, (1991).

[3] Jieting Luo and John-Jules Meyer, 'A formal account of opportunism based on the situation calculus', *AI & SOCIETY*, 1–16, (2016).

[4] Jieting Luo, John-Jules Meyer, and Max Knobbout, 'Monitoring opportunism in multi-agent systems', in *Proceedings of the Coordination, Organizations, Institutions and Norms workshop @ECAI2016*, p. to appear. Springer.

[5] Raymond Reiter, *Knowledge in action: logical foundations for specifying and implementing dynamical systems*, MIT press, 2001.

[6] Oliver E Williamson, 'Markets and hierarchies: analysis and antitrust implications: a study in the economics of internal organization', (1975).

[7] Michael Wooldridge, *An introduction to multiagent systems*, John Wiley & Sons, 2009.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1638

Learning a Bayesian Network Classifier by Jointly Maximizing Accuracy and Information

Dan Halbersberg and **Boaz Lerner** [1]

Abstract. Although recent studies have shown that a Bayesian network classifier (BNC) that maximizes the classification accuracy (i.e., minimizes the 0/1 loss function) is a powerful tool in knowledge representation and classification, this classifier focuses on the majority class, is usually uninformative about the distribution of misclassifications, and is insensitive to error severity (making no distinction between misclassification types). We propose to learn a BNC using an information measure (IM) that jointly maximizes classification and information, and evaluate this measure using various databases. We show that an IM-based BNC is superior to BNCs learned using other measures, especially for ordinal classification and imbalanced problems, and does not fall behind state-of-the-art algorithms with respect to accuracy and amount of information provided.

1 Introduction and Background

Prediction and identification of key factors in imbalance ordinal problems are difficult for several reasons. First, classifiers that maximize accuracy (ACC) during learning do not account for error distribution and, thus, are not informative enough about the classification result. On the other hand, classifiers that account for error distribution usually are not accurate enough. Second, for imbalanced data, classifiers usually predict all minority samples as the majority class. Tackling imbalance by down-sampling the majority class, up-sampling the minority class, or applying different costs to different misclassifications provide an optimistic ACC estimate, and thus are not recommended [4]. Third, 0/1 loss function classifiers are not optimized to tackle different error severities differently; for instance, they consider misclassification of fatal accidents as severe, similar to misclassification of fatal accidents as minor. Fourth, classifiers (e.g., SVM, NN) usually excel in prediction but not in knowledge representation, which is a main goal of this study.

The Bayesian network classifier (BNC) excels in knowledge representation, which makes it ideal to identify key factors as required, but like other classifiers, it suffers from the first three problems. It has been claimed [2] and shown [3] that to achieve high ACC, a BNC should maximize a (discriminative) score which is specific to classification, and not a general inference score based on likelihood. Therefore, to tackle the above concerns, we first consider replacing ACC with four existing scores, each of which accounts for the entire confusion matrix (CM) and not just its diagonal (ACC): 1) *Mutual information* (**MI**) that is defined between two M-dimensional vectors, X and Y, holding predictions and true values for M possible classes, respectively [1]; 2) *Mean absolute error* (**MAE**) that is the average deviation between X and Y; 3) *Matthew correlation coefficient* (**MCC**) that is the correlation between the true (Y) and predicted (X) class matrices [1]; and 4) *Confusion entropy* (**CEN**) [5] that exploits the distribution of misclassifications of a class as any of the $M-1$ other classes.

Second, since none of the above measures accounts for all concerns, we propose a novel information measure (IM) that uses MI to evaluate the error distribution and a factor we introduce to measure error severity (ES) between predictions and true values,

$$IM_\alpha = \sum_x^X \sum_y^Y P(x,y)\left(-log\left(\frac{\alpha P(x,y)}{P(x)P(y)}\right) + log(1+\alpha|x-y|)\right). \quad (1)$$

When $\alpha = 1$, $IM_\alpha = IM$, and ES measures a "classification distance" $|x-y|$, which is transformed using the weighted by the joint distribution logarithm to MI "units". Both $P(X,Y)$ and $|x-y|$ are measured using the CM. When predictions are uniformly distributed (maximum entropy), MI contributes the most to IM. When there is no error between X and Y (off-diagonal elements are 0), the only contribution to IM is from MI, and if, in addition, the classes are balanced, IM is minimized to $-log(M)$. When the error severity is maximal, ES contributes $log(1+M-1) = log(M)$; hence, MI and ES contradict each other, and IM is balanced in $[-log(M), log(M)]$. We seek a classifier whose prediction and true value distributions correspond to each other, while its errors are the least severe. When $\alpha > 1$, α is a user or data-defined constant that balances ACC, information, and ES, IM_α is a generalization of IM. Then, it is easy to show that $IM_\alpha = IM - log(\alpha) \times ACC$, i.e., IM_α is monotonic with ACC, and when $\alpha \to \infty$, $log(\alpha)$ dominates IM_α.

To demonstrate the value of these two novel measures in comparison to the existing measures, we conducted experiments with synthetic CMs and summarize the most important properties of the measures regarding whether they: 1) balance ACC and information, 2) prefer balanced class distribution, 3) are sensitive to the error distribution, 4) tackle error severity, and 5) are sensitive to the number of classes. ACC and MCC do not meet any of the above properties; MI meets 1, 2, and 5; CEN meets 1 and 3; and MAE meets only 4. IM and IM_α are the only measures to meet them all.

2 Evaluation, Experiments, and Results

We compared the ability of each of the two proposed measures to augment learning of a state-of-the-art BNC called RMCV [3] with those of the existing measures (ACC, MI, MAE, MCC, and CEN), suggesting seven algorithms (classifiers) for evaluation. In each learning step of the RMCV algorithm, neighboring graphs (edge addition/deletion/reversal) are compared with the current graph as part of a greedy hill climbing, and learning proceeds if the measure computed on the validation set is improved by any of the neighboring

[1] Ben-Gurion University, Israel, emails: halbersb@bgu.ac.il; boaz@bgu.ac.il

Table 1. Mean|std ACC and normalized IM_α values of BNCs learned using seven measures for 23 artificial (ART) and 16 real-world (RL) DBs.

	ACC performance							IM_α performance						
	IM	IM_α	MI	CEN	MCC	MAE	ACC	IM	IM_α	MI	CEN	MCC	MAE	ACC
ART	75.4|9	**76.2|9**	74.0|10	*46.8|19*	75.0|10	74.4|11	75.4|9	68.5|9	**69.0|9**	67.6|8	*43.7|11*	68.1|9	67.5|10	68.1|9
RL	78.8|15	**79.0|15**	78.6|15	*68.6|22*	78.7|16	78.7|15	78.6|16	66.6|15	**66.9|15**	66.5|15	*54.9|21*	66.6|15	66.5|15	66.2|15

graphs, which then becomes the new current graph. When learning is completed, a CM is computed using the test set. This CM was evaluated using the seven measures. IM_α was normalized using min-max values in order to select best α (we heuristically examined values of α in $[2 : M, M^2, M^3]$ for M classes, and selected the α that maximized IM_α on a validation set independent of the training and test sets). We made the evaluation using artificial (ART) and real-world (RL) DBs (all classification problems were ordinal). In the results reported (Table 1 and Table 2), **bold** and *underlined italic* fonts indicate the best and worst algorithms, respectively.

First, we evaluated the algorithms using 23 ART DBs that were generated from a synthetic 20-node BN structure, in which the class variable Markov blanket includes 4 parents, 3 children, and 3 parents to common children. To test various scenarios and to simulate a broad range of problems, we changed the number of values of the class variable between 2 and 9, and sampled 2,000 samples in DBs 1–8; kept four values to the class variable and changed the number of samples between 500 and 3,000 in DBs 9–14; and sampled 2,000 samples and kept four classes, but changed their prior probabilities to represent different degrees of imbalance – from pure balance, through different levels of imbalance, to very high imbalance – in DBs 15–23. Each DB was re-sampled to create ten data permutations, and each permutation was divided into five folds (i.e., CV5). Table 1 (top) shows evaluation of the seven algorithms averaged over the 23 ART DBs according to the ACC (left) and IM_α (right). BNCs learned based on the IM_α measure (1) perform better than BNCs learned based on all other measures, regardless if the evaluation is based on the IM_α measure or ACC, which is interesting to see because classifiers trained to maximize the IM_α measure are not expected to also maximize ACC. BNCs based on ACC, MCC, or IM, are behind, and those learned based on the CEN measure provide the poorest performance. The reason that the CEN-based BNC performs so poorly is because empty graph initialization of the RMCV algorithm creates an initial CM that has entries only for one class, the majority class, leaving the total CEN measure relatively low (better). Thus, it stops at this local minimum and cannot proceed further.

To check if these differences are statistically significant, we performed a Friedman non-parametric test followed by a Nemenyi post-hoc test. Table 2 (top) shows the average ranks of the algorithms (lowest is best) according to ACC and IM_α, based on the Friedman test. The Nemenyi test shows (with a 0.05 confidence level) that all algorithms are significantly better than CEN with respect to the ACC and IM_α measures, and the IM_α-based BNC is significantly superior to those based on MI, MCC, MAE, and ACC. In addition, the BNC-IM_α has significantly better average ranks than the other algorithms have regardless of the measure that evaluates performance.

Table 2. Average ranks according to ACC and IM_α of BNCs learned using seven measures for the ART and RL DBs.

		IM	IM_α	MI	CEN	MCC	MAE	ACC
AR	ACC	3.2	**1.6**	4.7	*6.9*	3.9	3.8	3.8
	IM_α	3.0	**1.7**	4.3	*6.9*	4.0	4.0	4.1
RL	ACC	2.6	**2.1**	4.1	*6.0*	4.1	4.3	4.8
	IM_α	2.6	**2.1**	3.8	*6.1*	4.4	4.1	4.9

Next, we extended the evaluation of the measures using 14 UCI RL DBs (Australian, Autombp, Bostonhousing, Car, Cleve, Corral, Glass, Hepatitis, Machinecpu, Mofn, Mushroom, Shuttle, Stocksdomain, and Voting) and two of our own DBs: Amyotrophic lateral sclerosis (ALS) and Missed due date. The problems represented by these 16 DBs have 2–10 classes, 7–29 variables, 80–10,500 samples, and different degrees of class imbalance, posing a range of challenges to the classifiers. Again, ten random permutations were made to each DB, which were used over a CV5 experiment. Table 1 (bottom) shows the evaluation of the seven algorithms according to ACC (left) and IM_α (right) performances. Once again, IM_α-based BNCs preform better on average than BNCs learned based on all other measures. Table 2 (bottom) shows the average ranks according to the Friedman test according to ACC and IM_α. The results are consistent with those of the ART DBs, showing that IM_α is ranked first, followed by IM, MCC, and MAE. The ACC-based BNC was the second worst classifier, which re-emphasizes the motivation to replace it. According to Nemenyi post-hoc test (with 0.05 confidence level), BNC-IM_α is superior to BNCs-MI, CEN, MCC, and ACC with respect to ACC and to BNCs-CEN, MCC, and ACC with respect to IM_α. In addition, we expended our evaluation to other state-of-the-art algorithms suitable for ordinal classification, such as ordinal regression and ordinal DT with a cost matrix equivalent to that used by IM and IM_α. Friedman and Nemenyi tests found BNC-IM_α to be superior to these algorithms.

3 Summary and Discussion

Learning by only maximizing ACC and ignoring the error distribution and severity in class-imbalance problems results in accurate classification of only the major class at the expense of incorrect prediction of the minor one. We proposed an information measure, IM, and a weighted version of it, IM_α, to tackle these limitations in ordinal classification problems. We implemented them as a discriminative score in an algorithm for learning a BNC and demonstrated their advantage compared to other measures. IM and IM_α are specifically suited to any imbalance ordinal classification problem. If a problem is not ordinal, the contribution and impact of the ES term in the measures will vanish, and with no imbalance, the measures advantage over others may decrease. For problems with high imbalance and error that account differently for different classes, the advantage of these measures over other ACC and information measures is large.

REFERENCES

[1] P. Baldi, et al., 'Assessing the accuracy of prediction algorithms for classification: an overview', *Bioinformatics*, **16**(5), 412–424, (2000).

[2] N. Friedman, et al., 'Bayesian network classifiers', *Machine Learning*, **29**(2-3), 131–163, (1997).

[3] R. Kelner and B. Lerner, 'Learning Bayesian network classifiers by risk minimization', *International Journal of Approximate Reasoning*, **53**(2), 248–272, (2012).

[4] F. Provost, 'Machine learning from imbalanced data sets', in *Proceedings of the AAAI Workshop on Imbalanced Data Sets*, pp. 1–3, (2000).

[5] J. M. Wei, et al., 'A novel measure for evaluating classifiers', *Expert Systems with Applications*, **37**(5), 3799–3809, (2010).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1640

Transfer of Reinforcement Learning Negotiation Policies: From Bilateral to Multilateral Scenarios

Ioannis Efstathiou and **Oliver Lemon**[1]

Abstract. Trading and negotiation dialogue capabilities have been identified as important in a variety of AI application areas. In prior work, it was shown how Reinforcement Learning (RL) agents in bilateral negotiations can learn to use manipulation in dialogue to deceive adversaries in non-cooperative trading games. In this paper we show that such trained policies can also be used effectively for multilateral negotiations, and can even outperform those which are trained in these multilateral environments. Ultimately, it is shown that training in simple bilateral environments (e.g. a generic version of "Catan") may suffice for complex multilateral non-cooperative trading scenarios (e.g. the full version of Catan).

1 Introduction

Work on automated conversational systems has previously been focused on cooperative dialogue, where a dialogue system's core goal is to assist humans in their tasks such as finding a restaurant [13]. However, non-cooperative dialogues, where an agent may act to satisfy its own goals, are also of practical and theoretical interest [6]. It may be useful for a dialogue agent not to be fully cooperative when trying to gather information from a human, or when trying to persuade, or in the area of believable characters in video games and educational simulations [6]. Another area in which non-cooperative dialogue behaviour is desirable is in negotiation [12]. Recently, Reinforcement Learning (RL) methods have been applied in order to optimise *cooperative* dialogue management, where the decision of the next dialogue move to make in a conversation is in focus, in order to maximise an agent's overall long-term expected utility [13, 9, 10]. Those methodologies used RL with reward functions that give positive feedback to the agent only when it meets the user's goals. This work has shown that robust and efficient dialogue management strategies can be learned, but until [3], has only addressed cooperative dialogue. Lately it has been shown [5] that when given the ability to perform both cooperative and non-cooperative (manipulative) dialogue moves, a dialogue agent can learn to bluff and to lie during trading so as to win games more often, under various conditions such as risking penalties for being caught in deception – against a variety of adversaries [4]. Here we transfer those learned bilateral policies to more complex multilateral negotiations, and evaluate them.

2 Learning in Bilateral Negotiations

To learn trading policies in a controlled setting we initially [5] used a 2-player version of the non-cooperative 4-player board game "Catan". We call the 2 players the "adversary" and the "Reinforcement learning agent" (RLA). The goal of the RLA was to gather a particular number of resources via trading dialogue. Trade occurred through proposals that might lead to acceptance or rejection from the adversary. In an agent's proposal (turn) only one 'give 1-for-1' or 'give 1-for-2' trading proposal might occur, or nothing (41 actions in total), e.g. "I will give you a brick and I need two rocks". To overcome issues related to long training times and high memory demands, we have implemented a state encoding mechanism [5] that automatically compresses all of our numeric trading game states.

We first investigated the case of learning trading policies against adversaries which always accepted a trading proposal. The *goal-oriented RLA* did not use any manipulative actions and learned to reach its goal resources as soon as possible. In the case where the goal was to build a city it learned to win 96.8% of the time [5]. We then trained the *manipulative (dishonest) RLA* [5], which could ask for resources that it did not really need. It could also propose trades without checking whether the offered resource was available. The manipulated adversary [5] was implemented based on the intuition that a rational adversary will act so as to hinder other players in respect of their expressed preferences. The above trained policies of both of the agents are now evaluated in JSettlers [11].

3 Evaluating in Multilateral Negotiations

The experiments here are all conducted using JSettlers [11], a research environment developed in Java that captures the full multiplayer version of the game Catan, where there is trading and building. 10k games were played for each experiment. The players are:

The original STAC Robot (Bot) is based on the original expert rule-based agent of JSettlers [11] which is further modified to improve its winning performance. This agent (the Bot), which is the "benchmark" agent described in [7], uses complex heuristics to increase performance by following a dynamic building plan according to its current resource needs and the board's set-up.

Our trained RLA is in fact a Bot which has been modified to make offers based on our four learnt policies (for the development of city, road, development card, and settlement) in our version of the game "Catan" (Section 2). These policies were either the *goal-oriented* ones or the *manipulative (dishonest)* ones.

The Bayesian agent (Bayes) [8] is a Bot whose trading proposals are made based on the human corpus that was collected from Afantenos et al. [1]. The Bayesian agent was 65.7% accurate in reproducing the human moves.

The Manipulated Bot is a Bot which can be manipulated by our trained dishonest agent (i.e. the weights of the resources that they offer and ask for change according to the trained manipulative RL proposals). There are 3 types of manipulated Bots as we will see.

[1] Interaction Lab, Heriot-Watt University, Edinburgh, email: i.efstathiou, o.lemon@hw.ac.uk

3.1 Evaluation without Manipulation

Trained RLA (goal-oriented) vs. 3 Bots: Our trained RLA resulted in a performance of 32.66%[2], while those of the Bots were 22.9%, 22.66% and 21.78% respectively. This was interesting because it proved that our generic 2-player version of the game (Section 2) was enough to train a successful policy for the multi-player version of the game, by effectively treating all three opponents as one. Hence our RLA proposed only *public trades*. Furthermore the 32.66% performance of our RLA was *around 7% better than that of* [8], who trained it in the real multilateral negotiations environment (JSettlers).

Trained RLA (goal-oriented) vs. 3 Bayes: In this experiment our trained agent scored a performance of 36.32%, which is much higher than those of the three Bayes agents. Their performances were 21.43%, 21.02% and 21.23% respectively.

3.2 Evaluation with Manipulation

Here we evaluated our previously trained dishonest RL policies against the 3 types of Manipulated Bots and the Bayes agents.

Trained Dishonest RLA vs. 3 Manipulated Bots: In this experiment the 3 manipulated Bots win rates were 21.44%, 20.79% and 21.42% respectively. Our trained Dishonest RLA won by 36.35%.

Trained Dishonest RLA vs. 3 Manipulated Bots (Weights based on Building Plan): The Bot's probabilities are adjusted further according to the building plan (BP) in this case. That means that the Bots are initially biased towards specific resources, as the BP indicates the next piece to build (e.g. city). The results of this experiment were still satisfying: the 3 manipulated Bots won by 22.53%, 21.47% and 21.8% respectively. Our trained Dishonest RLA won by 34.2%.

Trained Dishonest RLA vs. 3 Manipulated Bots (Weights based on Building Plan and Resource Quantity): This case is identical to the above but the trade probabilities are additionally adjusted according to the goal resource quantity. The results of this experiment for the trained Dishonest RL policies were as good as the above: the 3 manipulated Bots win rates were 21.72%, 21.5% and 22.47% respectively. Our trained Dishonest RLA won by 34.33%. This result, along with the two above, suggested that the RLA's dishonest manipulative policies were very effective against the Bots of the multi-player version of the game, showing that our transition from a bilateral negotiation environment to a multilateral one was successful.

Trained Dishonest RLA vs. 3 Bayes: We hypothesised in this case that the human players might have been affected by their opponents' manipulation (if any occurred in the data collection [1]), and we wanted to test that by using our Dishonest policy. The results proved our hypothesis: the 3 Bayes agents won by 21.97%, 20.58% and 21.64% respectively. Our trained Dishonest RLA won by 35.81%. This was an evidence that the Bayes agents were indeed affected by manipulation, and now by the Dishonest RLA's manipulative policy too, and its success resulted in almost 14% more winning games.

4 Conclusion

We showed that our trained bilateral RL policies from our generic version of "Catan" were able to outperform (by at least 10%) the agents of the JSettlers [11] environment and even managed to successfully manipulate them. That demonstrated how successful trained policies from bilateral negotiations can be, when evaluated in more complex multilateral ones, even compared to those which are trained in these multilateral negotiations. Hence training RL policies in complex multilateral negotiations may be unnecessary in some cases. Furthermore, by considering all of the opponents as one player, and by proposing public trades for all players, we bypass complexities that arise by personalizing the agent's trading proposals for each distinct opponent. Our findings show that an explicit model of each adversary is not required for successful RL policies to be learned in this case. Ultimately, it suggests that an implicit model of a complex trading scenario may be enough for effective RL, providing that efficient selection of the state representation and of the actions has been made.

Further work explores Deep Reinforcement Learning approaches to trading dialogue [2].

[2] The baseline performance in a four-player game is 25%

Acknowledgements

This work is funded by the ERC, grant no. 269427 (STAC project).

REFERENCES

[1] Stergos Afantenos, Nicholas Asher, Farah Benamara, Anais Cadilhac, Cedric Degremont, Pascal Denis, Markus Guhe, Simon Keizer, Alex Lascarides, Oliver Lemon, Philippe Muller, Soumya Paul, Verena Rieser, and Laure Vieu, 'Developing a corpus of strategic conversation in The Settlers of Catan', in *Proceedings of SemDial 2012*, (2012).

[2] Heriberto Cuayahuitl, Simon Keizer, and Oliver Lemon, 'Strategic Dialogue Management via Deep Reinforcement Learning', in *NIPS workshop on Deep Reinforcement Learning*, (2015).

[3] Ioannis Efstathiou and Oliver Lemon, 'Learning non-cooperative dialogue behaviours', in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pp. 60–68, Philadelphia, PA, U.S.A, (2014).

[4] Ioannis Efstathiou and Oliver Lemon, 'Learning to manage risks in non-cooperative dialogues.', in *Proceedings of the 18th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2014 - DialWatt)*, pp. 173–175, Edinburgh, Scotland, U.K., (2014).

[5] Ioannis Efstathiou and Oliver Lemon, 'Learning non-cooperative dialogue policies to beat opponent models: "the good, the bad and the ugly"', in *Proceedings of the 19th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2015 - GoDial)*, pp. 33–41, Gothenburg, Sweden, (2015).

[6] Kallirroi Georgila and David Traum, 'Reinforcement learning of argumentation dialogue policies in negotiation', in *Proc. INTERSPEECH*, (2011).

[7] Markus Guhe and Alex Lascarides, 'Game strategies in *the settlers of catan*', in *Proceedings of the IEEE Conference on Computational Intelligence in Games*, Dortmund, (2014).

[8] Simon Keizer, Heriberto Cuayahuitl, and Oliver Lemon, 'Learning trade negotiation policies in strategic conversation', in *The 19th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2015 - goDIAL)*, pp. 104–112, (2015).

[9] Verena Rieser and Oliver Lemon, *Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation*, Theory and Applications of Natural Language Processing, Springer, 2011.

[10] Verena Rieser, Oliver Lemon, and Xingkun Liu, 'Optimising information presentation for spoken dialogue systems', in *Proc. ACL*, (2010).

[11] R. Thomas and K. Hammond, 'Java settlers: a research environment for studying multi-agent negotiation', in *Proc. of IUI '02*, pp. 240–240, (2002).

[12] David Traum, 'Extended abstract: Computational models of non-cooperative dialogue', in *Proc. of SIGdial Workshop on Discourse and Dialogue*, (2008).

[13] Steve Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, 'The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management', *Computer Speech and Language*, **24**(2), 150–174, (2010).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1642

Substantive Irrationality in Cognitive Systems

Pierre Bisquert[1] and **Madalina Croitoru**[2] and **Florence Dupin de Saint-Cyr**[3] and **Abdelraouf Hecham**[2]

Abstract. In this paper we approach both procedural and substantive irrationality of artificial agent cognitive systems and consider that when it is not possible for an agent to make a logical inference (too expensive cognitive effort or not enough knowledge) she might replace certain parts of the logical reasoning with mere associations.

1 INTRODUCTION

In artificial agents two kinds of biases have been highlighted ([8], [12], [14]). On one hand, the agent's beliefs and preferences may be incomplete and the agent may not know all the preferences or beliefs needed for complete reasoning (*e.g.* the agent's utility function is not available, or some constraints about the real world are not known). This kind of representational issued biases refers to the so called Type 1 irrationality or substantive irrationality that concerns the compliance of the results of reasoning with the agent's explicit goals and beliefs. For instance, a substantive irrational agent may eat fat while its rational goals and beliefs are in favor of healthy food. Type 2 irrationality, also known as procedural irrationality, concerns with the case when, due to the fact that computational resources (time or space available for representing and reasoning) are limited, the agent needs to make good choices in the process of deciding how to apply its efforts in reasoning. In this case what is rational for one agent is not rational for another with different limitations. Achieving procedural rationality means making rational choices about what inferences to perform, how to apply them, basically thinking about how to think. We investigate both substantive and procedural irrationality and build upon the model proposed in [4, 3]. We propose a more natural transition between two systems of reasoning: a logic based and an association based ones and propose a first cognitive model for substantive and procedural irrational agents that accounts for utterance acceptance in a logic encoding beliefs and preferences.

2 AGENT COGNITIVE MODEL

We define the cognitive model of an agent to contain beliefs, opinions, preferences and associations. The beliefs are represented using a finite set B of formulas taken from a propositional language $\mathcal{B}_\mathcal{L}$. We define an *opinion* about a belief $\varphi \in \mathcal{B}_\mathcal{L}$, denoted $\heartsuit\varphi$ (and resp. $\heartsuit\!\!\!\!\blacksquare\varphi$) as a constraint, that imposes to the situations where φ holds to be preferred (resp. strictly preferred) to the situations where φ does not hold, the opinions are gathered in a finite base $O \subset \mathcal{O}_\mathcal{L}$ where $\mathcal{O}_\mathcal{L}$ is the set of opinion formulas (that are either basic opinions or Boolean combination of them). A basic preference is a formula of the form $\alpha \trianglerighteq \beta$ (resp. $\alpha \triangleright \beta$) where $\alpha, \beta \in \mathcal{B}_\mathcal{L}$, interpreted as constraints on the preferences such that the situations where α holds should be preferred (resp. strictly preferred to) situations where β holds. Associations (elicited using [13]) encode Kahneman's System 1 [16], that is a human reasoning system dealing with quick, instinctive and heuristic thoughts. We denote by $\mathcal{A} = (\mathcal{B}_\mathcal{L} \cup \mathcal{O}_\mathcal{L} \cup \mathcal{P}_\mathcal{L}) \times (\mathcal{B}_\mathcal{L} \cup \mathcal{O}_\mathcal{L} \cup \mathcal{P}_\mathcal{L})$ the set of all possible associations between any pair of formulae. We also denote $\mathcal{B}_R$, $\mathcal{P}_R$, $\mathcal{O}_R$, $\mathcal{A}_R$ the sets of inference rules that allow us to deduce new beliefs, preferences, opinions and associations.

We define the notion of "reasoning" as the process of inferring a formula φ using a rule application sequence R from the set of logical, preference, opinion and association rules on an initial set of pieces of information K, denoted $K \vdash_R \varphi$. We call the successive application of rules R a "reasoning path". Inside this reasoning path we differentiate the use of logical inference rules from the use of an association rule. A reasoning on a formula can be achieved using different reasoning paths, each path has a cost depending on the cognitive effort needed to use the rules it contains. Intuitively it is less costly to use association rules than logical inference rules and among associations some are more or less far-fetched than others. In order to represent the cognitive effort involved by the reasoning, we are going to use the effort function e that associates an effort to the associations and the inference rules used.

A cognitive model is defined as a tuple of beliefs, opinions, preferences, associations and their subsequent effort for reasoning.

Definition 1 (Cognitive model) *A* cognitive model *is a tuple*

$$\kappa = (B, O, P, A, e, \sqsubseteq)$$

- $B \subseteq \mathcal{B}_\mathcal{L}$ *is a set of wff representing beliefs,*
- $O \subseteq \mathcal{O}_\mathcal{L}$ *is a set of wff representing opinions,*
- $P \subseteq \mathcal{P}_\mathcal{L}$ *is a set of wff representing preferences,*
- $A \subseteq \mathcal{A}$ *is a binary relation representing the associations between formulae,*
- e *is a function* $\mathcal{B}_R \cup \mathcal{P}_R \cup \mathcal{O}_R \cup \mathcal{A}_R \rightarrow \mathbb{N} \cup \{+\infty\}$ *that represents the effort required to infer with each inference rule.*
- $\sqsubseteq \ \subseteq \mathcal{R} \times \mathcal{R}$ *is a preference relation based on* e *over reasoning paths;* $R_1 \sqsubseteq R_2$ *means* R_1 *is better than* R_2.

3 Argument Evaluation

In our work, agents reason about implicative utterances [2] and more generally about enthymemes (see [5, 9]) or arguments.

Definition 2 (Argument) *Given* $\mathcal{L} = \mathcal{B}_\mathcal{L} \cup \mathcal{P}_\mathcal{L} \cup \mathcal{O}_\mathcal{L}$*, an argument* arg *is a pair* $(\varphi \in \mathcal{L}, \alpha \in \mathcal{L})$.

An argument (φ, α) intends to state that having some beliefs and preferences described by φ leads to concluding α. In argumentation literature, some works (such as *e.g.,* [17]) propose to base the decision about whether or not an argument is acceptable on some critical

[1] GraphIK, INRA, Montpellier, France
[2] GraphIK, LIRMM, Univ. Montpellier, France
[3] IRIT, Univ. Toulouse, France

questions. For the sake of generality, we propose to base the evaluation of arguments on the classical notions that are used in argumentation in order to explain "attacks" between arguments. Classically three notions are used, called rebuttal, undermine and undercut. More precisely an argument (φ, α) can be attacked either on its conclusion (α) directly or on a part of its premises (φ) or on the link between the premises and the conclusion.

- CQ_1: $(B, O, P, A) \vdash \neg\alpha$ (is it possible to attack the conclusion?)
- CQ_2: $(B, O, P, A) \vdash \neg\varphi$ (is it possible to attack the premises?)
- CQ_3: $\varphi \vdash \alpha$ (does the premises allow to infer the conclusion?)

To define what are the answers to critical questions we will use reasoning paths. Based on the ELM model [6] we suppose here that each agent has a cognitive availability that represents the maximum cognitive effort ca she is willing to make in order to reason on an argument.

Given an argument and a finite cognitive availability ca, we can compute all the possible reasoning paths wrt ca. A positive answer to a critical question corresponds to the existence of a reasoning path that requires a cognitive effort under ca. If there is no such path, the answer to the critical question is negative.

Definition 3 (Positive/negative answers)
Given an inference $CQ : h \vdash c$ and a cognitive availability ca, given a reasoning path R, we denote:
$proof_{ca}(R, CQ) \stackrel{def}{=} Eff(R) \leq ca$ and $h \vdash_R c$ where $Eff(R) = \sum_{r \in R} e(r)$. Moreover, we say that:

- *CQ is* answered positively *wrt to ca iff $\exists R$ s.t. $proof_{ca}(R, CQ)$, denoted $positive_{ca}(CQ)$,*
- *CQ is* answered negatively *wrt to ca iff $\nexists R$ s.t. $proof_{ca}(R, CQ)$, denoted $negative_{ca}(CQ)$.*

Thanks to the previous definitions, we are in position to formally define the problem of argument evaluation wrt an agent cognitive model and its cognitive availability.

Definition 4 (Potential status of arguments) *Given an agent with a cognitive model $\kappa = (B, O, P, A, e, \sqsubseteq)$, a cognitive availability ca and an argument $arg = (\varphi, \alpha)$. Let $CQ_1 = B \cup O \cup P \cup A \vdash \neg\alpha$, $CQ_2 = B \cup O \cup P \cup A \vdash \neg\varphi$, $CQ_3 = \varphi \vdash \alpha$. We say that arg is:*

- acceptable$_{ca}$ *iff $\forall c_3 \leq ca$ s.t. $positive_{c_3}(CQ_3)$ and $\forall (c_1, c_2)$ s.t. $c_1 + c_2 + c_3 = ca$, we have $negative_{c_1}(CQ_1)$ and $negative_{c_2}(CQ_2)$.*
- rejectable$_{ca}$ *iff $positive_{ca}(CQ_1)$ or $positive_{ca}(CQ_2)$ or $negative_{ca}(CQ_3)$.*
- undecidable$_{ca}$ *if it is both acceptable$_{ca}$ and rejectable$_{ca}$.*

In other words, an argument is acceptable if the link between the premises and the conclusion can be established and the agent has not enough cognitive ability to find a counter-example for either the conclusion (CQ_1) or the premises (CQ_2). In order to be able to reject an argument it is enough to find a counterexample corresponding to one of the two first critical questions or to not have a sufficient cognitive ability to infer the causal link. An undecidable argument may be found if there is a proof for CQ_3 and for CQ_1 with a total cost above ca.

4 DISCUSSION AND RELATED WORK

The highly influential cognitive psychology work in dual systems ([16, 7, 11, 1, 10, 15]) associate such biases with two reasoning systems: one system that is slow but logically precise and another system that is fast but logically sloppy. The distinction does not make clear the interaction between biases due to logically flawed reasoning and biases due to sub optimal reasoning choices done because of cognitive limitations. This distinction is interesting when addressing the evaluation of biased argument.

In this paper we consider the problem of argument evaluation by agents that are both logically biased (*i.e.* may either reason exclusively logically or by combining logical reasoning with associations) and that have a limited cognitive availability. Following the highly influential cognitive psychology work in dual systems ([16, 7, 11, 1, 10, 15]) proposal considers that, when it is not possible for an agent to make a logical inference (too expensive cognitive effort or not enough knowledge), she might replace certain parts of the logical reasoning with mere associations. Using associations may lower the reasoning effort needed for argument evaluation and subsequently affect the argument acceptance.

ACKNOWLEDGEMENTS

The authors acknowledge the support of ANS1 1208 IATE INCOM INRA grant and ANR-12-0012 grant QUALINCA. We particularly thank Ulrike Hahn for her clear-sighted comments and the organizers of Dagstuhl Seminar 15221 on "Multi-disciplinary Approaches to Reasoning with Imperfect Information and Knowledge".

REFERENCES

[1] C. Beevers, 'Cognitive vulnerability to depression: A dual process model', *Clinical Psychology Review*, **25**(7), 975 – 1002, (2005).
[2] P. Besnard and A. Hunter, 'A logic-based theory of deductive arguments', *Artificial Intelligence*, **128**(1-2), 203 – 235, (2001).
[3] P. Bisquert, M. Croitoru, and F. Dupin de Saint-Cyr, 'Four ways to evaluate arguments according to agent engagement', in *Brain Informatics and Health*, 445–456, Springer, (2015).
[4] P. Bisquert, M. Croitoru, and F. Dupin de Saint-Cyr, 'Towards a dual process cognitive model for argument evaluation', in *Scalable Uncertainty Management*, 298–313, Springer, (2015).
[5] E. Black and A. Hunter, 'Using enthymemes in an inquiry dialogue system', in *Proc of the 7th Int. Conf. on Autonomous Agents and Multiag. Syst. (AAMAS 2008)*, pp. 437–444, (2008).
[6] J. Cacioppo and R. Petty, 'The elaboration likelihood model of persuasion', *Advances in Consumer Research*, **11**(1), 673–675, (1984).
[7] P. Croskerry, G. Singhal, and S. Mamede, 'Cognitive debiasing 1: origins of bias and theory of debiasing', *BMJ Quality & Safety*, **22**(Suppl 2), 58–64, (2013).
[8] J. Doyle, 'Rationality and its roles in reasoning', *Computational Intelligence*, **8**(2), 376–409, (1992).
[9] F. Dupin de Saint-Cyr, 'Handling enthymemes in time-limited persuasion dialogs', in *Int. Conf. on Scalable Uncertainty Management (SUM)*, number 6929 in LNAI, pp. 149–162. Springer-Verlag, (2011).
[10] S. Epstein, 'Integration of the cognitive and the psychodynamic unconscious.', *American Psychologist*, **49**(8), 709, (1994).
[11] J. Evans and J. Curtis-Holmes, 'Rapid responding increases belief bias: Evidence for the dual-process theory of reasoning', *Thinking & Reasoning*, **11**(4), 382–389, (2005).
[12] I. Good, 'The probabilistic explanation of information, evidence, surprise, causality, explanation, and utility', *Foundations of statistical inference*, 108–141, (1971).
[13] A. Hecham, M. Croitoru, P. Bisquert, and P. Buche, 'Extending gwaps for building profile aware associative networks', in *Proceedings of ICCS 2016*, pp. 43–58, (2016).
[14] H. Simon, 'From substantive to procedural rationality', in *25 Years of Economic Theory*, 65–86, Springer, (1976).
[15] S. Sloman, 'The empirical case for two systems of reasoning.', *Psychological Bulletin*, **119**(1), 3, (1996).
[16] A. Tversky and D. Kahneman, 'Judgment under uncertainty: Heuristics and biases', *Science*, **185**(4157), 1124–1131, (1974).
[17] D. Walton, C. Reed, and F. Macagno, *Argumentation Schemes*, Cambridge University Press, Cambridge, 2008.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1644

A History Tree Heuristic to Generate Better Initiation Sets for Options in Reinforcement Learning

Alper Demir and **Erkin Çilden** and **Faruk Polat** [1]

Abstract.

Options framework is a prominent way to improve learning speed by means of temporally extended actions, called options. Although various attempts focusing on how to derive high quality termination conditions for options exist, the impact of initiation set generation of an option is relatively unexplored. In this work, we propose an effective heuristic method to derive useful initiation set elements via an analysis of the recent history of events.

1 Introduction

Reinforcement Learning (RL) is commonly defined upon *Markov Decision Process* (MDP) model. Q-Learning [9] is probably the most popular RL algorithm, often credited for its simplicity and ease of use.

Options framework [8] is one of the prominent abstraction formalisms based on the semi-MDP model, which devices a way to define and invoke timed actions by means of composite actions on top of the MDP model in order to improve performance via a divide-and-conquer approach. Still keeping the unit time transition dynamics of MDPs, an action can now be generalized in the sense that it may last for a number of discrete time steps and referred to as an *option*. In other words, an "option" is a temporally extended counterpart of an "action".

An option is made up of three components: (1) an initiation set (states at which an option may start), (2) option's local policy, and (3) a termination condition (how an option terminates). A natural extension of Q-Learning to include options is Macro-Q Learning [3], where the value of each primitive action is updated as in regular Q-Learning while the value of an option is updated via a separate option-level update rule.

Options framework itself does not impose any strategy for designing meaningful or useful options. Nevertheless, there are methods that attempt to automatically generate options. One of the prominent family of algorithms are based on *subgoal discovery* techniques [4] which usually focus on the termination condition, since a partitioning mechanism based on bottlenecks would naturally interpret a splitting state or a region as the termination criterion.

Initiation sets, on the other hand, are usually defined via a simpler premise, like "all states other than termination states," [1] hoping for the action selection mechanism to restrict choices during learning later on. Restrictive parameters like "option lag" can be used to provide a problem specific limit for the option length [5]. Some methods incorporate a *reachability* criterion for states to the ones in the termination condition as a heuristic [7]. Almost none of the existing studies focuses solely on the potential of the initiation set, which we believe would improve the quality of generated options.

In this study, we propose a history tree based heuristic method to identify useful states to generate initiation sets of options. We empirically show that, since the initiation set of an option is an integral part of the abstraction, a good heuristic can positively affect the overall learning performance.

2 History Tree Heuristic

As an integral part of the option generation process, selection of states for the initiation set is important. Guiding the option with some information about the environment characteristics, like the relative orientation of reward peaks, is likely to have positive impact on option performance, especially at the initial stages of learning.

This work aims to improve the widely used greedy approach for initiation sets, which selects the initiation set states among the ones visited before the terminal state occurrences within an episode. The number of transitions to check prior to the terminal state of the option is an externally supplied parameter called the *option lag*.

Our intuition is that a terminal state usually possesses a relative orientation, in terms of short term benefit, with respect to some distinctive states of the state space. The proposed method is a heuristic making use of a history tree in order to generate an initiation set. It aims to construct options by means of the relative orientation of the goal state, meaning that it restricts the initiation set of a new option with the states that are more likely to construct it with an implicit "direction". The method also aims to incorporate a larger set of states as the initiation set compared to the option lag heuristic, due to the elimination of the redundant loops.

Given a terminal state s_t, our method creates a tree having s_r as the root node, which is possibly a goal state or a state yielding a reward peak, providing an accurate direction for the option to be created. The algorithm traverses all the sub-histories (or episodes) ending with s_r and generates a tree of shortest paths from each visited node to s_r. Every episode ending with s_r is traversed from the last visited state to the first, setting the parent states for each state as a representative of the best state to go on the way to s_r. As a result, a history tree is generated having the vertex set consisting of all states observed during the traversal and the edge set consisting of transitions from every state to its parent.

The method then employs a traversal in the tree (we preferred a breadth-first), starting from the terminal state s_t to add every visited state (except s_t) to the initiation set. With terminal state s_t in depth 0, the maximum depth of the traversal is provided as a parameter called *option depth*.

[1] Department of Computer Engineering, Middle East Technical University, 06310 Ankara, Turkey, {ademir,ecilden,polat}@ceng.metu.edu.tr

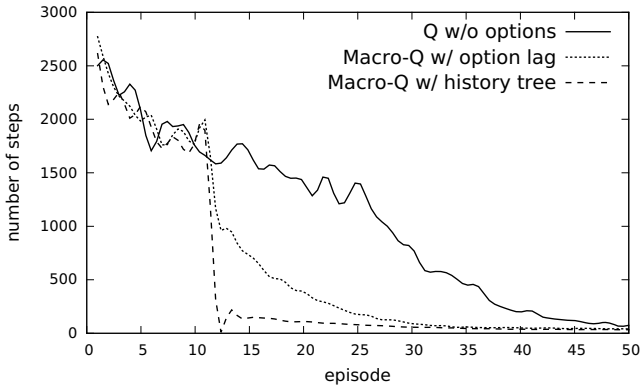


Figure 1. Average number of steps to goal for 2-rooms problem

3 Experiments

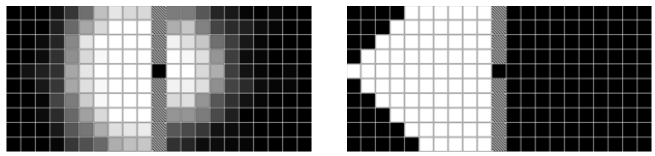
The problem that we used in the experiments is the 2 rooms grid-world domain with 1 doorway in between (*2-rooms*), which is a classical problem with a single bottleneck state [6], naturally partitioning the problem into two subproblems. In order to make a fair comparison between the history tree and the option lag heuristics, the terminal state (the state at the doorway) is provided to the agent beforehand. The agent starts option generation after a sufficient number of episodes is experienced, which is set as 10 for our experimentation.

After building the initiation sets, *Experience Replay* [2] is employed by the agent in order to generate the policy starting from the states of the initiation set to reach the terminal state. For the experience replay mechanism, a large reward is yielded upon reaching the terminal state, a large punishment value is provided for leaving the initiation set and a small punishment is applied for every other transition. The learning parameters $\alpha = 0.125$ and $\gamma = 0.9$ are used for experience replay sessions, where each session is repeated 10 times. An option is terminated upon reaching the terminal state with probability 1.0, while it is forbidden to terminate at any state in the initiation set.

The parameters *option lag* and *option depth* are set in such a way that the resulting options have initiation sets with approximately the same number of states on the average for both approaches. Macro-Q learning is used with learning parameters $\alpha = 0.05$, $\gamma = 0.9$ and $\epsilon = 0.1$. Both methods are compared against the regular Q-Learning using the same learning parameters without options. The test results are averaged over 200 experiments.

Results and Discussion It is clear from the result given in Figure 1 that, Macro-Q learning with options generated using the history tree heuristic for initiation set clearly outperforms the one with the option lag heuristic, in terms of average number of steps to reach the goal state. History tree provides an advantage to the Macro-Q learning algorithm, beginning from the early stages of the learning process.

In order to have a deeper insight on how our method realizes this improvement, Figure 2 visualizes the number of occurrences of each state within an initiation set on the average. Brighter color means that the state has been selected more as an initiation set element, while darker means less.



(a) option lag (b) history tree

Figure 2. The number of occurrences of each state within an initiation set for the 2-rooms problem

Although both methods have nearly the same number of states in the initiation sets, since the agent can cycle in loops leading back to the doorway, the option lag mechanism tends to choose many of the states on the right side of the doorway and add them to the initiation set. However, the initiation set generated by history tree heuristic consists only of the states of the room on west. This differentiation prevents the agent to explore an option among the states on the right side and results in improved learning performance.

4 Conclusion

In this paper, we propose a goal oriented option generation method utilizing a history tree heuristic. It restricts the initiation set of an option to the states from which employing the option would be useful. A future work can focus on root state selection to determine the direction of the generated options. Additionally, construction of the option's local policy by using the generated tree has the potential to improve option quality.

ACKNOWLEDGEMENTS

This work is partially supported by the Scientific and Technological Research Council of Turkey under Grant No. 215E250.

REFERENCES

[1] Nicholas K. Jong, Todd Hester, and Peter Stone, 'The utility of temporal abstraction in reinforcement learning', in *Proc. of the 7th Int. Conf. on Autonomous Agents and Multiagent Systems*, volume 1 of *AAMAS '08*, pp. 299–306, Portugal, (2008).

[2] Long-Ji Lin, 'Self-improving reactive agents based on reinforcement learning, planning and teaching', *Mach. Learn.*, **8**(3), 293–321, (1992).

[3] Amy McGovern, Richard S. Sutton, and Andrew H. Fagg, 'Roles of macro-actions in accelerating reinforcement learning', *Proc. of the 1997 Grace Hopper Celebration of Women in Computing*, 13–18, (1997).

[4] Ozgur Simsek, *Behavioral Building Blocks for Autonomous Agents: Description, Identification, and Learning*, Ph.D. thesis, University of Massachusetts Amherst, 2008.

[5] Ozgur Simsek and Andrew G. Barto, 'Using relative novelty to identify useful temporal abstractions in reinforcement learning', in *Proc. of the 21st Int. Conf. on Machine Learning*, ICML '04, pp. 95–102. ACM, (2004).

[6] Ozgur Simsek, Alicia P. Wolfe, and Andrew G. Barto, 'Identifying useful subgoals in reinforcement learning by local graph partitioning', in *Proc. of the 22nd Int. Conf. on Machine Learning*, ICML '05, pp. 816–823. ACM, (2005).

[7] Martin Stolle and Doina Precup, 'Learning options in reinforcement learning', in *Proc. of the 5th Int. Symposium on Abstraction, Reformulation, and Approximation*, volume 2371 of *LNCS*, pp. 212–223. Springer Berlin Heidelberg, (2002).

[8] Richard S. Sutton, Doina Precup, and Satinder Singh, 'Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning', *Artif. Intell.*, **112**(1-2), 181–211, (1999).

[9] Chris Watkins, *Learning from Delayed Rewards*, Ph.D. thesis, Cambridge University, 1989.

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1646

On Truthful Auction Mechanisms for Electricity Allocation

Heng Song[1] and **Junwu Zhu**[1,2,*] and **Bin Li**[1,3] and **Yi Jiang**[1]

Abstract. As technology evolves and electricity demand rises, more and more research focus on the efficient electricity allocation mechanisms so as to make consumer demand adaptive to the supply of electricity at all times. In this paper, we formulate the problem of electricity allocation as a novel combinatorial auction model, and then put forward a directly applicable mechanisms. It is proven that the proposed mechanism is equipped with some useful economic properties and computational traceability. Our works offer potential avenues for the stduy about efficient electricity allocation methods in smart grid.

1 INTRODUCTION

Recently, there is a sudden demand for electricity with the economic development of science and technology, then the task to make consumer demand adaptive to electricity supply at all times becomes especially challenging[1], which can reduce the risk of disastrous electricity network collapses, and bring financial and environmental benefits—as some generators can be run on idle, or long time overload may damage the generators, which even exerts a bad influence on the electricity network[2]. In order to over the challenge, we are motivated to design the mechanisms for electricity allocation, which is mainly faced with three obstacles:(1)***Truthfulness***; (2)***System Efficiency***; (3)***Low execution time***. Many excellent mechanisms for making consumer demand adaptive to electricity supply have been designed in the existing literatures(e.g.[3]-[5]), in particular, two strains of thought seem to dominate the effort to deal with this problem: one is abridging customers' consuming activities for electricity, the other is shifting customers' consuming activities for electricity to off-peak hours in order to reduce peak-to-average ratio(PAR).

In this paper, we present a novel game model of multi time slots combinatorial auction for the electricity allocation problem, in which electricity consumers can bid for the electricity in multiple time slots, and the electricity in each time slot can be simultaneously sold to multiple electricity consumers, and then propose a combinatorial auction mechanism with dynamic price called TAMEA-DP. In addition, we make it possible that the proposed mechanism satisfies individual rationality, budget balance and truthfulness.

2 THE MODEL AND OBJECTIVE

We assume that the large utility company (i.e., seller) is trustworthy, and has a set of time slots $M = \{1, 2, \cdots, m\}$, in which the quantities Q_j of the electricity (i.e. electricity capacity) sold to buyers. The electricity in each time slot can be simultaneously sold to multiple buyers, only if its electricity capacity is not reached. Furthermore, the seller declares the discrete price curve $p^j(.)$, which indicates the minimum unit price of electricity that the utility company wishes to sell. We denote the discrete price curve $p^j(.)$ as a two-level decreasing function: $p^j(< h_j) = p_j^H$ and $p^j(\geq h_j) = p_j^L$, with $p_j^L < p_j^H$. Here, h_j is the price threshold in time slot j, p_j^H denotes the normal unit price, and p_j^L is similar to a discount price or group price. In addition, we also further assume that the discount price in any time slot is less than the normal price in any other time slot. The seller's offer is defined by $O = \{< Q_j, p^j(.) >\}_{j \in M}$.

We also assume that there is a set $N = \{1, 2, \cdots, n\}$ of potential electricity consumers (i.e., buyers). Each buyer $i \in N$ requests her electricity demand for every time slot and has a valuation v_i on the requested electricity. Let $\mathbf{R}$ be an $N * M$ matrix where each row of the matrix, $\mathbf{r}_i$ represents the requested electricity demand of buyer i. Each entry r_{ij} is the electricity demand of buyer i for time slot j. The total aggregated demand in time slot j is $S_j = \sum_{i \in N} r_{ij}$, and the total demand of each agent during the whole planning period is $\tau_i = \sum_{j \in M} r_{ij}$. The electricity valuation v_i is a private information to the buyer i, and then to join the auction, each buyer i should also submit her bid b_i on the requested electricity. Obviously, if $b_i = v_i$, then buyer i is truthful, otherwise she lies about her valuation. In this auction, the buyers simultaneously submit their sealed offers, denoted by $B = (B_1, B_2, \cdots, B_n)$, where $B_i = (r_i, b_i)$. To keep the production line running, we assume that each buyer submits a single bid and is single-minded in the auction. In addition, we denote the charge of buyer $i \in N$ by p_i ,and define the utility u_i of buyer i to be the difference between her valuation v_i and the charge p_i, i.e., $u_i = v_i - p_i$. As stated above, the electricity allocation problem (EAP) can be defined as four constraints:

(1) $\sum_{i \in W} z_{ij} \leq Q_j \quad \forall j \in M$; (2)$g(\mathbf{z}_i) < p_i \leq b_i \quad \forall i \in W$;
(3)$\mathbf{z}_i = \mathbf{0}, p_i = 0 \quad \forall i \notin W$; (4)$\sum_{j \in M} z_{ij} = \tau_i \quad \forall i \in W$;

In this paper, we aim to design a Truthful Auction Mechanism for Electricity Allocation With Dynamic Price (TAMEA-DP), denoted by $\psi = (B, O)$, where given B and O, the auctioneer determines the winning buyer set W, the electricity allocation $\mathbf{z}_i$ for each winning buyer $i \in W$ and the charge p_i for each buyer $i \in N$, such that: (1) For each buyer i, u_i is maximized when bidding v_i. (2) For each buyer, $p_i \leq b_i$; (3) $\sum_{i \in W} p_i \geq \sum_{j \in M} p^j(S_j) * S_j$, which implies that the profit of auctioneer $U_{auc} \geq 0$

3 DESIGN OF TAMEA-DP

TAMEA-DP consists of three schemes: winner determination scheme which is used for deciding who will be the winning buyer,

[1] Institute of Information Engineering, Yangzhou University, China, email: song_heng@foxmail.com, {jwzhu, lb, yj}@yzu.edu.cn
[2] Department of Computer Science and Engineering, Guelph University, Canada; *Corresponding author
[3] State Key Laboratory for Novel Software Technology, Nanjing University, China

allocation scheme which aims to how to allocate the electricity, and charging scheme which determines the payment for buyers.

1)Winner Determination Scheme receives the seller's offer and buyers' bid as its input, and aims to determine a set of winning buyers. This algorithm firstly initializes that all buyers are losing buyers, and then ranks the buyers in non-increasing order of their bid density d_i, which can be calculated as: $d_i = \frac{v_i}{|\sum_{j \in M} r_{ij} k_j|^q}, where\ \ q > 0$. The following steps are the key process: this algorithm checks whether each buyer i' requested electricity go beyond the available electricity of each time slot. If not, buyer i will be added to the set W. Next, for each buyer $i \in W$, we assess whether her bid satisfies constraint (2), i.e. buyer i' bid exceeds the weighted sum of dynamic prices for all the requested electricity. If not, we will remove buyer i from the set W, and add her to the set L in which buyers' bids satisfy constraint (1) but not satisfy constraint (2). This process will be repeated until there is no buyer removed from the set W.

2)Allocation Scheme collects the set of winning buyers, and sellers' offer. To begin, this algorithm separates time slots in M into two sets: $M^+ = \{j | j \in M \text{ and } S_j > h_j\}$,$M^- = \{j | j \in M \text{ and } S_j \leq h_j\}$, and then the total electricity which goes beyond / within the price threshold can be calculated as $T^+ = \sum_{j \in M^+} (S_j - h_j)$ and $T^- = \sum_{j \in M^-} (h_j - S_j)$ respectively. Next, we should consider two cases: $T^+ \leq T^-$ and $T^+ > T^-$, and the algorithm proceeds as follows: If $T^+ \leq T^-$, the algorithm then ranks time slots in M^- by $l_j = S_j - h_j$ in non-decreasing order. Next, starting from the time slot with lowest l_j value in M^-, we increase its total aggregated electricity S_j until it is equal to its price threshold. This process is repeated until whole electricity which goes beyond the price threshold is transferred, i.e., $T^+ = 0$. Similarly, if $T^+ > T^-$, the algorithm firstly sorts time slots in M^+ by p_j^L in non-increasing order, and then greedily reduces its total aggregated electricity S_j in the sorted order until it is equal to its price threshold. We continue this way until there is no time slot in which the total electricity goes within its price threshold, i.e., $T^- = 0$. Finally, the adjusted electricity in each time slot will be equally shared by the winning buyers.

3)Charging Scheme collects the seller's offer, buyers' bid and the set of winners, and aims to determine the prices that each buyers should pay. The idea of our charing scheme is that the losing buyers pay nothing, and the winning buyer should pay her critical value, which is the lowest possible price that she should claim for her requested electricity in order to still be the winning buyer. Therefore, the process of calculating the price that buyer i pays can be conduced as follows: If buyer i is a winning buyer, the algorithm firstly excludes her bid b_i to construct a new market setting B'. Next, it runs Winner Determination Scheme with B', and then only selects the new winning buyers compared to the initial winners to form the set LW_i of losing competitors. If LW_i is not empty, the algorithm calculates the competitive price p_i^{comp} as the product of $|\sum_{j \in M} r_{ij} k_j|^q$ and the maximum bid density in LW_i, otherwise p_i^{comp} is 0. Finally, the highest value between the competitive price p_i^{comp} and the allocation-specific reserve price $g(\mathbf{z}_i)$ is determined as the prices that winning buyer i should pay, and losing buyer should pay nothing.

4 SIMULATIONS

The simulation code is written in C# with .NetFramework 4.0 and run on a local machine. We consider linear bid density with $q = 1$, and randomly generate the buyers $N = \{1, 2, \cdots, 10\}$ who can randomly request between 11 and 20 electricity. In addition, buyer's valuation is $v_i = \beta \sum_{j \in M} r_{ij} * p^j(1)$, which is generated as a random value based on the unit price of each time slot and the scale factor β distributed over [0.8,1.2]. In this experiment, we also assume that the ratio of buyer's bid to her valuation is δ, then it represents that the buyer is truthful when $\delta = 1$, otherwise it means that the buyer submits a mendacious bid (It indicates that the buyer states lower and inflated bids when setting $\delta < 1$ and $\delta > 1$ respectively). As shown in Figure 1, we note that for each buyer $i \in N$, she can gain the highest utility when $\delta = 1$, i.e., no buyer can improve her utility by bidding untruthfully.

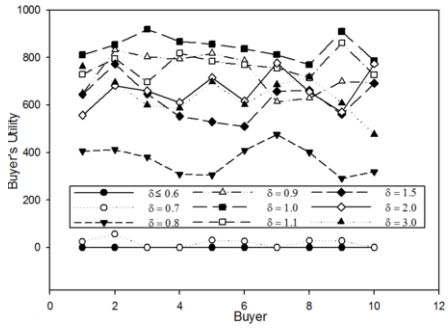


Figure 1. Truthfulness for TAMEA-DP

5 CONCLUSION

In this paper, we propose a truthful combinatorial auction mechanisms with dynamic price for allocating the electricity to achieve the goal of making consumer demand adaptive to electricity supply at all times. It is verified that the proposed mechanisms can simultaneously achieve three important economic properties including individual rationality, truthfulness and budget balance. In our future work, we will explore the impacts of different bid density on the proposed mechanisms with the different market setting, and further improve the proposed mechanism to prevent electricity consumers from collaborating with each other.

ACKNOWLEDGEMENTS

Project supported by the National Nature Science Foundation of China (Grant No.61170201, No.61070133, No.61472344), Six talent peaks project in Jiangsu Province (Grant No.2011–DZXX–032), Jiangsu Science and Technology Project (Grant No. BY2015061-06,BY2015061-08), Yangzhou Science and Technology Project (Grant No. SXT20140048, SXT20150014, SXT201510013), Natural Science Foundation of the Jiangsu Higher Education Institutions (Grant No.14KJB520041).

REFERENCES

[1] Fang X, Misra S, Xue G, et al, 'Smart Grid – The New and Improved Power Grid: A Survey', *IEEE Communications Surveys & Tutorials*, **14**(4), 944-980, (2012).

[2] Medina J, Muller N, Roytelman I, 'Demand Response and Distribution Grid Operations: Opportunities and Challenges', *IEEE Transactions on Smart Grid*, **1**(4), 193-198, (2010).

[3] Charilaos Akasiadis, Georgios Chalkiadakis, 'Agent Cooperatives for Effective Power Consumption Shifting', in *Proc. of AAAI*, pp.1263-1269, (2013),

[4] Mohsenian-Rad, H Wong, V Jatskevich, Schober R, Leon-Garcia, 'Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid', *IEEE Transactions on Smart Grid*, **1**(3), 320-331, (2010).

[5] Veit Andreas, Xu Ying, Zheng Ronghuo, et al, 'Multiagent coordination for energy consumption scheduling in consumer cooperatives', in *Proc. of AAAI*, pp.1362-1368, (2013).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1648

Towards a BDI Player Model for Interactive Narratives

Jessica Rivera-Villicana and **Fabio Zambetta** and **James Harland** [1] and **Marsha Berry** [2]

Abstract. Player Modelling is one of the challenges in Interactive Narratives (INs), where a precise representation of the players mental state is needed to provide a personalised experience. However, how to represent the interaction of the player with the game to make the appropriate decision in the story is still an open question. In this paper, we aim to bridge this gap identifying the information needed to capture the players interaction with an IN using the Belief-Desire-Intention (BDI) model of agency. We present a BDI design to mimic a players interaction with a simplified version of the interactive fiction Anchorhead.

1 INTRODUCTION

Interactive Narratives (INs) have been long popular in computer games, delivering non-linear stories, so that players can affect the direction of the narrative with their actions. Titles such as Heavy Rain [3], The Elder Scrolls [4] and The Witcher [5] implement this technique. However, the task of writing every branch of the story becomes more expensive as the author wants to give more freedom to the player, because the number of different sub-stories that need to be written increases significantly [5].

To address this issue, research in *Narrative Generation* aims to automatically generate the story tree with all its possibilities given a *plot graph* that contains a set of conditions established by the author in order to ensure the coherence of the story. These conditions include precedence of events and status of the main character, among other elements. An automatically generated IN is expected to adapt to each players preferences, raising the need for Player Modelling(PM) techniques to capture information useful to make such adaption happen. But despite the increasing popularity of Player Modelling in video games, there are only exploratory attempts focused on INs[7].

Some of the existing PM approaches tend to classify players into different behaviour categories (e.g. Tactician, Specialist, Storyteller, etc.). However, making predictions based on such models has proven challenging. Previous research has also implemented theories derived from psychology [8], but the results show that it is not accurate to model human behaviour in a virtual world using theories that are only applicable in the real world. Based on this, we propose to use the Belief-Desire-Intention model of agency, a model commonly used in Agent Programming (e.g., Non-Player Characters [3]) that is based on a model of human practical reasoning[1]. By using a model that is applicable to both the real and virtual worlds, we expect to have a more accurate player model than existing approaches. Having a more accurate player model will then give us a detailed insight into how each player discovers a specific branch of an IN.

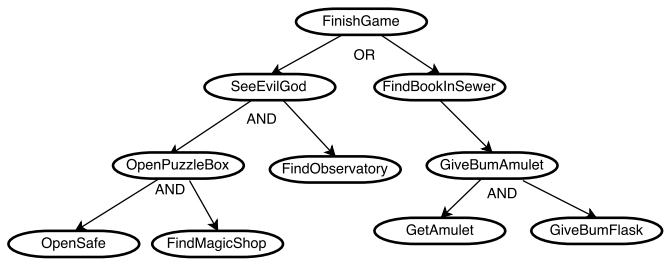


Figure 1. A basic goal tree design following the Prometheus methodology

Our aim in this paper is to design a Belief-Desire-Intention(BDI) Player Model that captures the way players interact with an Interactive Narrative(IN). The IN we use to develop this work is an extract of the Interactive Fiction "Anchorhead". This same extract has been previously used by other researchers because it captures the main structure of an IN in a relatively short game [2, 6].

2 METHODOLOGY

The main contribution of this work lies in understanding that the way we would design an accurate player model is different than that to design a normal "bot" or artificial player. If our aim was to create an optimal BDI artificial player for an IN, in this case Anchorhead, the way to do it according to the Prometheus Methodology [4] would be to take the plot graph of the IN and turn it into a goal tree. Figure 1 shows how a basic goal-plan tree of the BDI agent would look like. However, this design assumes that the player knows what to aim for since the beginning of the game. For example, an agent based on this design would go straight after the combination for the safe, knowing that this is the first step to reach one of the ends.

The resulting game trace of such agent would be much shorter in comparison with that of a real player, and it will unlikely contain plot points from the other branch. This behaviour is unrealistic, since we have observed that players in this game are usually lost at the beginning, exploring the places and trying to figure out what the goals are. The resulting trace is then much larger than the optimal, with interactions with plot-points from both story branches.

In this section, we explain our BDI design approach based on the premise that players do not interact with an IN in an optimal way, but they rather explore the world, and as they do so, they discover and prioritize their goals, until they eventually complete the game.

To replicate real players behaviour in the IN, we were led to re-think our design of the BDI model, still taking as a base the

[1] School of Science, RMIT University, Australia
[2] School of Media and Communication, RMIT University, Australia, email: {jessica.riveravillicana, fabio.zambetta, james.harland, marsha.berry}@rmit.edu.au
[3] http://www.quanticdream.com/en/#!/en/category/heavy-rain
[4] http://www.elderscrolls.com/
[5] http://www.thewitcher.com

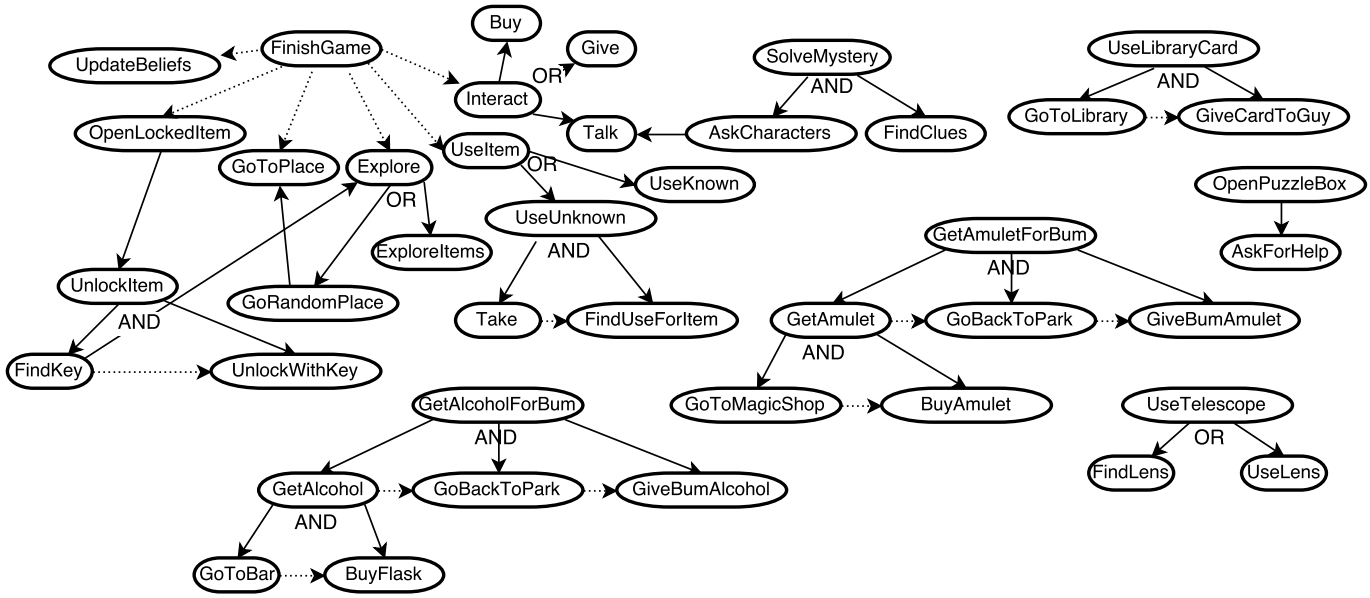


Figure 2. The goal diagram of our BDI design

Prometheus Methodology. As a result, instead of having one single goal tree with a top level goal, we designed several scattered goal trees that might interact or run concurrently as players define their goals. Figure 2 shows the set of goal trees in the BDI Player Model.

3 PERFORMANCE ANALYSIS

We tested our BDI Player Model creating an agent that completed 20 games. The first matter to discuss is how human-like the generated trace of our BDI agent looks. Below we show an example of a possible optimal trace:
[get-safe-combo, open-safe, find-magic-shop, open-puzzle-box, see-evil-god]

In this trace, every single plot-point is followed by exactly its successor in a plot branch. On the other hand, the following is the summary generated by a human:

[examine-album, get-card, get-crypt-key, get-silver-locket, read-bedroom-pages, read-basement-clippings, find-williams-coffin, see-skull, get-skull, find-magic-shop, get-library-book, get-amulet, buy-magic-ball, open-safe, get-safe-combo, open-puzzle-box, see-evil-god]

Lastly, the following is a summary trace generated by the BDI agent:

[find-magic-shop, start-talking-to-bum, get-crypt-key, read-basement-clippings, read-bedroom-pages, get-card, get-safe-combo, open-safe, examine-album, get-flask, give-bum-flask, ask-bum-about-william, show-bum-skull, ask-bum-about-crypt, open-puzzle-box, see-evil-god]

The three examples reach the same end. But as we can observe, the real human trace and the model generated trace are considerably longer than the optimal trace presented first.

Another point to analyse is whether the agent can reach both story endings. Out of 20 runs, the agent reaches the see-evil-god ending 18 times. This can be explained by the fact that reaching the alternative ending requires more character interaction, as well as going back and forth between places such as the park and the bar in order to get the bottle of alcohol and the amulet requested by the bum and get him to reveal the secrets.

4 CONCLUSIONS AND FUTURE WORK

We have designed a BDI Player Model for the Interactive Fiction Anchorhead, showing that modelling a human player's interaction with an IN needs a different way to model goals, depending on the current information discovered by the player or agent. We have succeeded in the task of generating a human-looking game trace using said Player Model, as shown in section 3.

As future work, we propose to evaluate our PM comparing it to a set of traces collected by real players. A second step is to analyse factors that determine each player's actions to enhance our current BDI PM and investigate whether it can generate more player-specific traces by making use of such factors.

REFERENCES

[1] M Bratman, *Intention, plans, and practical reason*, Harvard University Press, 1987.
[2] M.J. Nelson, M. Mateas, D.L. Roberts, and C.L. Isbell, 'Declarative optimization-based drama management in interactive fiction', *IEEE Computer Graphics and Applications*, **26**(3), 32–41, (may 2006).
[3] Emma Norling and Liz Sonenberg, 'Creating Interactive Characters with BDI Agents', *Australian Workshop on Interactive Entertainment*, 8, (2004).
[4] Lin Padgham and Michael Winikoff, *Developing intelligent agent systems: A practical guide*, volume 13, Wiley. com, 2005.
[5] Mark O Riedl and Vadim Bulitko, 'Interactive Narrative: An Intelligent Systems Approach.', *AI Magazine*, **34**(1), 67–77, (2013).
[6] Manu Sharma, Santiago Ontanon, Manish Mehta, and Ashwin Ram, 'Drama management and player modeling for interactive fiction games', *Computational Intelligence*, **26**(2), 183–211, (2010).
[7] David Thue, Vadim Bulitko, Marcia Spetch, and Eric Wasylishen, 'Interactive Storytelling : A Player Modelling Approach', in *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference*, eds., Jonathan Schaeffer and Michael Mateas, volume Associatio of *Lecture Notes in Computer Science*, pp. 43–48. AAAI Press, (2007).
[8] Georgios N. Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André, 'Player Modeling', *Dagstuhl Follow-Ups*, **6**, 59, (2013).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1650

A Typicality-Based Revision to Handle Exceptions in Description Logics

Roberto Micalizio and **Gian Luca Pozzato**[1]

Abstract. We propose a methodology to revise a Description Logic knowledge base when detecting exceptions. Our approach relies on the methodology for debugging a Description Logic terminology, addressing the problem of diagnosing inconsistent ontologies by identifying a minimal subset of axioms responsible for an inconsistency. In the approach we propose, once the source of the inconsistency has been localized, the identified axioms are revised in order to obtain a consistent knowledge base including the detected exception about an individual x. To this aim, we make use of a nonmonotonic extension of the Description Logic $\mathcal{ALC}$ based on the combination of a typicality operator and the well established nonmonotonic mechanism of rational closure, which allows to deal with prototypical properties and defeasible inheritance.

1 INTRODUCTION

We focus on the problem of revising a Description Logic (DL for short) knowledge base when detecting an exception. We propose a methodology whose aim is to tackle the problem of revising a TBox in order to accommodate a newly received information about an exception represented by an ABox individual x. Our approach is inspired by the weakening-based revision introduced in [6] and relies on the methodology by Schlobach et al. [8, 7] for detecting exceptions by identifying a minimal subset of axioms responsible for an inconsistency. Once the source of the inconsistency has been localized, the identified axioms are revised in order to obtain a consistent knowledge base including the detected exception about the individual x. To this aim, we use a nonmonotonic extension of the DL $\mathcal{ALC}$ recently presented by Giordano and colleagues in [2]. This extension is based on the introduction of a typicality operator $\mathbf{T}$ in order to express typicality inclusions. The intuitive idea is to allow concepts of the form $\mathbf{T}(C)$, whose intuitive meaning is that $\mathbf{T}(C)$ selects the *typical* instances of a concept C. For instance, a knowledge base can consistently express that birds normally fly ($\mathbf{T}(Bird) \sqsubseteq Fly$), but penguins are exceptional birds that do not fly ($Penguin \sqsubseteq Bird$ and $Penguin \sqsubseteq \neg Fly$). The $\mathbf{T}$ operator is intended to enjoy the well-established properties of *rational logic*, introduced by Lehmann and Magidor in [4] for propositional logic. In order to reason about prototypical properties and defeasible inheritance, the semantics of this nonmonotonic DL, called $\mathcal{ALC}^{\mathbf{R}}_{min}\mathbf{T}$, is based on rational models and exploits a minimal models mechanism based on the minimization of the rank of the domain elements. This semantics corresponds to a natural extension to DLs of Lehmann and Magidor's notion of *rational closure* [4].

[1] Dipartimento di Informatica, Università di Torino, Italia, email: {roberto.micalizio,gianluca.pozzato}@unito.it

Given a consistent knowledge base $K = (\mathcal{T}, \mathcal{A})$ and a consistent ABox $\mathcal{A}' = \{D_1(x), D_2(x), \ldots, D_n(x)\}$, such that $(\mathcal{T}, \mathcal{A} \cup \mathcal{A}')$ is inconsistent, we define a typicality-based revision of $\mathcal{T}$ in order to replace some inclusions $C \sqsubseteq D$ in $\mathcal{T}$ with $\mathbf{T}(C) \sqsubseteq D$, resulting in a new TBox $\mathcal{T}^{new}$ such that $(\mathcal{T}^{new}, \mathcal{A} \cup \mathcal{A}')$ is consistent in $\mathcal{ALC}^{\mathbf{R}}_{min}\mathbf{T}$ and that $\mathcal{T}^{new}$ captures a notion of *minimal changes*.

2 DESCRIPTION LOGICS AND EXCEPTIONS

The logic $\mathcal{ALC}+\mathbf{T}_{\mathbf{R}}$ is obtained by adding to standard $\mathcal{ALC}$ the typicality operator $\mathbf{T}$ [2]. The intuitive idea is that $\mathbf{T}(C)$ selects the *typical* instances of a concept C. We can therefore distinguish between the properties that hold for all instances of concept C ($C \sqsubseteq D$), and those that only hold for the normal or typical instances of C ($\mathbf{T}(C) \sqsubseteq D$). The semantics of the $\mathbf{T}$ operator can be formulated in terms of *rational models*: a model $\mathcal{M}$ is any structure $\langle \Delta^{\mathcal{I}}, <, .^{\mathcal{I}} \rangle$ where $\Delta^{\mathcal{I}}$ is the domain, $<$ is an irreflexive, transitive, well-founded and modular (for all x, y, z in $\Delta^{\mathcal{I}}$, if $x < y$ then either $x < z$ or $z < y$) relation over $\Delta^{\mathcal{I}}$. In this respect, $x < y$ means that x is "more normal" than y, and that the typical members of a concept C are the minimal elements of C with respect to this relation. An element $x \in \Delta^{\mathcal{I}}$ is a *typical instance* of some concept C if $x \in C^{\mathcal{I}}$ and there is no C-element in $\Delta^{\mathcal{I}}$ *more typical* than x. In detail, $.^{\mathcal{I}}$ is the extension function that maps each concept C to $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each role R to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. For concepts of $\mathcal{ALC}$, $C^{\mathcal{I}}$ is defined as usual. For the $\mathbf{T}$ operator, we have $(\mathbf{T}(C))^{\mathcal{I}} = Min_{<}(C^{\mathcal{I}})$. A model $\mathcal{M}$ can be equivalently defined by postulating the existence of a function $k_{\mathcal{M}} : \Delta^{\mathcal{I}} \longmapsto \mathbb{N}$, where $k_{\mathcal{M}}$ assigns a finite rank to each world: the rank function $k_{\mathcal{M}}$ and $<$ can be defined from each other by letting $x < y$ if and only if $k_{\mathcal{M}}(x) < k_{\mathcal{M}}(y)$.

Given standard definitions of satisfiability of a KB in a model, we define a notion of entailment in $\mathcal{ALC}+\mathbf{T}_{\mathbf{R}}$. Given a query F (either an inclusion $C \sqsubseteq D$, or an assertion $C(a)$, or an assertion of the form $R(a, b)$), we say that F is entailed from a KB in $\mathcal{ALC}+\mathbf{T}_{\mathbf{R}}$ if F holds in all models satisfying KB.

Even if the typicality operator $\mathbf{T}$ itself is nonmonotonic (i.e. $\mathbf{T}(C) \sqsubseteq E$ does not imply $\mathbf{T}(C \sqcap D) \sqsubseteq E$), what is inferred from a KB can still be inferred from any KB' with KB $\subseteq$ KB', i.e. the logic $\mathcal{ALC}+\mathbf{T}_{\mathbf{R}}$ is monotonic. In order to perform useful nonmonotonic inferences, in [2] the authors have strengthened the above semantics by restricting entailment to a class of minimal models. Intuitively, the idea is to restrict entailment to models that *minimize the untypical instances of a concept*. The resulting logic is called $\mathcal{ALC}^{\mathbf{R}}_{min}\mathbf{T}$, and it corresponds to a notion of *rational closure* on top of $\mathcal{ALC}+\mathbf{T}_{\mathbf{R}}$. Such a notion is a natural extension of the rational closure construction provided in [4] for the propositional logic.

The nonmonotonic semantics of $\mathcal{ALC}^{\mathbf{R}}_{min}\mathbf{T}$ relies on minimal ra-

tional models that minimize the *rank of domain elements*. Informally, given two models of KB, one in which a given domain element x has rank 2 (because for instance $z < y < x$), and another in which it has rank 1 (because only $y < x$), we prefer the latter, as in this model the element x is assumed to be "more typical" than in the former. Query entailment is then restricted to minimal *canonical models*. The intuition is that a canonical model contains all the individuals that enjoy properties that are consistent with the knowledge base. This is needed when reasoning about the rank of the concepts: it is important to have them all represented. A model $\mathcal{M}$ is a minimal canonical model of KB if it satisfies KB, it is minimal and it is canonical[2]. Finally, a query F is minimally entailed from a KB (or, equivalently, F belongs to the rational closure of KB) if it holds in all minimal canonical models of KB. In [2] it is shown that minimal entailment in $\mathcal{ALC}^{\mathbf{R}}_{min}\mathbf{T}$ is in ExpTime.

3 TYPICALITY-BASED REVISION OF A KNOWLEDGE BASE

Similarly to what done in [6], we define a notion of revised knowledge base, precisely a *typicality-based* revised knowledge base. Given a consistent knowledge base $K = (\mathcal{T}, \mathcal{A})$, we have to tackle the problem of accommodating a further ABox information $\mathcal{A}'$, describing an individual x that belongs to the extensions of the concepts $D_1, D_2, \ldots, D_n$, and that is an *exception* to the knowledge described by K; namely, given a consistent ABox $\mathcal{A}' = \{D_1(x), D_2(x), \ldots, D_n(x)\}$, we have that the knowledge base $(\mathcal{T}, \mathcal{A} \cup \mathcal{A}')$ is inconsistent.

We revise the TBox $\mathcal{T}$ of K by replacing some standard inclusions $C \sqsubseteq D$ with typicality inclusions $\mathbf{T}(C) \sqsubseteq D$, in a way such that the resulting revised knowledge base is consistent in $\mathcal{ALC}^{\mathbf{R}}_{min}\mathbf{T}$. Given $K = (\mathcal{T}, \mathcal{A})$, we denote with $Rev_{\mathbf{T},\mathcal{A}'}(K)$ the set of all typicality-based weakenings of $\mathcal{T}$ given a newly received ABox $\mathcal{A}'$. Among all typicality-based weakenings in $Rev_{\mathbf{T},\mathcal{A}'}(K)$ we select the one, called $\mathcal{T}^{new}$, capturing a notion of *minimal changes* needed in order to accommodate the discovered exception. The following example, inspired by the well known problem of the Nixon diamond, shows how we revise a knowledge base after discovering an exception.

Example 1 *Let $K = (\mathcal{T}, \emptyset)$ where $\mathcal{T}$ is:*

$$Quacker \sqsubseteq Christian \quad (1)$$
$$Christian \sqsubseteq Pacifist \quad (2)$$
$$RepublicanPresident \sqsubseteq Republican \quad (3)$$
$$Republican \sqsubseteq \neg Pacifist \quad (4)$$

and let $\mathcal{A}' = \{Quacker(nixon), RepublicanPresident(nixon)\}$.

It can be shown that there are eleven different typicality-based weakenings in $Rev_{\mathbf{T},\mathcal{A}'}(K)$, but the one chosen as the typicality-based revision $\mathcal{T}^{new}$ is as follows:

$$Quacker \sqsubseteq Christian$$
$$\mathbf{T}(Christian) \sqsubseteq Pacifist$$
$$RepublicanPresident \sqsubseteq Republican$$
$$\mathbf{T}(Republican) \sqsubseteq \neg Pacifist$$

The resulting knowledge base $K^{new} = (\mathcal{T}^{new}, \mathcal{A} \cup \mathcal{A}')$ is consistent.

[2] In Theorem 10 in [2] the authors have shown that for any KB there exists a finite minimal canonical model of KB.

4 COMPUTING A REVISED TBOX

We have introduced an algorithm that revises a given knowledge base $K = (\mathcal{T}, \mathcal{A})$ according to the typicality-based weakening outlined in the previous section. Our algorithm relies on the computation of a Minimal Unsatisfiability-Preserving Sub-TBoxes (*mups*), introduced by Schlobach et al. in their seminal work [8] about the problem of debugging a DL terminology, that singles out the subset of inclusions strictly involved in the inconsistency.

Definition 1 (MUPS, Definition 3.1 [8]) *Let C be a concept which is unsatisfiable in a TBox $\mathcal{T}$. A set $\mathcal{T}' \subseteq \mathcal{T}$ is a minimal unsatisfiability-preserving sub-TBox (mups) of $\mathcal{T}$ if C is unsatisfiable in $\mathcal{T}'$, and C is satisfiable in every sub-TBox $\mathcal{T}'' \subset \mathcal{T}'$.*

Following [8], we have restricted our approach to *unfoldable* TBoxes, only containing unique, acyclic definitions. An axiom is called a definition of A if it is of the form $A \sqsubseteq C$, where A is an atomic concept. An axiom $A \sqsubseteq C$ is unique if the KB contains no other definition of A. An axiom is acyclic if C does not refer either directly or indirectly (via other axioms) to A [1].

5 FUTURE ISSUES

We aim at extending our typicality-based revision also to not *unfoldable* TBoxes: to this aim, in [5, 3] axiom pinpointing is extended to general TBoxes. A set of algorithms for computing axiom pinpointing, in particular to compute the set of *mups* for a given terminology $\mathcal{T}$ and a concept A, is also provided. Furthermore, we intend to develop an implementation of the proposed algorithms, by considering the integration with existing tools for manually modifying ontologies when inconsistencies are detected.

Acknowledgements

The authors are partially supported by the projects "AThOS: Accountable Trustworthy Organizations and Systems" and "ExceptionOWL: Nonmonotonic Extensions of Description Logics and OWL for defeasible inheritance with exceptions" , Progetti di Ateneo Università degli Studi di Torino and Compagnia di San Paolo, call 2014, line "Excellent (young) PI".

REFERENCES

[1] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, *The Description Logic Handbook - Theory, Implementation, and Applications, 2nd edition*, Cambridge, 2007.
[2] L. Giordano, V. Gliozzi, N. Olivetti, and G.L. Pozzato, 'Semantic characterization of rational closure: From propositional logic to description logics', *Artificial Intelligence*, **226**, 1–33, (2015).
[3] A. Kalyanpur, B. Parsia, B. Cuenca-Grau, and E. Sirin, 'Axiom pinpointing: Finding (precise) justifications for arbitrary entailments in $\mathcal{SHOIN}$ (owl-dl)', *Technical report, UMIACS, 2005-66*, (2006).
[4] D. Lehmann and M. Magidor, 'What does a conditional knowledge base entail?', *Artificial Intelligence*, **55**(1), 1–60, (1992).
[5] B. Parsia, E. Sirin, and A. Kalyanpur, 'Debugging owl ontologies', in *WWW*, pp. 633–640, (2005).
[6] G. Qi, W. Liu, and D. A. Bell, 'A revision-based approach to handling inconsistency in description logics', *Artificial Intelligence Review*, **26**(1-2), 115–128, (2006).
[7] S. Schlobach, 'Diagnosing terminologies', in *Proceedings of AAAI*, pp. 670–675, (2005).
[8] S. Schlobach and R. Cornet, 'Non-standard reasoning services for the debugging of description logic terminologies', in *IJCAI*, pp. 355–362, (2003).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1652

A New Stochastic Local Search Approach for Computing Preferred Extensions of Abstract Argumentation

Dangdang Niu[1] and **Lei Liu**[1] and **Shuai Lü**[1,2,3+]

Abstract. In this paper, we proposed a new stochastic local search algorithm Inc-CCA$_{EP}$ for computing the preferred extensions in (abstract) argumentation frameworks (AF). Inc-CCA$_{EP}$ realizes an incremental version of Swcca, specially designed for computing the preferred extensions in AF. Experiments show that, Inc-CCA$_{EP}$ notably outperforms the state-of-the-art solvers consistently on random benchmarks with non-empty preferred extensions.

1 Introduction

Dung's theory of (abstract) argumentation frameworks (AF) provides a general model for computational argumentation [1]. For preferred semantics in AF, CEGARTIX [2] and ArgSemSAT [3] are two representative SAT-based argumentation systems, which both rely on iteratively calling to complete SAT solvers. Each of above two systems is ranked 1st or 2nd in the SE-PR, DS-PR and EE-PR tracks of the first International Competition on Computational Models of Argumentation (ICCMA'15).

There are two popular kinds of algorithms for SAT: conflict driven clause learning, and stochastic local search (SLS). The efficiency of SLS algorithms mostly depend on the heuristic methods selected by them. An efficient heuristic method named configuration checking (CC) [4] has been proposed. Then Swcca (smoothed weighting and configuration with aspiration) [5] is designed based on the heuristics named configuration checking with aspiration (CCA) which is an improvement of CC.

Only complete SAT solvers have been exploited for computing preferred extensions in AF so far. It is a natural question that how the appealing SLS approaches could advance the performance. The aim of this paper is to answer above question. And a novel approach Inc-CCA$_{EP}$ is proposed based on the SLS algorithm of SAT. This paper is organized as follows. Section 2 recalls the basic concepts of AF and CCA. Section 3 introduces our SLS algorithms for computing preferred extensions, while Section 4 describes the test setting and comments the experimental results. Section 5 concludes the paper.

2 Preliminaries

An AF is a pair $F = (A, R)$ where A is a set of arguments and $R \subseteq A \times A$ is the attack relation. An extension $S \subseteq A$ is *conflict–free* iff $\nexists a, b \in S$ s.t. $a \rightarrow b$. An argument $a \in A$ is *acceptable* with respect to a set $S \subseteq A$ iff $\forall b \in A$ s.t. $b \rightarrow a$, $\exists c \in S$ s.t. $c \rightarrow b$. $S \subseteq A$ is *admissible* iff S is *conflict–free* and every element of S is *acceptable* with respect to S. $S \subseteq A$ is an *admissible extension* iff S is *admissible*. $S \subseteq A$ is a *preferred extension* iff S is a maximal *admissible extension*.

Given an AF $F = (A, R)$, a key problem of instantiating the SAT-based framework is how the AF reasoning tasks are encoded as CNF formulae. Let $\varphi_{adm}(F)$ be the CNF formula which corresponds to the expression of admissible semantics of F, the set of all variables appear in $\varphi_{adm}(F)$ is in correspondence with A. $\varphi_{adm}(F)$ is shown as follows [3]:

$$\varphi_{adm}(F) = \bigwedge_{(b,a)\in R}((\neg v_a \vee \neg v_b) \wedge (\neg v_a \vee \bigvee_{(c,b)\in R} v_c)) \quad (1)$$

Given a model P, let $\alpha(P)$ be the set of positive literals in P. Then, a model P of $\varphi_{adm}(F)$ is in correspondence with a preferred extension of F iff there is no model Q with $\alpha(P) \subset \alpha(Q)$.

Let $V(F)$ be the set of all variables appear in the CNF formula F. $N(x) = \{y \mid y \in V(F)$ and y occurs in at least one clause with $x\}$ is the set of all neighboring variables of a variable x. Then the configuration of a variable $v \in V(F)$ is a vector C_v consisting of truth values of all variables in $N(v)$ under the current assignment s.

In the implementation of CCA, each clause $c \in \Phi$ is associated with a positive integer number $w(c)$ as its weight, in which w is a weighted formula. $cost(\Phi, s)$ denotes the total weight of all unsatisfied clauses under the assignment s. Let $score(v) = cost(F, s) - cost(F, s')$, measuring the benefit of flipping v, where s' is obtained from s by flipping v. Any element in the array *confChange* is an indicator for a variable. $confChange[v] = 1$ means the configuration of variable v has been changed since v's last flip; and $confChange[v] = 0$ on the contrary.

CCA heuristics: A configuration changed decreasing (CCD) variable v is a variable with both $confChange[v] = 1$ and $score(v) > 0$. A significant decreasing (SD) variable v is a variable with $score(v) > g$, where g is a positive integer large enough. In the literature [5], g is set to the averaged clause weight (over all clauses). CCA selects CCD variable with greatest score to flip firstly. The SD variable with the greatest score is selected to flip if there are no CCD variables. If there are neither CCD variables nor SD variables, CCA switches to diversification mode [5].

3 Inc-CCA$_{EP}$

We are now in a position to introduce our proposed procedure Inc-CCA$_{EP}$ which is listed in Algorithm 1.

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China, email: ddniu15@mails.jlu.edu.cn, {liulei, lus}@jlu.edu.cn.
[2] College of Mathematics, Jilin University, Changchun 130012, China.
[3] Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun 130012, China.

Alogrithm 1 Inc-CCA$_{EP}$

1: **Input**: $F = (A, R)$
2: **output**: $E_p \subseteq 2^A$
3: $E_p = \emptyset$
4: $\Phi_0 = \varphi_{adm}(F)$
5: (Φ, M) = unit-propagation(Φ_0)
6: $s \leftarrow$ randomly generated truth assignment
7: initialize heuristic information
8: **for** *step* $\leftarrow$ 1 **to** *maxsteps* **do**
9: **if** s satisfies Φ
10: **while** $\exists \neg v_i \in s$ & $s|_{\text{flip}(vi)}$ satisfies Φ
11: $s = s|_{\text{flip}(vi)}$
12: update heuristic information
13: $E_p = E_p \cup \{s\}$
14: $\Phi = \Phi \wedge \bigvee_{\neg vi \in s} v_i$
15: add heuristic information about $\bigvee_{\neg vi \in s} v_i$
16: v = CCA()
17: $s = s|_{\text{flip}(v)}$
18: update heuristic information
19: E_p = merge(E_p, M)
20: **return** E_p

The main idea behind Inc-CCA$_{EP}$ is that it intends to compute as many as possible admissible extensions based on $\varphi_{adm}(F)$ and then finding all preferred extensions. In order to efficiently compute preferred extensions, we design two strategies in Algorithm 1:

Strategy 1. There may be some unit clauses in $\varphi_{adm}(F)$. So we use unit propagation to reduce the scale of $\varphi_{adm}(F)$ in Line 5.

Strategy 2. After Swcca searches a model T, if T is still a model after flipping some negative literals in it, we flip them in Line 10-11. This strategy can reduce the calling times of Swcca.

In Algorithm 1, the heuristic information appeared in Inc-CCA$_{EP}$ mainly concludes the scores of all variables, the weights of all clauses and the flipping time stamps of all variables [5]. Since Inc-CCA$_{EP}$ is an incomplete algorithm, it does not know whether all possible preferred extensions have been found or not. So, we use *maxsteps* to control the end of Inc-CCA$_{EP}$ in Line 8. In order to avoid searching an admissible extension which is the subset of any admissible extension searched before, we add $\bigvee_{\neg vi \in s} v_i$ to Φ in Line 14. It employs CCA heuristics to pick flipping variable in Line 16. In Line 19, merge function is used to add all literals in M to each model in E_p and then find all possible preferred extensions.

4 Experimental results

Our experiments are conducted on the PC with a quad-core Intel(R) Core(TM) i7-3700, 8GByte RAM and Ubuntu 14.04 operating system. Because Inc-CCA$_{EP}$ cannot finish their search processes automatically even if they have found all preferred extensions, we use actually searching time to measure their efficiencies. We do the experiments on randomly generated AFs by *probo*[4] which is used in ICCMA'15. We input two parameters $|A|$ and p to *probo*. $|A|$ is the number of arguments, and p is the probability that there is an attack for each ordered pair of arguments (self-attacks are include). CEGARTIX[5] and ArgSemSAT[6] are two alternative AF reasoners in our experiments.

[4] https://sourceforge.net/p/probo/code/HEAD/tree/trunk/doc/
[5] http://www.dbai.tuwien.ac.at/research/project/argumentation/cegartix/
[6] http://sourceforge.net/projects/argsemsat/

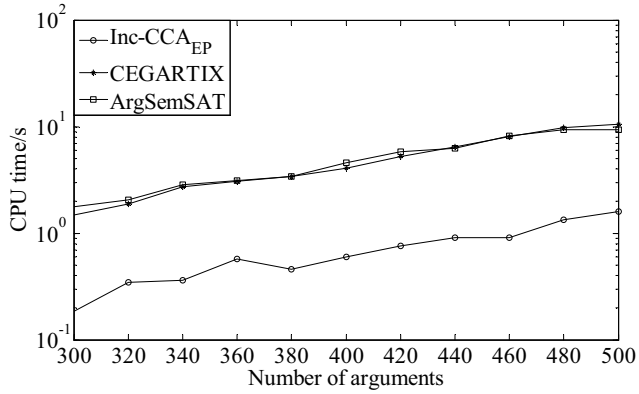


Figure 1. Experiments on random AF instances with fixing $p = 0.5$.

We generate AF instances with fixing $p = 0.5$, and $|A|$ is ranging from 360 to 500 with a step of 20. All instances have non-empty preferred extensions. The test results are given in Figure 1.

In Figure 1, we can see that the actual searching time of Inc-CCA$_{EP}$ is far less than the executing time of CEGARTIX and ArgSemSAT. When fixing $p = 0.5$, the efficiency of Inc-CCA$_{EP}$ is only about a tenth of the efficiency of CEGARTIX or ArgSemSAT. ArgSemSAT is a little better than CEGARTIX. The reason is that CEGARTIX is more suitable for hard AF instances.

5 Conclusions

We innovatively exploit stochastic local search for enumerating preferred extensions in abstract argumentation in this paper. And Inc-CCA$_{EP}$ is proposed based on Swcca for enumerating preferred extensions in abstract argumentation. Experimental results show that Inc-CCA$_{EP}$ significantly outperform existing systems on random AF instances with non-empty preferred extensions.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant No. 61300049; the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20120061120059; and the Natural Science Research Foundation of Jilin Province of China under Grant Nos. 20140520069JH, 20150101054JC.

REFERENCES

[1] P. M. 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).
[2] W. Dvořák, M. Järvisalo, J. P. Wallner and S. Woltran, 'Complexity-sensitive decision procedures for abstract argumentation', *Artificial Intelligence*, 206, 53-78, (2014).
[3] F. Cerutti, M. Giacomin and M. Vallati, 'ArgSemSAT: Solving argumentation problems using SAT', in *Proc. 5th International Conference on Computational Models of Argument (COMMA)*, pp. 455-456, Scottish Highlands, UK, (2014).
[4] S. Cai, K. Su and A. Sattar, 'Local search with edge weighting and configuration checking heuristics for minimum vertex cover', *Artificial Intelligence*, 175(9-10), 1672-1696, (2011).
[5] S. Cai and K. Su, 'Configuration checking with aspiration in local search for SAT', in *Proc. 26th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 434-440, Toronto, Ontario, Canada, (2012).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1654

Crowdsourced Referral Auctions

Praphul Chandra [1] and **Sujit Gujar**[2] and **Y Narahari**[3]

Abstract. Motivated by web based marketplaces where the number of bidders in an auction is a small subset of potential bidders, we consider auctions where the auctioneer (seller) wishes to increase her revenue and/or social welfare by expanding the pool of participants. To this end, the seller crowdsources this task by offering a *referral bonus* to the participants. With the introduction of referrals, a participant can now bid and/or refer other agents to bid. We call our auctions *crowdsourced referral auctions* since the seller exploits the knowledge that agents have about other potential participants in the crowd. We introduce the notion of *price of locality* to quantify the loss in social welfare due to restricted (local) access of the seller to potential bidders. We introduce the notion of Crowdsourced Referral Auction Mechanisms (CRAMs), propose two novel versions of CRAMs and study the induced *referral game* in the canonical context of an auction for selling a single indivisible item. We compare their revenue performance and game theoretic properties and show that both of them outperform the baseline auction without referrals.

1 INTRODUCTION

We propose the use of auctions where the seller crowdsources the job of increasing participation in the auction. Azar et. al. [2] propose that the seller crowdsource the task of gathering information about bidders' distribution over valuations. We use crowdsourcing with a different motivation: since the seller has access to only a subset of potential bidders, we propose the seller crowdsources the job of *increasing participation* in the auction by offering a referral bonus.

Several web based marketplaces rely on auction based mechanisms. Since the social welfare (sum of utilities of all agents) as well as the seller's revenue increase with increased competition (number of bids received) [3], many marketplaces support sellers to promote their auction in their social networks in order to expand the pool of bidders. Some marketplaces [1] encourage agents other than the seller, to promote items by posting URLs to the item on their (micro-)blogs. A purchase made through a referral URL leads to a bonus for the referring agent. Our current work may be seen as a systematic and structured generalization of this model to an auction setting where *every* bidder is treated as a potential referrer.

In CRAMs, a participant who refers other potential participants may himself be a bidder. This is a *dilemma* for each participant with two possible options: (1) *to refer* other agents at the risk of potentially reduced probability of winning and potentially reduced utility from winning in the auction, or (2) *not to refer* other agents to keep the competition low but forgo the possibility of receiving a referral bonus in the case of losing in the auction.

[1] Indian Institute of Science, Bangalore, India, email: praphulcs@gmail.com
[2] International Institute of Information Technology, Hyderabad, India, email: sujit.gujar@iiit.ac.in
[3] Indian Institute of Science, Bangalore, India, email: hari@csa.iisc.ernet.in

We consider a single (indivisible) item, forward auction setting. A participant's private information consists of his true value for the item (θ_i). In addition, a participant can choose to refer additional agents. Correspondingly, a participant's strategy (ψ_i) consists of his bid (x_i) and his referral strategy: 0 (Don't refer) or 1 (Refer), thus $\psi_i = (x_i, 0/1)$. Agent id $i = 0$ refers to the seller and $\theta_0 = 0$. The set of agents (excluding the seller) is N with $n = |N| - 1$. The vector of agent values is $\theta = (\theta_1, ..., \theta_n) \in \mathbb{R}^n$. Throughout this paper, the subscript $-i$ is used to refer to values excluding agent i, e.g., θ_{-i} refers to a vector of types excluding θ_i. Agent types are i.i.d. (independent, identically distributed) random variables drawn from a distribution $F()$. So, the maximum valuation in θ is $\theta_{[n,n]} \sim F^n(.)$ and the second highest valuation in θ is $\theta_{[n-1,n]} \sim n(1 - F(.))F^{n-1}(.)$. The set of agents who bid in the mechanism is S with $|S| = s \leq n$ and the vector of bids is $\mathbf{x} = (x_1, ..., x_s) \in \mathbb{R}^s$. Agents are nodes in a network and an edge between two nodes means that the two nodes can interact with each other. The set of agents referred by agent i (M_i with $m_i = |M_i|$) are its children. Each node has exactly one parent: if more than one agent refers the same agent, the earliest referral takes precedence. Thus, the set of agents who bid, form a tree with the seller at the root.

Definition 1. *(Price of Locality) In a population of n agents, the price of locality of a mechanism Z which leads to $m \leq n$ bids is the ratio of the expected efficiency achieved when all n agents bid to the expected efficiency achieved when m agents bid: $PoL_Z = \frac{\mathbb{E}_\theta[\theta_{n,n}]}{\mathbb{E}_\theta[\theta_{m,m}]}$.*

2 CRAM

A Crowdsourced Referral Auction Mechanism (CRAM), Z, consists of an allocation rule, $\mathbf{k} : \mathbb{R}^s \to \{0,1\}^s$, which decides which agent to allocate the item to, a payment rule, $\mathbf{t} : \mathbb{R}^s \to \mathbb{R}^s$, which determines the monetary amount to be charged to agents given the bids ($\mathbf{x} \in \mathbb{R}^s$) received from the agents, and a referral bonus rule $\mathbf{b} : \mathbb{R}^s \to \mathbb{R}^s$ that determines the referral bonus to be paid to agents. We aim to design CRAMs so that agents bid their true valuations and refer additional agents from their social network. For a mechanism Z, the social welfare (efficiency) is η_Z and the seller's revenue is γ_Z. With a quasi-linear environment, agent value and utility are:

$$\begin{aligned} v_i(\psi_i, \psi_{-i}; \theta_i) &= \theta_i k_i(\psi) = v_i(\psi_i, \mathbf{k}(\psi)) \\ u_i(\psi_i, \psi_{-i}; \theta_i) &= \theta_i k_i(\psi) - t_i(\psi) + b_i(\psi) \end{aligned}$$

2.1 REVA1

In REVA1, agents can bid and refer others to bid. The set of bidders (M), is the seller's immediate neighbors *and the referred agents*.

With, $j^* = \arg\max_{i \in M} x_i$, REVA1 is specified as:

$$t_{j^*}(\psi) = \sum_{j \in M, j \neq j^*} v_j(\theta_j, \mathbf{k}^*_{-\mathbf{j}^*}(\psi)) - \sum_{j \in M, j \neq j^*} v_j(\theta_j, \mathbf{k}^*(\psi))$$
$$b_{j^*}(\psi) = 0$$

And $\forall i \neq j^*$:

$$t_i(\psi) = 0$$
$$b_i(\psi) = \sum_{j \in M, j \neq i} v_j(\theta_j, \mathbf{k}^*(\psi)) - \sum_{j \in M, j \neq i} v_j(\theta_j, \mathbf{k}^*_{-\mathbf{i}}(\psi))$$

where $\mathbf{k}^*$ is the social welfare maximizing allocation and $\mathbf{k}^*{}_{-i}$ is the social welfare maximizing allocation in absence of i *and* his descendants in a referral tree. REVA1 allocates the item to the bidder with the highest value and the winning agent (j^*) pays the second highest bid from the set of bids that would have been received without him *and his referrals*. REVA1 incentivizes referrals by a payment rule where agent(s) who referred the winning agent are paid a referral bonus which is calculated as a VCG payment. Even though REVA1 uses payment and bonus rules similar to VCG, it is not a direct mechanism. We can show that REVA1 is EPIR and prove the following:

Theorem 1. *In REVA1, $\forall i\ \psi_i^* = (\theta_i, 1)$, that is, for all agents bidding truthfully and referring is a dominant strategy equilibrium.*

2.2 REVA2

REVA2 uses an allocation rule where the bids of referring agents are not considered when deciding the winner and the payment that the winner must make. Effectively, this partitions the set of agents who *participate* in the auction into two discrete sets: A and D (with $a = |A|$ and $d = |D|$). A is the set of agents who refer additional agents and D is the set of agents who bid and do not refer any agent. In REVA2, the set of bidders is effectively $S = D$. Furthermore, REVA2 uses a referral bonus rule where the referral bonus is fixed to a pre-specified amount $b > 0$ and only the *parent* of the winning node is paid this referral bonus. REVA2 restricts referrals to the seller's immediate neighbors and the maximum height of the referral tree is two. Similar to REVA1, REVA2 is EPIR. With, $j^* = \arg\max_{i \in D} x_i$, REVA2 is specified as:

$$t_{j^*}(\psi) = \sum_{j \in D, j \neq j^*} v_j(\theta_j, \mathbf{k}^*_{-\mathbf{j}^*}(\psi)) - \sum_{j \in D, j \neq j^*} v_j(\theta_j, \mathbf{k}^*(\psi))$$
$$b_{j^*}(\psi) = 0$$

$$\text{And} \quad \forall i \neq j^* \quad t_i(\psi) = 0$$
$$b_i(\psi) = -b$$

Thus, the net payment from agent i is:

$$r_i(\psi) = \begin{cases} x_{[d-1,d]} - 0 & \text{if } i = j^* \\ b & \text{if } j^* \in M_i \\ 0 & \text{otherwise} \end{cases}$$

Let d^i be the number of agents who bid if i bids (and does not refer). The probability that agent i will win the auction if he bids x_i given that d^i agents bid is $G_i(x_i|d^i) = F^{d^i}(x_i)$. Since, agent i does not know d^i, his belief that he will win the auction if he bids x_i is $G_i(x_i) = \sum_{j=1}^{\infty} F^{d^i}(x_i)\mathbb{P}(d^i = j)$. Similarly, let $\beta_i(x_i)$ be agent i's expected second highest bid given that he wins by bidding x_i; so, $\beta_i(x_i) = \sum_{j=1}^{\infty} \mathbb{E}_{\theta_{-i}}[\theta_{[d^i-1,d^i]}|\theta_{[d^i,d^i]} = x_i; d^i = j]\mathbb{P}(d^i = j)$. Let $\bar{d}^{-i}$ be the expected number of agents who bid in the auction when i refers. With this notation, we can prove the following:

Theorem 2. *Let $b > 0$. In REVA2, the following is a Bayesian Nash equilibrium:*

$$\psi_i^* = \begin{cases} (\theta_i, 0) & \text{if } \theta_i > \alpha_i \\ (\theta_i, 1) & \text{otherwise} \end{cases}$$

where α_i is the smallest non-negative root of:

$$(\alpha - \beta_i(\alpha))\,(G_i(\alpha)) - \frac{m_i}{\bar{d}^{-i}} b = 0$$

3 Discussion

Our simulations show that, when compared to the baseline VCG auction without referrals, both REVA1 and REVA2 achieve higher social welfare. REVA1 achieves the highest social welfare since it is a dominant strategy for agents to refer and this leads to a maximal expansion of the set of bidders. In REVA2, since the bids of agents who refer are not considered when determining the auction winner, the efficiency achieved by REVA2 is lower than that achieved by REVA1. For any network $\text{PoL}_{\text{VCG}} \geq \text{PoL}_{\text{REVA2}} \geq \text{PoL}_{\text{REVA1}} = 1$ where the equalities hold in a star network with the seller at the center. The results for revenue are more nuanced. We identify the

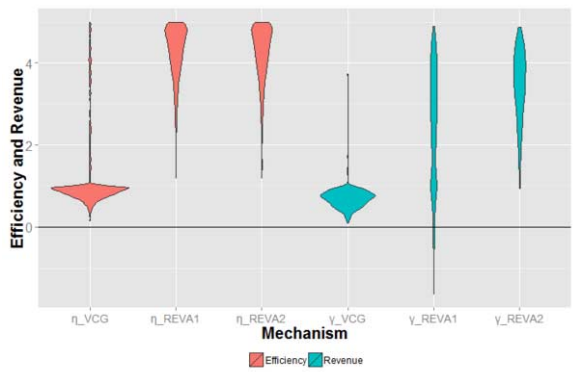


Figure 1. Comparison of VCG, REVA1, REVA2 for efficiency and revenue with a referral tree of height 2; 5% high value agents.

following cases when REVA1 may yield negative revenue: (i) when there are few high value agents, ($< 5\%$ in our simulations), (ii) when seller's access to high value agents or a large subset of the potential bidders is controlled by a few agents. In such cases, REVA2 proves a good alternative to REVA1 since it bounds the loss in revenue and yet achieves an expanded pool of bidders. The number of agents who bid in REVA2 at Bayesian Nash equilibrium (BNE) depends on the referral bonus, b and the number of agents which the seller has direct access too. When the seller can directly sell to a very few agents, a high referral bonus is needed to ensure that a given agent refers at the BNE. As the number of direct access agents increases, even a small referral bonus is sufficient.

REFERENCES

[1] amazon.com. What is the Amazon Associates program?, 2015. Available online at `https://affiliate-program.amazon.com/gp/associates/join/getstarted`.

[2] Pablo Azar, Jing Chen, and Silvio Micali, 'Crowdsourced bayesian auctions', in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 236–248. ACM, (2012).

[3] Jeremy Bulow and Paul Klemperer, 'Auctions versus negotiations', *The American Economic Review*, **86**(1), 180–194, (1996).

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1656

Minisum and Minimax Committee Election Rules for General Preference Types

Dorothea Baumeister[1] and **Toni Böhnlein**[2] and **Lisa Rey**[1] and **Oliver Schaudt**[2] and **Ann-Kathrin Selker**[1]

Abstract. In committee elections it is often assumed that voters only (dis)approve of each candidate or that they rank all candidates, as it is common for single-winner elections. We suggest an intermediate approach, where the voters rank the candidates into a fixed number of groups. This allows more diverse votes than approval votes, but leaves more freedom than in a linear order. A committee is then elected by applying the minisum or minimax approach to minimize the voters' dissatisfaction. We study the axiomatic properties of these committee election rules as well as the complexity of winner determination and show fixed-parameter tractability for our minimax rules.

1 Introduction

A central point in computational social choice is the analysis of voting systems, see for example the book chapter by Zwicker [11]. Whereas the initial focus was mainly on single-winner elections, the study of committee elections recently received considerable attention. In a committee election a winner is a subset of candidates of a predefined size.

Most voting rules require the voters to either rank all candidates in a strict linear order, which might be impossible given a large set of candidates, or to divide them into two groups, i. e., approval ballots, which might be too rough to fully express the voters' preferences. As an intermediate approach, we propose ℓ-ballots. Voters group the candidates into a fixed number of groups, where all candidates in one group are tied. We use this type of ballot – a slight variant of the model proposed by Obraztsova et al. [10]– to define committee election rules that minimize the voters' dissatisfaction and study computational and axiomatic properties of these rules. To that end, we apply the well-known minisum method where the sum of the distances to the individual votes is minimized, and the minimax method where the maximal distance to an individual vote is minimized. Originally, the minisum and minimax methods have been applied to approval votes by Brams et al. [4]. The most relevant papers for our study are those by Baumeister et al. [2, 3] who extended this approach to determine winning committees for different forms of votes, namely trichotomous votes as well as complete and incomplete linear orders. Elkind et al. [6] studied axiomatic properties such as consistency, monotonicity, and solid coalitions for different multiwinner voting rules, including STV, Bloc, k-Borda and different variants of the Chamberlin-Courant and Monroe's rule. We adapt some of these properties to our setting and study them for the class of ℓ-group rules. The parameterized complexity of minimax voting rules has been studied by Misra et al. [9] for approval votes as well as by Liu and Guo [8] for trichotomous votes and linear and partial orders. In both papers it is shown that, for their respective voting rules, winner determination is W[2]-hard when parameterized by the size of the committee and that computing a winning committee is fixed-parameter tractable with respect to a distance parameter.

[1] Heinrich-Heine-Universität Düsseldorf, Germany, email: {baumeister, lrey, selker}@cs.uni-duesseldorf.de
[2] Universität zu Köln, Germany, email: boehnlein@zpr.uni-koeln.de, schaudto@uni-koeln.de

2 Definitions

Let $C = \{c_1, \dots, c_m\}$ be a set of candidates and $V = (v_1, \dots, v_n)$ a profile, i. e., a list of voters represented by their vote. In an *ℓ-ballot* over C, a vote is given as a list of ℓ pairwise disjoint sets of candidates, which may also be empty: $v = (G_1, \dots, G_\ell)$ where $G_i \cap G_j = \emptyset$ for $1 \le i, j \le \ell$ and $i \ne j$, and $\bigcup_{1 \le i \le \ell} G_i = C$. Considering a set of candidates $C = \{c_1, c_2, c_3, c_4\}$, a possible 3-ballot is $(\{c_3, c_4\}, \{\}, \{c_1, c_2\})$ which means that candidates c_3 and c_4 are preferred to all other candidates, and candidates c_1 and c_2 are the most disliked ones.

A very similar ballot model has been introduced by Obraztsova et al. [10]. The predefined ℓ groups correspond to their preference levels. In contrast to our model, they assume that the first and last group are never empty and that at least one voter specifies no empty group. However, these are only technical requirements that are not crucial for our results.

A *committee* is a subset of C. Let $F_k(C)$ denote the set of all committees of size k. A *committee election* is a triple $E = (C, V, k)$, where C is the set of candidates, V is a list of voters, represented by ℓ-ballots for some fixed constant ℓ over C, and $k \in \mathbb{N}$ denotes the committee size. A *committee election rule* $\mathcal{R}$ is a function that, given a committee election, returns a set of tied winning committees.

Now we introduce the ℓ-group voting rules discussed in this paper. For this sake we define $\delta_\ell(v, W) = \sum_{c \in C} |v(c) - W(c)|$ as the *dissatisfaction* (or *distance*) between an ℓ-ballot v and a committee $W \in F_k(C)$ where $W(c) = 1$ for a candidate $c \in W$, and $W(c) = \ell$ for a candidate $c \notin W$, and where $v(c)$ denotes the group number of a candidate c. For the case of $\ell = 2$ this distance corresponds to the Hamming distance between the vote and the committee. The following two rules elect the winning committee(s) for profiles consisting of ℓ-ballots.

Definition 1 (minisum/minimax ℓ-group rule) • *Minisum ℓ-group rules are functions f^ℓ_{sum} so that $f^\ell_{sum}((C, V, k)) = \operatorname{argmin}_{W \in F_k(C)} \sum_{v \in V} \delta_\ell(v, W)$, i. e., f^ℓ_{sum} minimizes the sum of the voters' dissatisfaction to the winning committees.*

• *Minimax ℓ-group rules are functions f^ℓ_{max} so that $f^\ell_{max}((C, V, k)) = \operatorname{argmin}_{W \in F_k(C)} \max_{v \in V} \delta_\ell(v, W)$, i. e., f^ℓ_{max} minimizes the dissatisfaction of the least satisfied voter with the winning committees.*

Note that the minisum/minimax voting rules defined by Baumeister and Dennisen [2] correspond to our minisum/minimax ℓ-group rules for $\ell = 2, 3$, and m, and without allowing empty groups.

3 Results

Due to space restrictions we present only the results of our work. For the axiomatic study, we first adapt the existing definitions for some properties to handle the more general input type of ℓ-ballots. Then we can show that the minisum ℓ-group rules satisfy nearly all properties at hand, whereas the minimax ℓ-group rules violate some of them. An overview of our results is given in Table 1.

Properties	ℓ-group rules	
	minisum	minimax
Non-imposition, Homogeneity	✓	✓
Consistency	✓	×
Independence of clones	✓	×
Committee monotonicity	✓	×
(Candidate) monotonicity	✓	✓
Positive responsiveness	✓	×
Pareto criterion	✓	✓
(Committee) Condorcet consistency	×	×
Solid coalitions, Consensus committee	×	×
Unanimity	strong	strong

Table 1: Properties for minisum and minimax ℓ-group rules

Next, we study the complexity of computing a winning committee for minisum and minimax ℓ-group rules. For the minisum rule the problem can be solved in polynomial time, as it can be shown that the candidates c with the lowest score $\sum_{v \in V} v(c)$ form a winning committee.

For the study of minimax rules we need the following auxiliary decision problem.

MINIMAX ℓ-SCORE	
Given:	A committee election $E = (C, V, k)$, and a nonnegative integer d.
Question:	Is there a committee $W \in F_k(C)$ such that $\max_{v \in V} \delta_\ell(v, W) \leq d$?

LeGrand et al. [7] show that a problem corresponding to our MINIMAX 2-SCORE is NP-hard, a result that can be generalized to every greater value of ℓ. On these grounds we resort to the study of parameterized complexity. Thus, our goal is to formulate an efficient algorithm when certain parameters of the problem are small, i. e., can be treated as a constant.[3] For approval voting Misra et al. [9] show that the problem is W[2]-hard, when parameterized by the size of the committee. This hardness result also applies to MINIMAX ℓ-SCORE. Hence, an attempt to tune an algorithm with respect to the size of the committee is most likely going to result in failure.

As a positive result we give an algorithm that efficiently solves the MINIMAX ℓ-SCORE problem when the parameter d is treated as a constant, which proofs the following theorem.

Theorem 1 *There is an algorithm solving* MINIMAX ℓ-SCORE *whose running time is in* $O\left((mn + m \log m)\left(\frac{\sqrt{33}}{2}d\right)^d\right)$. *In particular,* MINIMAX ℓ-SCORE *is fixed-parameter tractable when parameterized by* d.

[3] For formal definitions and background regarding parameterized complexity we refer to the book of Downey and Fellows [5].

4 Conclusion

We have introduced different ways of expressing the voters' preferences in committee elections, namely ℓ-ballots, an intermediate between approval votes and linear orders. In addition to axiomatic properties, we have studied the computational complexity of winner determination. While in the minisum case computing a winning committee under ℓ-group rules can be done efficiently, MINIMAX ℓ-SCORE is NP-hard. However, there exists a fixed-parameter tractable algorithm that determines a winning committee.

Note that the input type of ℓ-ballots is only one form of a more general vote. In our setting the differences in scores between two groups are always equivalent and there may be situations where for example the first two groups are of greater importance than the other ones. So as a very general framework one could consider that each voter reports two dissatisfaction values (a, b) to each candidate, one for the case that the candidate is in the committee, the other one for the case where the candidate is not in the committee. We call the resulting voting rules minisum/minimax (a, b)-rules. Obviously our ℓ-group rules are obtained as a special case of such (a, b)-rules, when we restrict the input to $a + b = l - 1$ for each voter. More interestingly, we can show that under some mild restrictions the results obtained in this paper even hold for the very general class of (a, b)-rules.

As a task for future work we propose to identify other interesting special cases of (a, b)-rules and provide a characterization for them. Furthermore, we want to consider different rules for these types of input and identify which of the properties from Table 1 are satisfied, and especially find rules that fulfill Condorcet consistency and committee Condorcet consistency. Closely related to the setting of minisum and minimax elections are the systems of proportional representation, which themselves are related to the interesting concept of justified representation [1]. Thus, a task for future research is to redefine and study these concepts for more general types of votes.

REFERENCES

[1] H. Aziz, M. Brill, V. Conitzer, E. Elkind, R. Freeman, and T. Walsh. Justified representation in approval-based committee voting. In *Proc. AAAI'15*, 2015.

[2] D. Baumeister and S. Dennisen. Voter Dissatisfaction in Committee Elections. In *Proc. AAMAS'15*. IFAAMAS, 2015. Extended abstract.

[3] D. Baumeister, S. Dennisen, and L. Rey. Winner determination and manipulation in minisum and minimax committee elections. In *Proc. ADT'15*, pages 469–485. Springer-Verlag *LNCS #9346*, 2015.

[4] S. Brams, D. Kilgour, and R. Sanver. A minimax procedure for electing committees. *Public Choice*, 132:401–420, 2007.

[5] R. Downey and M. Fellows. *Fundamentals of Parameterized Complexity*. Springer-Verlag, 2013.

[6] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of multiwinner voting rules. In *Proc. AAMAS'14*, pages 53–60. IFAAMAS, 2014.

[7] R. LeGrand. Analysis of the Minimax Procedure. Technical Report WUCSE-2004-67, Department of Computer Science and Engineering, Washington University, St. Louis, Missouri, 2004.

[8] H. Liu and J. Guo. Parameterized Complexity of Winner Determination in Minimax Committee Elections. In *Proc. AAMAS'16*. IFAAMAS, 2016.

[9] N. Misra, A. Nabeel, and H. Singh. On the Parameterized Complexity of Minimax Approval Voting. In *Proc. AAMAS'15*, pages 97–105. IFAAMAS, 2015.

[10] S. Obraztsova, E. Elkind, M. Polukarov, and Z. Rabinovich. Doodle poll games. In *Proc. of the 1st IJCAI-Workshop on Algorithmic Game Theory*, 2015.

[11] W. Zwicker. Introduction to the Theory of Voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 2. Cambridge University Press, 2016.