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

doi:10.3233/978-1-61499-672-9-1658

Space Debris Removal: A Game Theoretic Analysis

Richard Klima[1] and **Daan Bloembergen**[1] and **Rahul Savani**[1] and **Karl Tuyls**[1] and **Daniel Hennes**[2] and **Dario Izzo**[3]

Abstract. We analyse active space debris removal efforts from a strategic, game-theoretic perspective. An active debris removal mission is a costly endeavour that has a positive effect (or risk reduction) for all satellites in the same orbital band. This leads to a dilemma: each actor (space agency, private stakeholder, etc.) has an incentive to delay its actions and wait for others to respond. The risk of the latter action is that, if everyone waits the joint outcome will be catastrophic leading to what in game theory is referred to as the 'tragedy of the commons'. We introduce and thoroughly analyse this dilemma using simulation and empirical game theory in a two player setting.

1 INTRODUCTION AND RELATED WORK

Since the late 1950s a number of public and private actors have launched a multitude of objects into Earth orbits with low or no incentive to remove them after their life span. As a consequence, there are now many inactive objects orbiting Earth, which pose a considerable risk to active spacecraft. By far, the highest spatial density of such objects is in the Low Earth Orbit (LEO) environment, defined as the region of space around Earth within an altitude of 160 km to 2,000 km. According to most simulations and forecast, the density of objects in LEO is destined to increase due to the rate of new launches, on-orbit explosions, and object collisions being higher than the capability of the LEO environment to clean itself using the natural orbital decay mechanism. The objective of this paper is to model this effect and understand its consequences. We thus introduce a non-cooperative game between self-interested agents in which the agents are the owners of space assets. Using a high-fidelity simulator we estimate payoffs to the agents for different combinations of actions taken, and analyse the resulting game in terms of best-response dynamics and (Nash) equilibria. Contrary to the urgency of the space debris dilemma there has not been much attention to this problem in scientific circles. To the best of our knowledge we are the first to consider this dilemma in the context of multi-agent strategic decision making using empirical game theoretic techniques.

Our study can be placed in the context of two different areas of related work. Firstly, from a simulation modelling perspective various attempts have been made to accurately predict the evolution of space debris and the resulting risk of collisions for active spacecraft. One of the earliest analyses of the projected evolution of space debris was done by Donald J. Kessler in 1978 [4]. This study led to the definition of the "Kessler Syndrome", a scenario where the density of objects in LEO becomes high enough to cause a cascade of collisions, each producing new debris and eventually saturating the environment, rendering future space missions virtually impossible. As a result, *active debris removal* (ADR) methods, in which spacecraft are deployed to capture and de-orbit larger pieces of debris and out-of-service satellites, are now considered by many as a necessary step to ensure sustainability of LEO [5]. Secondly, from a game theoretic perspective, researchers have utilised similar methods to study related problems of environmental pollution, and the shared exploitation of scarce resources. For example, carbon dioxide abatement modelled as a differential game [7].

We base our study on Liou and Johnson's single-agent approach [5] but, in contrast, consider a multi-agent scenario in which different space actors independently choose their removal strategy. In our model we implement individualised object removal criteria based on the potential risk to important assets of each of the actors. Our analysis is based on methods from empirical game theory [8] to convert empirical data to strategic (normal-form) games.

2 DEBRIS SIMULATION AND GAME MODEL

Our simulator builds on the Python scientific library PyKep [2], which provides basic tools for astrodynamics research such as satellite orbit propagators. To simulate the future development of space debris in Low Earth Orbit (LEO) we develop several sub-modules, including a collision model and a break-up model. To evaluate the probability of collision between objects we implement the *Cube* approach [6]. We follow NASA's standard breakup model [3] to generate the population of fragments resulting from a collision event. The initial input data to our model comes from the satellite catalogue SATCAT[4] and the TLE (two-line element set) database[5].

We model the space debris removal dilemma as a two-player game, with players being the United States (US) and the European Union (EU). The strategic interaction results from the fact that debris removal by one player may affect the collision risks to others as well. The players' actions are defined by the number of debris objects that will be removed, being either 0, 1, or 2 high risk objects every 2 years. We assume self-interested agents, meaning that each player first removes objects which directly threaten their active satellites, and only then consider objects which present a collision risk in general. The payoffs are based on risk of collision to each player's active satellites, multiplied by the cost of losing an asset C_l, and minus the costs of object removal C_r.

3 SIMULATION RESULTS AND PROJECTIONS

We use our simulator to project the evolution of debris and collision risks with a time horizon of 150 years, i.e. the period 2016-2165, while repeating the launch history of 2006-2015 with a 0.5%

[1]University of Liverpool, UK, email: {richard.klima, d.bloembergen, rahul.savani, k.tuyls}@liverpool.ac.uk [2]DFKI, Bremen, Germany, email: daniel.hennes@dfki.de [3]European Space Agency, Noordwijk, NL, email: dario.izzo@esa.int

[4] https://celestrak.com [5] https://www.space-track.org/

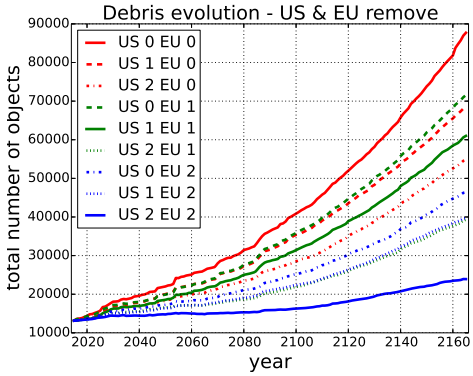


Figure 1. Debris evolution in LEO for next 150 years

yearly increase. For each combination of actions we average over 160 Monte-Carlo runs to account for randomness in the collision and break-up modules. Figure 1 shows the evolution of objects in LEO for different numbers of objects removed by the US and the EU. We observe an exponential growth trend without mitigation, in line with previous findings [5]. One can clearly see that removing high risk objects leads to reduced growth in the total number of objects in LEO. The cumulative risks to both players resulting from the debris evolution in Figure 1 are given in the following table:

	EU 2	*EU 1*	*EU 0*
US 2	0.03413, *0.03733*	0.05247, *0.07108*	0.07704, *0.27474*
US 1	0.06073, *0.06352*	0.09499, *0.10405*	0.10885, *0.31401*
US 0	0.25022, *0.07368*	0.28848, *0.12447*	0.34261, *0.36385*

4 GAME THEORETIC ANALYSIS

We derive the payoff matrix for the two players (US and EU) from the risks given above for varying levels of cost of removal C_r (assuming w.l.g. $C_l = 1$), and find the Nash equilibria. We identify two interesting regions in the range of costs C_r. For very low costs, removing 0 will never be a best response for either player. Similarly, for high costs, removing 2 will never be a best response. Therefore we can focus on two sub-games defined by the action-pairs $\{0, 1\}$ and $\{1, 2\}$. We compute Nash equilibria for a range of C_r, and visualise the results in Figure 2 for the sub-game $\{0, 1\}$ (we observe similar results for the sub-game $\{1, 2\}$). On the y-axis we have the probability of playing the first action in each sub-game (which equals 1 minus the probability of the second action) for US (top) and EU (bottom). The colours/line styles indicate the action pairs that make up the equilibria, e.g. the solid lines in Figure 2 correspond to the pure Nash equilibria $(0, 0)$ (black) and $(1, 1)$ (red).

In the figure we see transitions from the single Nash equilibrium at $(0, 0)$, to a situation where three equilibria exist (at $(0, 1)$, $(1, 0)$, and one mixed), and finally back to a single pure equilibrium at $(1, 1)$. These transition phases also include a stage in which only one of the asymmetric pure equilibria at $(1, 0)$ or $(0, 1)$ exists. These result from the asymmetry that is inherent in the risk matrix due to players having different numbers of assets in different orbits.

In general, ADR has a positive effect not only for the instigator of the removal but also for other players, and this is the cause of the dilemma that we are studying. In game-theoretic terminology, this suggests that we have games with a *weak strategic substitutes property*. Any two-player game that has this property admits a pure equilibrium, which is important for many practical purposes [1].

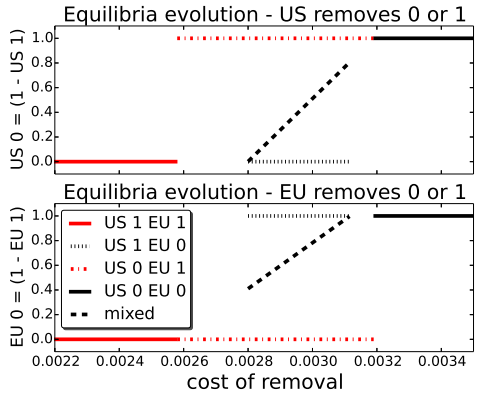


Figure 2. Equilibrium strategies for the sub-game $\{0, 1\}$ for a range of C_r.

5 CONCLUSIONS AND FUTURE WORK

We have introduced a multi-player non-cooperative game named the *Space Debris Removal Dilemma* based on prediction data from our space debris and satellite simulator. The game highlights how the rational behaviour of players varies depending on the cost of active debris removal versus the value of active satellites. In our game-theoretic analysis we identified which removal strategies for the different actors are in equilibrium with each other, i.e. which strategies purely rational actors are expected to decide on. We demonstrated the sensitivity of these equilibrium strategies to the ratio between cost of debris removal and the value of the active satellites. Although the costs of active debris removal are still prohibitively high at the moment they are expected to decrease with future technological developments while the value of orbiting assets may increase. The results of this work help to better understand the debris removal problem and its short and long term consequences.

In future work we aim to move from a one-shot normal-form game to a stochastic or extensive-form game, where the agents can decide on their strategy based on the history of past play. In addition, we will consider a larger set of players, representing e.g. the main space agencies and commercial stakeholders that are currently active.

REFERENCES

[1] Pradeep Dubey, Ori Haimanko, and Andriy Zapechelnyuk, 'Strategic complements and substitutes, and potential games', *Games and Economic Behavior*, **54**(1), 77 – 94, (2006).

[2] Dario Izzo, 'PYGMO and PYKEP: Open source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization)', Technical report, Advanced Concept Team - European Space Research and Technology Centre (ESTEC), (2012).

[3] L. Johnson, N., H. Krisko, P., J.-C. Liou, and D. Anz-Meador, P., 'Nasa's new breakup model of EVOLVE 4.0', *Adv. Space Res. Vol. 28, No. 9, pp. 1377-1384, Elsevier Science Ltd*, (2001).

[4] D. J. Kessler and Burton G. Cour-Palais, 'Collision frequency of artificial satellites: The creation of a debris belt', *Journal of Geophysical Research*, (1978).

[5] J.-C. Liou and L. Johnson, N., 'A sensitivity study of the effectiveness of active debris removal in LEO', *Acta Astronautica*, **64**, 236–243, (2009).

[6] J.-C. Liou, D. J. Kessler, M. Matney, and G. Stansbery, 'A new approach to evaluate collision probabilities among asteroids, comets, and kuiper belt objects', in *Lunar and Planetary Science Conference*, volume 34, p. 1828, (2003).

[7] Olli Tahvonen, 'Carbon dioxide abatement as a differential game', *European Journal of Political Economy*, **10**(1), 685 – 705, (1994).

[8] Michael P. Wellman, 'Methods for empirical game-theoretic analysis', in *Proceedings of the National Conference on Artificial Intelligence*, volume 21, pp. 1552–1555, (2006).

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

doi:10.3233/978-1-61499-672-9-1660

Cuilt: A Scalable, Mix-and-Match Framework for Local Iterative Approximate Best-Response Algorithms

Mihaela Verman and **Philip Stutz** and **Robin Hafen** and **Abraham Bernstein** [1]

Abstract. We implement CUILT, a scalable mix-and-match framework for Local Iterative Approximate Best-Response Algorithms for DCOPs, using the graph processing framework SIGNAL/COLLECT, where each agent is modeled as a vertex and communication pathways are represented as edges. Choosing this abstraction allows us to exploit the generic graph-oriented distribution/optimization heuristics and makes our proposed framework scalable, configurable, as well as extensible. We found that this approach allows us to scale to problems more than 3 orders of magnitude larger than results commonly published so far, to easily create hybrid algorithms by mixing components, and to run the algorithms fast, in a parallel fashion.

1 Introduction

A Distributed Constraint Optimization Problem (DCOP) consists of a set of variables, a set of domains for the variables, a set of constraints over subsets of the variables, and utility functions for each of the constraints. Variables are controlled by agents, and the goal is to find the variable assignment that will maximize the global utility function, usually defined as the sum of utilities over all constraints. Chapman *et al.* [3, 4] propose a theoretical unifying framework for the class of Local Iterative Approximate Best-Response Algorithms, where agents can only be aware of their immediate neighbors' states. CUILT[2] is based on this framework. Chapman *et al.* identify three components for the algorithms, making them easily comparable and enabling the creation of hybrid algorithms: (1) **the state evaluation**, which updates an algorithm-specific **target function** to evaluate prospective states; (2) **the decision rule**, which represents how the agent decides which action to take next by using the already computed target function from the state evaluation step; (3) **the adjustment schedule**, which refers to the order in which the agents execute their processes.

There are other implemented frameworks for algorithms for DCOPs [6, 12, 10, 7], but to our knowledge, none is tailored for this class of algorithms or takes advantage of their modularity. CUILT seeks to model exactly that, and we implement it inside a graph processing framework that comes with several other benefits.

SIGNAL/COLLECT [9] is a framework for distributed large-scale graph processing implemented in Scala[3]. The programming model is vertex-centric: vertices communicate with each other via signals that are sent along directed edges (*signal*), and use the received signals to update their state (*collect*). A problem and an algorithm are specified by providing the graph structure, initial states, and the *signal* and *collect* functions. We chose SIGNAL/COLLECT not only because DCOPs can map well to graph abstractions, but also because of the capabilities and unique features of this graph processing framework: parallel and distributed execution, synchronous/asynchronous scheduling, aggregation operations and convergence detection.

2 CUILT

In CUILT, variables are represented as vertices; the neighborhood (two variables sharing at least one constraint) relationship is described by edges. The utility function of the "agent" responsible for a variable is dependent on the state of the adjacent vertices.

The **Algorithm** will be specified by the way in which different components are mixed and matched. To add different components we use Scala traits, which can be easily combined. CUILT defines the base Algorithm trait, with the types it needs and the methods that it requires. Implementations for the specified CUILT modules then get mixed and need to cover all the required methods of the Algorithm. The flexibility of CUILT comes from the fact that the components are loosely coupled and can be easily mixed and matched. The **SignalCollectAlgorithmBridge** provides the implementations for the vertices and edges, which assemble the components provided by the implementations of other modules, and describes, through its *collect* method, the behaviour of an agent at every step. The methods of Algorithm, which are used inside the *collect* step, are each provided by implementations of the several modules that extend Algorithm: the **TargetFunction**, **DecisionRule** and **AdjustmentSchedule**, providing the functionality described by Chapman *et al.*, and the added **StateModule**, **Utility** and **TerminationRule**.

To illustrate the flexibility of the framework, we provide implementations (through the recombination of components) of four algorithms from the category of iterative approximate best-response algorithms for DCOPs: the Distributed Stochastic Algorithm (DSA-$\overline{A}$, DSA-$\overline{B}$)[11, 1], Distributed Simulated Annealing with Termination Detection (DSAN-TD), Joint Strategy Fictitious Play with Inertia (JSFPI) [8], Weighted Regret Matching with Inertia (WRMI) [2]. Besides these components, we also added the ϵ-Greedy Decision Rule [3]. DSAN-TD is a modification of Distributed Simulated Annealing [1]. While DSAN exhibits oscillations in under-constrained problems [5], in DSAN-TD, to enable termination detection, the probability of exploring decreases over time even when the utilities for the candidate and current value are equal.

3 Evaluation

First, we show how hybrids can be exhaustively evaluated and that they can play an important role when it comes to both speed of convergence and solution quality. *Second*, we show that our framework can handle problems with up to 1 million variables on one machine.

We chose the canonycal *vertex coloring problem*, chromatic numbers of 3, 4 and 5, average edge density of 3, constraints with maximum utility 1, and constructed five graphs per size and chromatic

[1] University of Zurich, Switzerland
[2] https://github.com/elaverman/cuilt/
[3] www.signalcollect.com

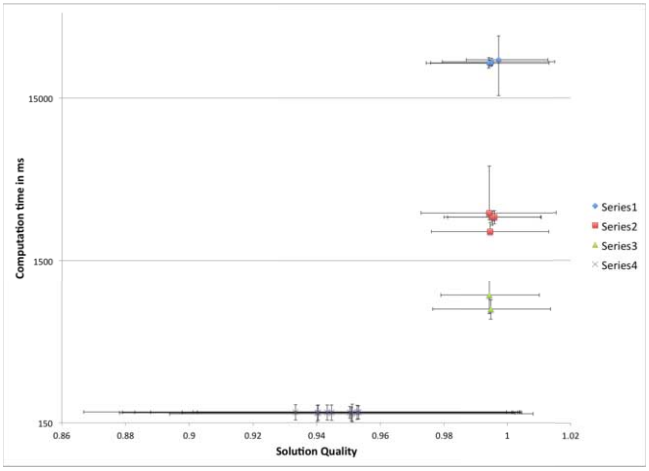


Figure 1. Solution quality vs. computation time in ms (log) for the top 10 average solution quality and top 10 computation time hybrids.

number. In the case of algorithms with the Parallel Random adjustment schedule, for the asynchronous mode we used a degree of parallelism 0.95, as asynchronous mode does not shield from thrashing.

Hybrid algorithms We generated algorithms by taking all possible combinations of the implemented components, with different parameter values. The graphs had 40 vertices and were constructed like in [4]. All the algorithms were run 5 times on these graphs. The machines that we used have 128 GB RAM and two E5-2680 v2 at 2.80GHz processors, with 10 cores per processor.

Figure 1 shows the top 10 combinations in terms of solution quality and convergence speed. We can observe four clusters. The bottom cluster in Series 4 is represented by all the top 10 convergence speed algorithms. They all ran asynchronously, with a Flood adjustment schedule, and with either argmax or ϵ-Greedy adjustment schedules. They were all new hybrids, three being modifications of Fading Memory JSFPI, with a flood schedule, and one being a modification of WRMI with a flood schedule. Most of the top 10 solution quality algorithms (Series 1-3) used the Simulated Annealing decision rule (one of them being DSAN-TD). The algorithms with lowest computation time were asynchronous hybrids with Flood schedule, most of them using the Average Weighted Expected Utility target function from Fading Memory JSFPI and either argmax or ϵ-Greedy decision rules.

Hybrid algorithms can lead to both reduced computation time and improved quality, and running the algorithms asynchronously seems to positively impact speed of convergence.

Data scalability We varied the number of vertices between 10 and 1 million. The graphs were constructed similarly to [4][4]. The timeout was 300 seconds and we executed each configuration 10 times. Convergence was detected only by using the signaling scores of the vertices, which depend on the termination condition. We ran the known algorithms DSA-$\overline{A}$, DSA-$\overline{B}$, and JSFPI on machines with two twelve-core AMD Operon™ 6174 processors and 66 GB RAM[5].

With regard to solution quality, the algorithms behave similarly, dropping from size 10 to 100, and then keeping a relatively stable quality, even when the size of the graph increases. We can see that the asynchronous variants (with probabilities 0.95) usually behave better, but suffer another drop in quality on the largest graphs.

DSA-$\overline{A}$ appears to have the best convergence speed. One possible explanation for the non-linearity of the computation time is reaching the memory limit of the available machines, which makes the computation slower. This could be addressed by using a machine with more memory, or by distributing the framework across multiple machines.

4 Conclusions and Future Work

We introduced CUILT, a software framework based on a mapping of Chapman *et al.*'s theoretical framework to a graph processing model. CUILT is flexible and configurable, allowing to easily create and mix components into hybrid algorithms that can then be exhaustively evaluated to determine a good balance between speed of convergence and solution quality. It has automatic convergence detection, and it enables scaling to problems with 1 million variables and exploiting the advantages of asynchronous execution.

In the future, we intend to implement more components, evaluate the hybrids on different scales and types of problems, including real-world ones, and investigate the behaviour of the algorithms on even larger problems, using the distributed version of SIGNAL/COLLECT.

We believe that our findings show how CUILT enables the systematic exploration of new hybrid algorithms and opens the possibility of applying them to big real-world constraint problems.

ACKNOWLEDGEMENTS

We would like to thank the Hasler Foundation for their generous support, and Dr. J. Enrique Munoz de Cote for his advice.

REFERENCES

[1] M. Arshad and M.C. Silaghi, 'Distributed simulated annealing and comparison to dsa', in *Proceedings of the fourth international workshop on distributed constraint reasoning (DCR-03)*, (2003).

[2] G. Arslan, J.R. Marden, and J.S. Shamma, 'Autonomous vehicle-target assignment: A game-theoretical formulation', *Journal of Dynamic Systems, Measurement, and Control*, **129**, 584, (2007).

[3] A.C. Chapman, A. Rogers, N.R. Jennings, and D.S. Leslie, 'A unifying framework for iterative approximate best-response algorithms for distributed constraint optimization problems', *Knowledge Engineering Review*, **26**(4), 411–444, (2011).

[4] Archie Chapman, Alex Rogers, and Nicholas Jennings, 'Benchmarking hybrid algorithms for distributed constraint optimisation games', *Autonomous Agents and Multi-Agent Systems*, **22**, 385–414, (2011).

[5] A. Flueckiger, M. Verman, and A. Bernstein, 'Improving approximate algorithms for dcops using ranks', in *OptMAS*, (2016).

[6] Thomas Léauté, Brammert Ottens, and Radoslaw Szymanek, 'FRODO 2.0: An open-source framework for distributed constraint optimization', in *Distributed Constraint Reasoning Workshop*, Pasadena, California, USA, (2009). http://frodo2.sourceforge.net.

[7] B. Lutati, I. Gontmakher, M. Lando, A. Netzer, A. Meisels, and A. Grubshtein, 'Agentzero: A framework for simulating and evaluating multi-agent algorithms', in *Agent-Oriented Software Engineering*, pp. 309–327. Springer Berlin Heidelberg, (2014).

[8] J. R. Marden, G. Arslan, and J. S. Shamma, 'Joint strategy fictitious play with inertia for potential games', *IEEE Transactions on Automatic Control*, **54**(2), 208–220, (2009).

[9] P. Stutz, D. Strebel, and A. Bernstein, 'Signal/Collect: Processing Large Graphs in Seconds', *Semantic Web Journal - forthcoming*, (2015).

[10] E. Sultanik, R. Lass, and W. Regli, 'Dcopolis: A framework for simulating and deploying distributed constraint reasoning algorithms', in *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, (2008).

[11] G. Tel, *Introduction to distributed algorithms*, Cambridge Univ Press, 2000.

[12] Mohamed Wahbi, Redouane Ezzahir, Christian Bessiere, and El Houssine Bouyakhf, 'Dischoco 2: A platform for distributed constraint reasoning', in *Distributed Constraint Reasoning Workshop*, (2011).

[4] We picked colors for each vertex, we randomly added edges between vertices with different colors, and then randomly edited edges to connect isolated vertices.

[5] Full results at: `https://docs.google.com/spreadsheets/d/1A0auNnM0MedEgu0SjNk9OjWIn9wdyefMLhmnfcanvwQ/edit#gid=2036767986`

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

doi:10.3233/978-1-61499-672-9-1662

Not Being at Odds with a Class: A New Way of Exploiting Neighbors for Classification

Myriam Bounhas[1] and **Henri Prade**[2] and **Gilles Richard**[3]

Abstract. Classification can be viewed as a matter of associating a new item with the class where it is the least at odds w.r.t. the other elements. A recently proposed oddness index applied to pairs or triples (rather than larger subsets of elements in a class), when summed up over all such subsets, provides an accurate estimate of a global oddness of an item w.r.t. a class. Rather than considering all pairs in a class, one can only deal with pairs containing one of the nearest neighbors of the item in the target class. Taking a step further, we choose the second element in the pair as another nearest neighbor in the class. The oddness w.r.t. a class computed on the basis of pairs made of two nearest neighbors leads to low complexity classifiers, still competitive in terms of accuracy w.r.t. classical approaches.

1 Introduction

Several classification methods rely on the idea that a new item x should be classified in the class w.r.t. which it appears to be the least at odds. Logical proportions [7] are Boolean expressions that link four Boolean variables through equivalences between similarity or dissimilarity indicators pertaining to pairs of these variables. Among logical proportions, the heterogeneous ones [7] provide a natural basis to build a global oddness measure of an item w.r.t. a multiset, by cumulating the oddness index w.r.t. different triples of elements in the multiset for the different features [3]. This has suggested the simple procedure of classifying a new item x into the class $\mathcal{C}$ which minimizes this global oddness measure. Moreover, it has been noticed that the oddness measure of an item w.r.t. a triple can be generalized to a subset of any size, as well as to numerical features [3]. We investigate here the use of oddness measures based on *selected* pairs, since our aim is to show that we can keep on with the same idea of oddness while lightening the computational cost. Doing so, we preserve the accuracy while further reducing the complexity by constraining the set of considered pairs. The idea is to constrain the choice of the *two* elements in the pairs. We study the options of using pairs including one nearest neighbour, and then two nearest neighbors [4].

2 Oddness of an item w.r.t. a multiset of values

In order to build an oddness index for a Boolean x w.r.t. a multiset of Boolean values $\{a_i \mid i \in [1,n]\}$, we look for a formula $F(a_1,\ldots,a_n,x)$ holding iff $a_1 = \ldots = a_n = 0$ and $x = 1$, or if $a_1 = \ldots = a_n = 1$ and $x = 0$. The oddness $odd(\{a_i | i \in [1,n]\},x)$ of a *Boolean* x w.r.t. a multi-set of Boolean values $\{a_i \mid i \in [1,n]\}$ can be summarized as:

$$\neg(\bigvee_{i\in[1,n]} a_i \equiv x) \wedge \neg(\bigwedge_{i\in[1,n]} a_i \equiv x) \quad (1)$$

It is clear that *odd* holds true only when the value of x is seen as being *at odds* among the other values: x is the intruder in the multiset of values. In the case $n = 2$, $odd(\{a_1,a_2\},x)$ is 0 if and only if the value of x is among the majority value in the multiset $\{a_1,a_2,x\}$. When $n = 3$, $odd(\{a_1,a_2,a_3\},x)$ does not hold true only in the situations where there is a majority among values in a_1,a_2,a_3,x and x belongs to this majority (e.g. $odd(\{0,1,0\},0) = 0$), or when there is no majority at all (e.g. $odd(\{0,1,1\},0) = 0$).

Extension to *real values* is quite straightforward. Assuming that numerical features are renormalized between 0 and 1, we use the standard definitions of the logical connectives in the $[0,1]$-valued Łukasiewicz logic [8]. Then, a graded counterpart to formula (1) is:

$$\min(|x - \max\{a_1,\ldots,a_n\}|, |x - \min\{a_1,\ldots,a_n\}|) \quad (2).$$

A natural extension $odd(\{\overrightarrow{a_i} | i \in [1,n]\}, \overrightarrow{x})$ to *vectors* with m features is to consider the sum componentwise of the odd values computed via expression (1) or (2) as:

$$\Sigma_{j=1}^{m} odd(\{a_i^j | i \in [1,n]\}, x^j) \quad (3),$$

where x^j is the j-th component of $\overrightarrow{x}$ and the a_i^j's are the j-th components of the vectors $\overrightarrow{a_i}$ respectively. If $odd(\{\overrightarrow{a_i} | i \in [1,n]\}, \overrightarrow{x}) = 0$, it means that no feature indicates that $\overrightarrow{x}$ behaves as an intruder among the a_i's. On the contrary, high values of $odd(\{\overrightarrow{a_i} | i \in [1,n]\}, \overrightarrow{x})$ means that, on *many* features, $\overrightarrow{x}$ appears as an intruder.

3 Global oddness measure for classification

Given a set $\mathcal{C} = \{\overrightarrow{a_i} | i \in [1,n]\}$ of vectors gathering examples of the same class, one might think of computing $odd(\mathcal{C}, \overrightarrow{x})$ as a way of evaluating how much $\overrightarrow{x}$ is at odds w.r.t. $\mathcal{C}$. An immediate classification algorithm would be to compute $odd(\mathcal{C}, \overrightarrow{x})$ for every class and to allocate to $\overrightarrow{x}$ the class which minimizes this number. Nevertheless, this number is not really meaningful when the size of $\mathcal{C}$ is large. Indeed, we have to be careful because then $\{\overrightarrow{a_i} | i \in [1,n]\}$ is summarized by two vectors made respectively by the minimum and the maximum of the feature values among the examples of $\mathcal{C}$ (due to expressions (2) and (3)). These two vectors have high chance to be *fictitious* in the sense that, usually, they are not elements of $\mathcal{C}$. Approximating our knowledge of the set $\mathcal{C}$ using only the maximal ranges of the feature values over the members of the set seems very crude. An idea is then to consider small subsets S of the class $\mathcal{C}$, then compute $odd(S, \overrightarrow{x})$ and finally add all these atomic oddness indices to get a global measure of oddness of $\overrightarrow{x}$ w.r.t. $\mathcal{C}$. This approach leads to the following initial formula: $\Sigma_{S\subseteq\mathcal{C},|S|=n}\, odd(S, \overrightarrow{x})$. To take into account the relative size of the different classes, it is fair to introduce a normalization factor and our final definition is:

$$Odd_n(\mathcal{C}, \overrightarrow{x}) = \frac{1}{\binom{|\mathcal{C}|}{n}} \Sigma_{S\subseteq\mathcal{C},|S|=n}\, odd(S, \overrightarrow{x})$$

[1] LARODEC Laboratory, ISG de Tunis, Tunisia & Emirates College of Technology, Abu Dhabi, United Arab Emirates, email: myriam_bounhas@yahoo.fr
[2] IRIT, University of Toulouse, France, & QCIS, University of Technology, Sydney, Australia, email: prade@irit.fr
[3] IRIT, University of Toulouse, France, & BITE, London, UK, email: richard@irit.fr

A classification algorithm follows, where we allocate to a new item x the class $\mathcal{C}$ minimizing $Odd_n(\mathcal{C}, \overrightarrow{x})$ for a given n, thus defining a family of classifiers also denoted Odd_n.

Algorithm 1 Oddness-based algorithm

Input: a training set TS of examples $(\overrightarrow{z}, cl(\overrightarrow{z}))$
a new item $\overrightarrow{x}$,
an integer n,
Partition TS into sets $\mathcal{C}$ of examples with the same label c.
for each $\mathcal{C}$ **do**
Compute $Odd_n(\mathcal{C}, \overrightarrow{x})$ for subsets of size n
end for
$cl(\overrightarrow{x}) = argmin_c(Odd_n(\mathcal{C}, \overrightarrow{x}))$
return $cl(\overrightarrow{x})$

4 Experimentations

The experimental study is based on 15 datasets taken from the U.C.I. machine learning repository [6], applying standard 10 folds cross-validation technique. On these datasets, we compare classifiers Odd_1, Odd_2, Odd_3 accuracy to the one of state-of-the-art classifiers C4.5, SVMs, JRip and IBk (shown in Table 1). Our first experiments

Table 1. Classification accuracy of state-of-the-art classifiers

Datasets	C4.5	SVM	SVM	JRip	IBk
		Poly-Kernel	PUK-Kernel		(k=5,k=15)
Balance	78	90	89	76	84,88
Car	95	91	87	91	92,76
Monk1	99	75	100	98	100,100 -
Monk2	95	67	67	73	64,67
Monk3	100	100	100	100	99,98
Spect	81	81	83	81	80,79
Voting	96	96	96	95	93,91
Diabetes	74	77	77	76	73,74
Cancer	96	97	96	96	97,94
Heart	77	84	81	81	78,81
Magic	76	77	81	77	78 ,77
Ionosphere	91	88	94	88	85,84
Iris	96	96	96	95	95,96
Wine	94	98	99	93	95,96
Sat. Image	94	94	95	93	95,94

lead to rather poor performances due to the huge number of subsets considered in each class, giving them equal importance, while a lot of them blur the accuracy of oddness measure through the summation.

In our first experiments, considering the average accuracy on all 15 datasets, it appears that Odd_2 is a better performer than Odd_1 and Odd_3. We then focus on Odd_2, trying to privilege subsets including elements of particular interest such as nearest neighbors in the classes. Since selecting one element of a pair as a nearest neighbour of the item x in the target class leads to good accuracy rates, we also consider the option of taking the second element as another nearest neighbour in the class, with the normalization factor chosen accordingly as $\frac{1}{\binom{k}{2}}$, leading to $Odd_2(NN, NN)$. This is also beneficial from a complexity viewpoint.

In Tables 2 and 3, we provide classification results respectively for $Odd_2(NN, Std)$ (using only one nearest neigbour), and $Odd_2(NN, NN)$ (using two nearest neigbours) for different values of k (k being the number of nearest neigbours used).

In the last column of Table 3, we assign respectively a positive $'+'$, negative $'-'$, or neutral $'.'$ mark if the $Odd_2(NN, NN)$ is respectively better, worse or similar to $Odd_2(NN, Std)$ for $k = 15$. This comparison shows that the two classifiers have similar efficiency for most datasets, except for Monk2 where $Odd_2(NN, NN)$ outperforms not only $Odd_2(NN, Std)$, but also SVM and IBk.

We can also note that $Odd_2(NN, NN)$ performs more or less in the same way as the best known algorithms. Especially we compared this classifier to IBk for k=15 using the Wilcoxon Matched-Pairs Signed-Ranks [5]. We get a p-value = 0.024 which shows that $Odd_2(NN, NN)$ is significanly better than IBk.

Table 2. Classification accuracies given as mean and standard deviation with $Odd_2(NN, Std)$ classifier for $k = 3, 7, 13, 15$

Datasets	$Odd_2(NN, Std)$			
Value of k	3	7	13	15
Balance	75.61±4.99	85.65±4.11	86.23±3.59	86.33±3.57
Car	87.73±3.99	92.68±3.17	90.85±2.99	90.1±2.86
Monk1	99.59±2.95	99.76±2.19	99.38±2.87	99.33±2.87
Monk2	36.39±8.57	53.03±5.85	61.34±5.34	64.28±4.4
Monk3	99.82±0.72	99.87±0.66	99.6±1.31	99.14±1.82
Spect	82.98±5.17	84.08±4.7	83.82±5.76	83.38±5.76
Voting	92.96±4.29	94.33±3.38	94.98±2.63	94.39±3.82
Diabetes	73.61±3.99	75.1±3.49	75.5±3.03	75.93±3.98
W. B. Cancer	95.79±2.32	96.31±1.82	96.37±1.91	96.49±2
Heart	81.55±3.94	82.52±5.39	82.1±5.34	82.34±5.55
Magic	78.39±3.27	78.32±3.03	78.83±3.18	78.94±3.09
Ionosphere	91.22±4.11	90.56±4.92	91.58±5.4	91.69±5.6
Iris	94.53±5.47	94.53±5.47	95.21±4.68	95.21±4.68
Wine	97.58±2.71	98.24±2.44	97.67±2.87	97.58±2.87
Sat. Image	95.1±1.48	94.59±1.36	93.93±1.77	93.84±1.67

Table 3. Classification accuracies given as mean and standard deviation with $Odd_2(NN, NN)$ classifier for $k = 3, 7, 13, 15$

Datasets	$Odd_2(NN, NN)$				
Value of k	3	7	13	15	
Balance	56.37±6.56	83.63±3.89	83.72±4.12	84.47±3.54	(-)
Car	75.38±4.53	92.39±3.89	91.26±2.89	90.36±3.83	(.)
Monk1	85.95±4.66	99.76±0.24	99.56±0.74	99.57±0.94	(.)
Monk2	53.72±8.1	59.7±4.6	67.93±4.23	67.73±3.13	(+)
Monk3	94.49±4.97	99.95±0.05	99.91±0.71	99.59±1.25	(.)
Spect	76.52±6.02	82.77±5.09	83.82±3.68	84.51±4.80	(+)
Voting	90.69±2.92	93.36±2.85	94.3±3.57	94.86±3.03	(.)
Diabetes	68.48±5.2	73.06±3.22	74.55±3.48	74.84±3.03	(-)
W. B. Cancer	94.05±2.48	95.86± 2.88	96.05±2.38	96.02±2.32	(.)
Heart	74.29±4.72	79.46±8.26	81.44±6.07	82.58±7.60	(.)
Magic	75.25±3.41	78.68±3.73	79.24±2.53	79.52±3.15	(+)
Ionosphere	91.17±4.2	91.64±2.88	91.81±2.0	91.85±2.12	(.)
Iris	94.59±5.62	94.59±5.62	94.7±5.63	94.98±5.24	(.)
Wine	97.64±4.81	97.81±2.65	97.77±2.78	97.74±2.78	(.)
Sat. Image	95.51±1.75	95.29±1.1	94.79±1.69	94.77±1.81	(+)

5 Conclusion

In this paper, we suggest a new way to evaluate the oddness of an item w.r.t. a class. Since using subsets of pairs (Odd_2) provides better accuracy results than singletons (Odd_1) or triples (Odd_3), we further investigate this option by filtering candidate pairs. We first choose one item in a pair as a nearest neighbour, then two elements in a pair as nearest neighbours in the class. Experiments show that we are still competitive with state of the art classifiers (k-NN, SVM) while having drastically decreased complexity. These results suggest that it may be beneficial to consider *pairs of nearest neighbours* and then to minimize an oddness measure, which departs from the k-NN view dealing with nearest neighbours in isolation.

REFERENCES

[1] M. Bounhas, H. Prade, and G. Richard, 'Analogical classification: A new way to deal with examples', in *Proc. 21st Europ. Conf. on Artificial Intelligence, Praha, ECAI'14*, volume 263, pp. 135–140. IOS Press, (2014).

[2] M. Bounhas, H. Prade, and G. Richard, 'Analogical classification: Handling numerical data', in *Proc. of Scalable Uncertainty Management - 8th Int. Conf, SUM*, eds., U. Straccia and A. Calì, volume 8720 of *LNCS*, pp. 66–79. Springer, (2014).

[3] M. Bounhas, H. Prade, and G. Richard, 'Oddness-based classifiers for boolean or numerical data', in *KI 2015: Advances in Artificial Intelligence - 38th Annual German Conference on AI, Dresden, Germany, September 21-25, 2015, Proceedings*, pp. 32–44, (2015).

[4] *Nearest Neighbor: Pattern Classification Techniques (Nn Norms : Nn Pattern Classification Techniques)*, ed., B. V. Dasarathy, IEEE Computer Society Press, 1990.

[5] J. Demsar, 'Statistical comparisons of classifiers over multiple data sets', *Journal of Machine Learning Research*, **7**, 1–30, (2006).

[6] J. Mertz and P.M. Murphy, 'Uci repository of machine learning databases', *Available at: ftp://ftp.ics.uci.edu/pub/machine-learning-databases*, (2000).

[7] H. Prade and G. Richard, 'Homogenous and heterogeneous logical proportions', *IfCoLog J. of Logics and their App.*, **1**(1), 1–51, (2014).

[8] N. Rescher, *Many-valued Logic*, McGraw-Hill Inc., New York, 1969.

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

doi:10.3233/978-1-61499-672-9-1664

Value-Based Reasoning and Norms[1]

Trevor Bench-Capon [2]

Abstract. Norms are designed to guide choice of actions. Value-based practical reasoning is an approach to explaining and justifying choice of actions in terms of value preferences.. Here we explore how value-based practical reasoning can be related to norms and their evolution. Starting from a basic model of a society and the norms that can arise from it, we consider how additional values, and a more sophisticated model, with more detailed states and a history, and a finer grained description of actions, can accommodate more complex norms, and a correspondingly more complex social order.

1 Norms and Values

Norms are a topic of considerable interest in agents systems, since they are seen as a way of regulating open agent systems. Simple two player games, such as the prisoner's dilemma (PD), can be used to explore norms [3], [8]. Empirical studies suggest, however, that public goods games do not provide a very realistic model of actual human behaviour. Studies such as [7] demonstrate that the canonical model is rarely followed in practice. An alternative approach is provided by Value-Based Reasoning, in which agents are associated with a set of social *values*, the aspirations and purposes an agent might pursue, and their choice of actions explained by these values [1]. Norms can in turn be explained in terms of these choices. An example illustrating this approach using the fable of *The Ant and the Grasshopper* and the parable of *The Prodigal Son* can be found in [4].

In both the fable and parable there are two agents (ant and grasshopper and father and son, respectively). In both there is a choice of working through the summer to build a food surplus, or playing for one's own pleasure. In terms of values, the grasshopper and the son choose pleasure over work. When winter comes and they have no food they ask to be fed. The ant refuses, saying that it is the grasshopper's own fault, but the father does give the son food. Thus, in terms of values, the ant prefers its own pleasure (feasting on its store) to the grasshopper's life, whereas the father makes the opposite choice. The norms underlying the fable seems to be a kind of work ethic (one should choose work, reinforced by a norm saying that those who do not work should starve). The parable shares the first norm (the son is *prodigal*), but regards the life of the son as more important than the father's feasting. Since the preference of the ant seems rather selfish, perhaps even immoral [2], it is important that the norm obliging work is recognised, so that the refusal can be seen as punishment of violation of the norm, promoting the value Justice rather than a selfish preference. Given the norm, the father's action can be seen as promoting a value such as Mercy or Forgiveness, by withholding a deserved punishment. Thus we see how the existence of norms allows the introduction of additional values. Punishment of violations is necessary to avoid normative collapse [6], so the repentance of the son (an additional action which will extend the basic model) is an essential part of the parable: it is expected that he will work in future summers.

Although norms to work and to punish violations (in the fable, repeated violations in the parable) will give rise to an equitable and sustainable society, the society could be critiqued: there is no net pleasure, no choice, no diversity, and the pleasure that does exist (feasting in winter) is rather basic, whereas the pleasure denied (singing) can be seen as a 'higher" pleasure, which utilitarians such as Mill have valued more highly than basic pleasures. It can be seen as a mark of a civilised society that it uses surplus food to enable the development of arts and sciences. This can be reflected by distinguishing three types of pleasure: bodily pleasures, higher pleasures and mere frivolity. Now our norms can make use of these distinctions, by punishing frivolity, but encouraging higher pleasures by giving food to those who are in need because of their pursuit of these pleasures. The problem is ensure that there will be enough surplus food to feed those who do choose higher pleasures over work: this requires at least half the population to work. But why should they choose work? There are a number of ways in which we can accommodate agents choosing to play. Some require disparity between agents, while others require a redescription of the world: additional values and refined descriptions of actions and states.

Power. We first consider a disparity of power. In this situation some agents are sufficiently more powerful than the others as to be able to compel them to surrender their food. The powerful can choose pleasure without fear of starving. But a norm is required to prevent them demanding non-surplus food, representing a preference for Life over (any form of) Pleasure. Also we will wish to discourage frivolity, so there will be a norm (addressed only to the powerful, the rest will still be obliged to work) forbidding frivolity (allowing work as a choice) or obligating pursuit of higher pleasures.

This means that there is one norm for the powerful and one norm for the powerless, which requires some kind of social order, recognised by all. One example of such a society is Feudalism. If there are relatively few powerful agents, they can demand low rents and so leave some surplus to the tenants If such a society is to be sustainable, the powerless need to respect the social order so that they do not rise up and overthrow the elite. Revolutions must be avoided. The social order can be supported by additional values, such as *Deference*, respected by the powerless and a kind of *Nobelsse Oblige* respected by the powerful. Acceptance can be reinforced in several ways including patriotism, in which the powerless are encouraged to take pride in the cultural achievements of their masters, or religion. As a further reinforcement, prudence suggests that the rents should not be too high. The proportion to take resembles the Ultimatum Game [7]. A further possibility is that some workers may be taken out of food

[1] This is a short version of a paper presented at the ECAI 2016 Workshop *AI for Justice*. That paper has a fuller discussion and set of references.

[2] Department of Computer Science, University of Liverpool, UK. email: tbc@csc.liv.ac.uk

production and used for other purposes of benefit to all, which might be additional cultural activities (e.g. minstrels), building works (e.g. the pyramids), or whatever, and then fed from the tribute ("bread and circuses"). In addition to Feudalism, there are other models: slavery is one, and the kind of brigandry depicted in the film *The Magnificent Seven* is another, but these afford far less opportunity for keeping the powerless content.

Money. In post-feudal societies we find that class and disparity remain, but that this disparity is manifested as wealth rather than physical coercion. When wealth is the source of power, the forcibly coercive demands of the powerful are replaced by the ability to buy the surplus. Selling is not compulsory, but avoids the wealthy starving and allows the possibility of acquiring money to allow for future pleasures, or deferring work through debt. This is the underlying idea of holidays, pensions, and more recently of "gap years". How the surplus is distributed can be left to the individuals and so made to depend on the preferences of individuals, or there may be limits imposed by a state. This could lead to a fair degree of equality, since eventually the initially wealthy will have spent all their money, and so be forced to work. There are, however, mechanisms which tend to allow the wealthy to maintain their position: including land ownership, access to external wealth (e.g. colonies) and usury. The notion of money which allows consumption to be deferred or anticipated requires a norm obligating the repayment of debt, or more generally to honour agreements, and will be accompanied by new values such as *trustworthiness* and *honesty*. In some cases deference or genorosity may mean that some people (e.g. monarchs, priests or those who cannot work) are supported without payment. This may lead to support a norm obligating giving alms to those who cannot support themselves (to the poor, or to those performing a worthwhile service, or both). Once the need to honour agreements has been recognised the possibility of turn-taking arrangements arises. Such arrangements are common amongst children and households sharing chores, and as been been shown to emerge in certain kinds of agent simulations [5].

The "play" may also be of value to the working agents. Here some agents may be prepared to part with a (typically) small part of their surplus. Since the singing of a single grasshopper may entertain a whole colony of ants, it is even more attractive if the cost can be shared across a large number of individuals. Where this is so, a variety of entertainers can be supported, and other services performed. Money greatly assists this arrangement, and places it on a contractual footing. As such we might expect the emergence of a service and entertainments sector, where some agents are able to adopt the role of providers of cultural activities willingly supported by groups of other agents. This is likely to be increasingly the case when productivity rises, so that workers generate larger surpluses. Now both food production and providing services for which others will pay can be seen as "work". Only the best will be paid to entertain, allowing for such play to be distributed on merit (rather than on power or wealth).

Government. As well as choosing to spend their surplus on providing themselves with culture, through paying others to play in particular ways, agents may choose to pay others to do their duties. In [6] it was shown empirically that to avoid norms collapsing it is necessary that they not only be backed by the punishment of violators, but that those who fail to punish must themselves be punished. Since punishment has a cost, however, there are reasons not to punish, and in societies where violations are comparatively rare, the cost of punishment falls unevenly and unpredictably. Recognising the need to punish is an important aspect of social cohesion. Once this is seen as a social duty it is a small step to organise and pay for a third party to punish violators. From this it is a small step to taxation, and the provision of services such as law enforcement by the State. And if law enforcement, why not other duties? In this way Governments may emerge, first as a Hobbesian *Leviathan* for mutual protection but, once established, available to take on the performance of other duties, such as supporting those incapable of work, or even centrally providing entertainers.. In addition to adding an additional actor to the model, an emergent state will lead to further new values such as *self-reliance*, *freedom*, *community*, and their relative preferences may provide insight into the form in which the Government emerges.

Discussion. Value-Based practical reasoning [1] was developed to explain and justify choices of actions in terms of the subjective aspirations of agents, represented as an ordering on values. Norms are designed to guide such choices and can be seen as corresponding to, and hence encouraging, particular value orderings. As such norms and value-based reasoning should relate to one another. In our very simple initial example, based on the fable of *The Ant and the Grasshopper*, for the society depicted to be sustainable a norm enjoining work over play was required. But since normative collapse requires the punishments of violations [6], it was important that the ant did not simply meet the grasshopper's needs. Because this appeared to be endorsed by the ant selfishly choosing pleasure over the grasshopper's life, the new value of justice was introduced to justify such punishment. Corresponding values such as forgiveness were then required to justify the action of the father in the parable of *The Prodigal Son*. But because violations eventually need to be punished to avoid normative collapse, we need to record the history of the system to recognise repeated violations. Thus we are elaborating the states, and introducing further values, some of them meta-values: values promoted and demoted by value orders rather than actions. *Greed* would be another meta-value, representing the undue preference for an agent's own wealth. Norms encourage particular orderings.

We have also shown how finer grained descriptions of actions (e.g. discriminating various kinds of pleasurable activity), and additional state information (recording disparities of power and wealth) give rise to, or are required by, more sophisticated norms. The introduction of money, for example, requires a set of accompanying norms to regulate its use. These norms in turn support a variety of social orders, and more specialised and diverse societies. Different normative systems can in this way be explained in terms an interplay of values, states, norms and social aspirations, and approved preference orderings, supported by meta-values.

REFERENCES

[1] K. Atkinson and T. Bench-Capon, 'Practical reasoning as presumptive argumentation using action based alternating transition systems', *Artificial Intelligence*, **171**(10-15), 855–874, (2007).

[2] K. Atkinson and T. Bench-Capon, 'Addressing moral problems through practical reasoning', *Journal of Applied Logic*, **6**(2), 135–151, (2008).

[3] R. Axelrod, *The evolution of cooperation*, Basic Books, 1987.

[4] F. Bex, K. Atkinson, and T. Bench-Capon, 'Arguments as a new perspective on character motive in stories', *Literary and Linguistic Computing*, **29**(4), 467–487, (2014).

[5] M. Lloyd-Kelly, K. Atkinson, and T. Bench-Capon, 'Fostering cooperative behaviour through social intervention', in *Proceedings of (SIMULTECH), 2014*, pp. 578–585. IEEE, (2014).

[6] S. Mahmoud, N. Griffiths, J. Keppens, A. Taweel, T. Bench-Capon, and M. Luck, 'Establishing norms with metanorms in distributed computational systems', *AI and Law*, **23**(4), 367–407, (2015).

[7] H. Oosterbeek, R. Sloof, and G. Van De Kuilen, 'Cultural differences in ultimatum game experiments: Evidence from a meta-analysis', *Experimental Economics*, **7**(2), 171–188, (2004).

[8] Y. Shoham and M. Tennenholtz, 'On the emergence of social conventions: modeling, analysis, and simulations', *Artificial Intelligence*, **94**(1), 139–166, (1997).

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

doi:10.3233/978-1-61499-672-9-1666

Using Recursive Neural Networks to Detect and Classify Drug-Drug Interactions from Biomedical Texts

Víctor Suárez-Paniagua and **Isabel Segura-Bedmar** [1]

Abstract. The purpose of this paper is to explore in detail how a Recursive Neural Network can be applied to classify drug-drug interactions from biomedical texts. The system is based on MV-RNN, a Matrix-Vector Recursive Neural Network, built from the Stanford constituency trees of sentences. Drug-drug interactions are usually described by long sentences with complex structures (such as subordinate clauses, oppositions, and coordinate structures, among others). Our experiments show a low performance that may be probably due to the parser not being able to capture the structural complexity of sentences in the biomedical domain.

1 INTRODUCTION

Nowadays there is a growing concern about drug adverse events (ADEs) since they are a serious risk for patient safety as well as a cause of rising health care costs. Drug–drug interactions (DDIs), a subset of ADE, are harms caused by the alteration of the effects of a drug due to recent or simultaneous use of one or more other drugs. Unfortunately, most DDIs are not detected during clinical trials, mainly because these trials are designed to assess the effectiveness of drugs rather than their safety. Physicians can consult several databases with information about DDIs before writing their prescriptions for medications. However, in general, the information in these databases is incomplete and is not up to date. It should denoted that some DDIs do not have any negative consequences on health patient or can be avoided with proper drug dosages or increasing the interval time between the drugs. Coupled with this, it should be noted that the number of articles published in the biomedical domain is increasing between 10,000 and 20,000 articles per week. Thus, physicians have to spend a long time reviewing DDI databases as well as the pharmacovigilance literature in order to assess the real clinical significance of a particular DDI and to prevent harmful drug interactions. Therefore, the automatic processing of the medical literature is becoming vital for the early detection of DDIs. Natural Language Processing (NLP), that is language technologies, can be a powerful tool to find out new insights regarding DDIs.

In recent years, several NLP challenges have been organized to promote the development of NLP techniques applied to the biomedical domain, in particular, to the pharmacovigilance subject. In particular, the goal of DDIExtraction Task [6] was the detection and classification of DDIs from biomedical texts. Each DDI is classified with one of the following types of DDIs: mechanism (when the DDI is described by their pharmacokinetic mechanism), effect (for DDIs describing an effect or a pharmacodynamic mechanism), advice (when sentence is providing a recommendation or advice about a DDI) and int (the DDI appears in the text without providing any additional information). Most of the participating systems as well as the systems developed later have been based on Support Vector Machines (SVM) and on both linear and non-linear kernels, being the state-of-the-art performance of 77.5% F1 for detection and 67% F1 for classification [4]. All of them are characterized by the use of large and rich sets of linguistic features proposed by text miners and domain experts. Deep learning methods can be an interesting alternative to the classical methods since they are able to learn the best features to represent a problem. To the best of our knowledge, only two works so far have used Deep Learning methods for the classification of DDIs from biomedical texts. The first work exploited a recursive neural network (RNN) [2], achieving 68.64% F1 for DDI classification, however the performance for each DDI type was not studied. The second one was based on the use of a convolutional neural network (CNN) [7] (F1=69.75%; 70.24% for mechanism; 69.33% for effect; 77.75% for advice and 46.38% for int). The goal of our work is to continue the work started by [2], performing a detailed study about if RNNs are able to correctly classify the different DDI types.

2 EXPERIMENTAL SETTINGS

2.1 Dataset

The DDI corpus [3] is a valuable annotated corpus that provides a gold standard data for training and evaluating supervised machine learning algorithms to extract DDIs form texts. The DDI corpus was used for the DDIExtraction 2013 Shared Task. It contains 233 selected abstracts about DDIs from Medline (DDI-MedLine) and other 792 texts from the DrugBank database (DDI-DrugBank). The corpus was manually annotated by two pharmacists with a total of 18,502 pharmacological substances and 5028 DDIs. The DDI corpus includes four different type of DDIs: *mechanism* (1618), *effect* (2029), *advice* (1043) and *int* (284). The next example represents an *advise* instance where the two entities in the relation are marked:
"*Caution should be used when <e1>NSAIDs</e1> are administered concomitantly with <e2>methotrexate</e2>.*"

2.2 MV-RNN

Our approach is based on Recursive Neural Network. In particular, Matrix-Vector spaces (MV-RNN) was the first Deep Learning architecture that obtained improvements in classification of semantic relationships [8]. This model can determine the meaning of each word and the rules used to combine them in long sentences. To this end, the model assigns a vector and a matrix to every word and it learns a compositional function for computing these representations.

[1] Computer Science Department, University Carlos III of Madrid, email: vspaniag,isegura@inf.uc3m.es

Firstly, MV-RNN uses as input a binarized parse tree of phrases and sentences of arbitrary syntactic type and length from the Stanford Parser [5] as the RNN structure. Then, MV-RNN learns, in every node of the tree, a vector that represents the meaning of a constituent (a word or a sentence), and a matrix that captures how this constituent changes the meaning of their neighbours. Initially, we use the pretrained 50-dimensional word vectors from [1] and the matrices as an identity matrix with a small Gaussian noise. Afterwards, the MV-RNN architecture computes the parent vector p and the non-terminal phrase matrix P of each node as a single layer neural network:

$$p = g(W \begin{bmatrix} C_1 c_2 \\ C_2 c_1 \end{bmatrix} + b) \qquad P = W_M \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

where c_1 and c_2 are the word vectors of their children in the binarized tree with dimensionality n, C_1 and $C_2 \in \mathbb{R}^{n \times n}$ are matrices for single words, W and $W_M \in \mathbb{R}^{n \times 2n}$ are the weight matrices, g is a non-linearity function and b is the bias term. Finally, the MV-RNN uses the computed vector of the highest nodes in the path between the pairs of words as features for predicting a DDI type label using a simple *softmax* classifier.

MV-RNN was adapted by Socher et al., [8] for the SemEval-10 task 8, whose goal was the classification of relationships between nominals. Thus, we had to transform the DDI corpus to the format of the SemEval-2010 task 8. Since the implementation of MV-RNN does not deal with discontinuous entities, we removed DDI candidates involving this kind of entities. It should be noted that sentences from the SemEval-2010 task 8 dataset are much simpler than our sentences in the DDI corpus. For this reason, we simplified complex entities (such as chemical compounds) and numerical expressions in order to clarify the tokenization of texts.

3 RESULTS AND DISCUSSION

Table 1 shows the performance of MV-RNN over the DDI corpus test dataset. The model achieves an F1 score of 46% using a syntactic information for building the RNN architecture. In general, precision is greater than recall due to the large number of false negatives in each class caused by the misclassification. The class *advice* achieves the best performance (54% in F-measure) with respect to the other classes because these recommendations follows specific patterns and are easy to learn.

Table 2 shows the results adding external features such as Part-of-Speech tags, the WordNet hypernyms and the name entity tags of the two words to the computed vector of the highest node in the relation for the classification in the *softmax* layer. These three features increase the performance F-measure (50%) and the Recall for all the classes. Although, the features raise the number of instances classified correctly, the False Positives are 38 instances bigger than without using external features. It may be due to an over-fitting in the *softmax* layer because in all the cases the False Negatives decreases whereas that the False Positives increases with respect to the Table 1 causing a trade-off problem.

The main cause of our low performance may be due to the Stanford parser is not able to correctly build the syntactic trees of sentences from the DDI corpus, which usually have complex structures (such as subordinate clauses, oppositions and coordinate structures, complex named entities among others). Wrong syntactic trees involve wrong RNN structures that are not able to capture the compositionality of sentences.

	True Positives	False Positives	False Negatives	Precision	Recall	F-measure
EFFECT	163	206	197	0.44	0.45	0.45
MECHANISM	105	102	194	0.51	0.35	0.42
ADVICE	108	71	113	0.60	0.49	0.54
INT	36	18	60	0.67	0.38	0.48
TOTAL	412	397	564	0.51	0.42	0.46

Table 1. Results on DDI Corpus using MV-RNN without external features.

	True Positives	False Positives	False Negatives	Precision	Recall	F-measure
EFFECT	193	232	167	0.45	0.54	0.49
MECHANISM	121	110	178	0.52	0.40	0.46
ADVICE	119	76	102	0.61	0.54	0.57
INT	37	17	59	0.69	0.39	0.49
TOTAL	470	435	506	0.52	0.48	0.50

Table 2. Results on DDI Corpus using MV-RNN with external features.

4 FUTURE WORK

As future work we propose to replace the pretrained word vectors from the Collobert and Weston model [1] by those from *word2vec*[2]. The model will train on a large collection of biomedical texts to include technical terms and jargon, which are not generally represented by [1]. Moreover, we would like to explore if the position embeddings and negative instance filtering improve our results in MV-RNN.

ACKNOWLEDGEMENTS

This work was supported by eGovernAbility-Access project (TIN2014-52665-C2-2-R).

REFERENCES

[1] Ronan Collobert and Jason Weston, 'A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning', in *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 160–167, New York, NY, USA, (2008). ACM.

[2] Javid Ebrahimi and Dejing Dou, 'Chain Based RNN for Relation Classification', in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1244–1249, Denver, Colorado, (May–June 2015). Association for Computational Linguistics.

[3] María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck, 'The DDIcorpus: An annotated corpus with pharmacological substances and drug–drug interactions', *Journal of Biomedical Informatics*, **46**(5), 914 – 920, (2013).

[4] Sun Kim, Haibin Liu, Lana Yeganova, and W. John Wilbur, 'Extracting drug–drug interactions from literature using a rich feature-based linear kernel approach', *Journal of Biomedical Informatics*, **55**, 23 – 30, (2015).

[5] Dan Klein and Christopher D. Manning, 'Accurate Unlexicalized Parsing', in *In Proceedings of the 41St Annual Meeting of the Association for Computational Linguistics*, pp. 423–430, (2003).

[6] Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo, 'Semeval-2013 Task 9: Extraction of Drug-Drug Interactions from Biomedical Texts', in *In Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval)*, (2013).

[7] Qingcai Chen Shengyu Liu, Buzhou Tang and Xiaolong Wang, 'Drug-Drug Interaction Extraction via Convolutional Neural Networks', *Computational and Mathematical Methods in Medicine*, **Vol. 2016**, 8 pages, (2016).

[8] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng, 'Semantic Compositionality Through Recursive Matrix-Vector Spaces', in *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (2012).

[2] http://code.google.com/p/word2vec/

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

doi:10.3233/978-1-61499-672-9-1668

Efficient Computation of Deterministic Extensions for Dynamic Abstract Argumentation Frameworks

Sergio Greco and **Francesco Parisi**
Department of Informatics, Modeling, Electronics and System Engineering,
University of Calabria, Italy,
email:{greco, fparisi}@dimes.unical.it

Abstract. We address the problem of efficiently computing the extensions of abstract argumentation frameworks (AFs) which are updated by adding/deleting arguments or attacks. We focus on the two most popular 'deterministic' semantics (namely, *grounded* and *ideal*) and present two approaches for their incremental computation, well-suited to dynamic applications where updates to an initial AF are frequently performed to take into account new available knowledge.

1 Introduction

Typically an argumentation framework (AF) models a temporary situation as new arguments and attacks can be added/removed to take into account new available knowledge. This may change significantly the conclusions that can be derived. For instance, when a new attack is added to an AF, existing attacks may cease to apply and new attacks become applicable. Surprisingly, the definition of evaluation algorithms and the analysis of the computational complexity taking into account such dynamic aspects has been mostly neglected, whereas in these situations incremental computation techniques can greatly improve performance. Sometimes changes to the AF can make small changes to the set of conclusions, and recomputing the whole semantics from scratch can be avoided. For instance, consider the situation shown in Figure 1: the initial AF $\mathcal{A}_0$, where h is not attacked by any other argument, is updated to AF $\mathcal{A}$ by adding attack (g, h). According to the most popular argumentation semantics, i.e. *grounded*, *complete*, *ideal*, *preferred*, *stable*, and *semi-stable* [2, 1], the initial AF $\mathcal{A}_0$ admits the extension $E_0 = \{a, h, g, e, l, m, o\}$, whereas the extension for the updated framework $\mathcal{A}$ becomes $E = \{a, c, g, e, l, m, o\}$. As it will be shown later, for the grounded and ideal semantics, which are deterministic, the extension E can be efficiently computed incrementally by looking only at a small part of the AF, which is "influenced by" the update operation. This part is just $\{h, c\}$ in our example, and we will show that the membership of the other arguments to E does not depend on the update operation, and thus we do not need to compute them again after performing update $+(g, h)$.

In this paper, we introduce the concept of *influenced set* that consists of the arguments whose status could change after an update. The influenced set refines the previously proposed set of *affected arguments* [4] and makes the computation more efficient. Incremental algorithms for recomputing the grounded and ideal extensions can be defined so that they compute the status of influenced arguments only. Experiments reported in the extended version of this paper [3] showed the effectiveness of our approach on real and synthetic AFs.

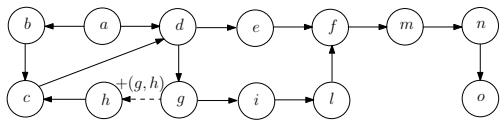


Figure 1. AFs $\mathcal{A}_0$ and $\mathcal{A} = +(g, h)(\mathcal{A}_0)$

2 Influenced Arguments

An *(abstract) argumentation framework* [2] (*AF*) is a pair $\langle A, \Sigma \rangle$, where A is a set of arguments and $\Sigma \subseteq A \times A$ is a set of *attacks*.

We assume the reader is familiar with abstract argumentation concepts, such as *conflict-free* and *admissible* sets of arguments, and argumentation semantics *complete* (`co`), *preferred* (`pr`), *semi-stable* (`ss`), *stable* (`st`), *grounded* (`gr`), and *ideal* (`id`), which specifies the criteria for identifying sets of arguments considered to be "reasonable" together, called *extensions*. Given an AF $\mathcal{A}$ and a semantics $\mathcal{S}$, we use $\mathcal{E}_{\mathcal{S}}(\mathcal{A})$ to denote the set of $\mathcal{S}$-extensions of $\mathcal{A}$. All the above-mentioned semantics except the stable admit at least one extension, and the grounded and ideal admit exactly one extension [2, 1]. Semantics `gr` and `id` are called *deterministic* or *unique status*.

The argumentation semantics can be equivalently defined in terms of *labelling* [1], that is using a total function $L : A \to \{\text{IN}, \text{OUT}, \text{UN}\}$ assigning to each argument a label. Between *complete extensions* and *complete labellings* there is a bijective mapping defined as follows: for each extensions E there is a unique labelling $L = \langle E, E^+, A \setminus (E \cup E^+) \rangle$, where E^+ denotes the set of arguments attacked by arguments in E, and for each labelling L there is a unique extension $in(L)$, that is the set of arguments labeled as IN.

Updates. An *update* u for an AF $\mathcal{A}_0$ consists in modifying $\mathcal{A}_0$ into an AF $\mathcal{A}$ by adding or removing arguments or attacks. *W.l.o.g., we can focus on updates consisting of adding/deleting one attack between arguments belonging to $\mathcal{A}_0$ [3].* We use $+(a, b)$ (resp. $-(a, b)$) to denote the addition (resp. deletion) of an attack (a, b), and $u(\mathcal{A}_0)$ to denote the application of update $u = \pm(a, b)$ to $\mathcal{A}_0$.

In [3], we identified sufficient conditions guaranteeing that a given $\mathcal{S}$-extension is still an $\mathcal{S}$-extension after performing an update. That is, let $\mathcal{A}_0$ be an AF, $u = \pm(a, b)$ an update, $\mathcal{S} \in \{$`co`, `pr`, `ss`, `st`, `gr`, `id`$\}$ a semantics, $E_0 \in \mathcal{E}_{\mathcal{S}}(\mathcal{A}_0)$ an extension of $\mathcal{A}_0$ under $\mathcal{S}$, we identified **conditions for extension (or labelling) preservation** under which it is ensured that $E_0 \in \mathcal{E}_{\mathcal{S}}(u(\mathcal{A}_0))$.

Let $\mathcal{A}_0$ be an AF, $u = \pm(a, b)$ an update, and E_0 an extension of $\mathcal{A}_0$ under a given semantics $\mathcal{S}$. The definition of *influenced set* of u w.r.t. $\mathcal{A}_0$ and E_0 (denoted as $\mathcal{I}(u, \mathcal{A}_0, E_0)$) is built on the following

observations: 1) if one of the conditions for extension (or labelling) preservation holds, then E_0 is still an extension for $u(\mathcal{A}_0)$ and, therefore, $\mathcal{I}(u, \mathcal{A}_0, E_0) = \emptyset$; 2) the status of an argument c can change when the status of argument b changes only if c is reachable from b; 3) if argument z is not reachable from b and its status is IN (i.e., $z \in E_0$), then also the status of the arguments attacked by z cannot change: their status remains OUT. The **influenced set** $\mathcal{I}(u, \mathcal{A}_0, E_0)$ is the set of arguments that can be reached from b without using any intermediate argument y whose status is known to be OUT because it is determined by an argument $z \in E_0$ which is not reachable from b.

Example 1 For the AF $\mathcal{A}_0$ of Figure 1, whose grounded extension is $E_0 = \{a, h, g, e, l, m, o\}$, we have that the influenced set of $u = +(g, h)$ is $\mathcal{I}(u, \mathcal{A}_0, E_0) = \{h, c\}$. Note that $d \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ since it is attacked by $a \in E_0$. Thus the arguments that can be reached only using d cannot belong to $\mathcal{I}(u, \mathcal{A}_0, E_0)$ either. □

3 Recomputing Unique Status Semantics

Grounded Semantics. We first identify the restricted subgraph of the given AF containing the arguments influenced by the update. Given an AF $\mathcal{A}_0$, the grounded extension E_0 for $\mathcal{A}_0$, and an update $u = \pm(a, b)$, the **restricted AF of $\mathcal{A}_0$ w.r.t. E_0 and u for the grounded semantics** (denoted as $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$) contains, in addition to the subgraph of $u(\mathcal{A})$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$, additional nodes and edges containing needed information on the "external context", i.e. information about the status of arguments which are attacking some argument in $\mathcal{I}(u, \mathcal{A}_0, E_0)$. Specifically, if there is in $\mathcal{A}_0$ an edge from node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN, so that, as a does not have incoming edges in $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$, its status is confirmed to be IN. Moreover, if there is in $\mathcal{A}_0$ an edge from a node $e \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $c \in \mathcal{I}(u, \mathcal{A}_0, E_0)$ such that e in UN, we add edge (c, c) to $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$ so that the status of c cannot be IN.

Example 2 Continuing Example 1, $\mathcal{R}_{\mathtt{gr}}(+(g, h), \mathcal{A}_0, E_0)$ consists of the subgraph induced by $\mathcal{I}(u, \mathcal{A}_0, E_0) = \{h, c\}$ and the edge (g, h) which is an attack towards argument $h \in \mathcal{I}(u, \mathcal{A}_0, E_0)$ coming from argument g outside $\mathcal{I}(u, \mathcal{A}_0, E_0)$ labelled as IN. Hence, $\mathcal{R}_{\mathtt{gr}}(+(g, h), \mathcal{A}_0, E_0) = \langle\{g, h, c\}, \{(g, h), (h, c)\}\rangle$. □

Example 3 Consider the AF $\mathcal{A}_0 = \langle\{a, b, c, d, e, f, g\}, \{(a, b), (b, a), (c, d), (d, c), (a, c), (b, c), (f, c), (g, f)\}\rangle$ and the update $u = +(e, d)$. We have that (*i*) the grounded extension of $\mathcal{A}_0$ is $E_0 = \{g, e\}$ (i.e. arguments a, b, c, d are all labeled as UN); (*ii*) the influenced set is $\mathcal{I}(u, \mathcal{A}_0, E_0) = \{c, d\}$; and (*iii*) the restricted AF is $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0) = \langle\{c, d, e\}, \{(c, d), (d, c), (e, d), (c, c)\}\rangle$. □

Our algorithm first checks if the restricted AF (computed w.r.t. update $u = \pm(a, b)$) is empty. If this is the case, then $E = E_0$. Otherwise, the status of arguments in $\mathcal{I}(u, \mathcal{A}_0, E_0)$ is recomputed and the extension E of $u(\mathcal{A}_0)$ is constructed by combining the arguments in E_0 not belonging to the influenced part and the arguments returned by an *incremental fixpoint function* which computes the grounded extension of the restricted AF $\mathcal{A}_d = \langle A_d, \Sigma_d\rangle$ compatible with the starting extension $E_0 \cap A_d$.

Example 4 For the AF $\mathcal{A}_0$ of Figure 1, where $E_0 = \{a, h, g, e, l, m, o\}$ and $u = +(g, h)$, the restricted AF $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0) = \mathcal{A}_d = \langle A_d, \Sigma_d\rangle$ with $A_d = \{g, h, c\}$ and $\Sigma_d = \{(g, h), (h, c)\}$ is computed. As A_d is not empty, the incremental fixpoint function with actual parameters $\mathcal{A}_d$ and $E_0 \cap A_d = \{g, h\}$ is called. The function returns the set $\{g, c\}$, and the algorithm returns $E = \{a, g, e, l, m, o\} \cup \{g, c\}$ where $\{a, g, e, l, m, o\}$ is the set of arguments in E_0 and not belonging to the influenced part. □

Ideal Semantics. Given an AF $\mathcal{A}_0$, the ideal extension E_0 for $\mathcal{A}_0$ and an update $u = \pm(a, b)$, **the restricted AF of $\mathcal{A}_0$ w.r.t. E_0 and u for ideal semantics** (denoted as $\mathcal{R}_{\mathtt{id}}(u, \mathcal{A}_0, E_0)$) contains the paths of $\mathcal{A}_0$, providing the information on the "context" outside the influenced set $S = \mathcal{I}(u, \mathcal{A}_0, E_0)$, that need to be considered to determine the new status of arguments in S. To determine the status of nodes in S we must consider all nodes and attacks occurring in paths (of any length) ending in S whose nodes outside S are all labeled as UN.

Example 5 Continuing Example 3 we have that the ideal extension of $\mathcal{A}_0$ is $E'_0 = \{g, e, d\}$ (i.e. arguments a and b are labeled as UN); and the restricted AF is $\mathcal{R}_{\mathtt{id}}(u, \mathcal{A}_0, E'_0) = \langle\{a, b, c, d, e\}, \{(a, b), (b, a), (c, d), (d, c), (a, c), (b, c), (e, d)\}\rangle$. □

Our algorithm computes the ideal extension E of an updated AF $\mathcal{A} = u(\mathcal{A}_0)$ using the ideal extension E_0 of AF $\mathcal{A}_0$. It starts by identifying the set S of arguments whose status need to be recomputed and the initial set that will be iteratively incremented to obtain the ideal extension. It first checks if the influenced set is empty; in such a case $E = E_0$ and then it stops. If this is not the case, it iterates to reach a fixpoint. At each step it first computes the restricted AF for the grounded semantics and updates the current extension E and the set S of unlabelled arguments by including in E all the arguments belonging to the grounded extension and removing from S the arguments that have been decided. Next, it decides if an unlabelled argument belong to E by searching for a winning strategy for it.

4 Discussion

We presented two incremental approaches for computing deterministic extensions of updated AFs. The algorithms exploit the initial extension of an AF for computing the set of arguments influenced by an update, and for detecting early termination conditions during the recomputation of the status of the arguments. Although we focused on updates consisting of adding/removing one attack, our technique can be extended to deal with the case of multiple updates, as shown in [3], where we reported on experiments conducted considering both single and multiple attack updates. The experiments showed that the incremental computation outperforms the base (non-incremental) computation. The experiments also showed that our definition of influenced set substantially restricts the portion of the AF to be analysed for recomputing the semantics of an AF after performing an update. This means that even using *any non-incremental* algorithm taking as input the restricted AF would result in a performance improvement, since the size of the input data would be significantly smaller.

REFERENCES

[1] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin, 'An introduction to argumentation semantics', *The Knowledge Engineering Review*, **26**(4), 365–410, (2011).

[2] Phan Minh Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games', *Artificial Intelligence*, **77**(2), 321–358, (1995).

[3] Sergio Greco and Francesco Parisi. Efficient computation of deterministic extensions for dynamic abstract argumentation frameworks: Technical report. Available at http://wwwinfo.dimes.unical.it/∼parisi/tr/grecoparisi.htm, 2016.

[4] Bei Shui Liao, Li Jin, and Robert C. Koons, 'Dynamics of argumentation systems: A division-based method', *Artificial Intelligence*, **175**(11), 1790–1814, (2011).

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

doi:10.3233/978-1-61499-672-9-1670

The Post-Modern Homunculus

Eric Neufeld and Sonje Finnestad[1]

Abstract. Throughout the ages, magicians, scientists and charlatans have created life-*like* artifacts, some purported to be intelligent. In one famous case, *the Chess Player,* the intelligence was a little person hidden inside doing the thinking. Analogously, throughout the history of philosophy, and cognition, theories have arisen to explain intelligence in *humans*, but a philosophical problem with many such explanations is that they use what is called a *homunculus* argument – the explanation, upon scrutiny reveals a "little one" (homunculus) in the proposed mental apparatus that is responsible for thinking. For most of the era of computing, the Imitation Game, as so simply yet subtly put forward by Alan Turing, has been considered the gold standard for measuring this mysterious quantity, though recently Hector Levesque has pointedly argued that the time has come to abandon Turing's test for a better one of his own design, which he describes in a series of acclaimed papers. In particular, we argue that Levesque, who has cleverly found the 'homunculus' in the arguments of others, has essentially regressed the problem of intelligence to a homunculus in his own system.

1 INTRODUCTION

Understanding intelligence, artificial or otherwise, continues to challenge humankind. Homunculus arguments slip into many discussions, benignly or unintentionally, and it can take considerable scrutiny to find them. For instance, consider a theory of human vision that notes how the human eye works like a camera, with the lens projecting an upside-down image of the world on the retina, which an internal mechanism in the brain can watch and interpret. (This is a highly oversimplified presentation of the Cartesian Theatre [2].) Here, the homunculus is the aforementioned "internal mechanism" that has solved the problem of vision that the theory proposed to explain.

In a series of articles, several authors, in particular, Hector Levesque, revisit key foundational questions in artificial intelligence. Levesque, although he does not use the term *homunculus,* brilliantly uncovers one [4] in the celebrated Chinese Room thought experiment of John Searle.

Elsewhere [5,6], Levesque goes on to ask whether the Turing Test is obsolete and should be replaced by something else, and poses a very clever constructive alternative, a claim startling enough to make the *New Yorker*. However, a study of Levesque's papers suggests that, one, Levesque's new test also uses (inadvertently) a variation of a homunculus argument (although it was very hard to find), and two, that the Turing test is of a different character than other landmark tests in AI. In light of the special theme of this conference on *AI and Human Values*, we claim that Turing's test transcends formal philosophies of science, whereas other artificial intelligence tests are benchmarks. Judging humanity in any number of ways also transcends science, and so this makes a clear contribution to this central theme of human values.

The following elaborates these points.

2 LEVESQUE AND SEARLE'S CHINESE ROOM

Searle's Chinese Room argument [7] tries to separate outward behavior from true intelligence. "Imagine," he says, "a native English speaker who knows no Chinese locked in a room full of boxes of Chinese symbols (a data base) together with a book of instructions for manipulating the symbols (the program)." People outside the room pass in questions to the English speaker, who mechanically (!) consults his instruction book, retrieves certain answer symbols from the boxes, and returns them to the asker.

If the asker is pretty happy with the answers, the question "Is the behavior of the English speaker *intelligent*?" may be asked.

One camp argues that the English speaker's behavior is intelligent, and if we can build its computational equivalent, why do we care whether it understands what it is doing? This seems obvious. Another camp argues that the behaviour is *not* intelligent, because the English speaker has no idea as to what is being discussed, even if he or she is a fully capable human. This also seem obvious.

Both answers have merit. The first camp is concerned with pragmatics; the second demands a deeper notion of consciousness.

The question can also be asked in a different way. "Is the behaviour of the *system* intelligent?" Putting the question this way, the system includes the English speaker, the Chinese Room full of boxes, and the instruction manual.

Thirty years later, Searle [8] said "Computation is defined purely formally or syntactically, whereas minds have actual mental or semantic contents, and we cannot get from syntactical to the semantic just by having the syntactical operations and nothing else." In other words, "no". Searle's view is that neither the person nor the system is intelligent, because however clever its answers are the system, the system does not understand what it is doing.

Levesque [4] plainly notes, *there is no such instruction book* – mooting the discussion. He goes on to write that "Searle exploits the fact that we do not yet have a clear picture of what a real book for Chinese would have to be like." Why? Because such a book would have to have an answer for every possible question asked in every possible context.

That is, the *instruction book* is a homunculus!

For Searle's Chinese Room to function as described, some one (or thing) would have had to have already solved the problem of language understanding perfectly and incorporated it into an instruction book with a handy index.

[1] Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, Canada, email: eric.neufeld@usask.ca

Levesque takes a different tack from that described in the preceding paragraph, using an illustration he calls the Summation Room, which we do not reproduce for reasons of space.

3 WINOGRAD SCHEMAS AND THE TURING TEST

The genius of the Turing test (or *Imitation Game*) [9] is that it identifies intelligence with perhaps the most ordinary and basic of human activities – having a successful conversation in the estimation of a judge or jury. Turing thus proposed a way to test mechanical intelligence without actually defining it – a brilliant finesse.

Pointing to the Loebner competition, Levesque [5,6] states that the Turing Test "has a serious problem: it relies too much on deception". A program "will either have to be evasive … or manufacture some sort of false identity (and be prepared to lie convincingly)." "All other things being equal," says Levesque, "we should much prefer a test that did not depend on chicanery of this sort". "Is intelligence just a bag of tricks?" he asks. And so on.

To combat such "chicanery", Levesque proposes the Winograd Schema Challenge, which consists of multiple choice questions like this:

Question: The trophy would not fit in the brown suitcase because it was too *big* (*small*). What was too *big (small)?*

Answer 0: the trophy

Answer 1: the suitcase.

Levesque gives good arguments that tests can be constructed that are easy for humans to solve, yet are challenging for machines. Moreover, he provides a handy grading formula.

However, with the act of grading, the WSC becomes another benchmark, rather than a replacement for the Turing test. By 'benchmark', we intend many of the various challenges at which computers have already succeeded: tic-tac-toe, sliding tile puzzles, championship-level checkers, chess, or Go, Jeopardy, poker, and so on. The ability to objectively measure success in terms of win/loss, dollars, or speed allows us to define success in a manner that pays no attention to the machine's performance.

To understand this, consider the televised competition between Watson and two human Jeopardy champions. Watson almost enchanted until it was given, in the category *U.S. Cities*, the clue, "Its largest airport is named for a World War II hero, its second largest for a World War II battle." Watson answered "Toronto".

To the North American audience, this was a hysterical blooper, as most viewers knew Toronto to be in *Canada*, not the United States. To be fair, Watson knew it was guessing. Nonetheless, for many, this dispelled the illusion of intelligence, despite Watson's landslide victory, measured by money won.

The Turing test cannot use objective grading. It only requires that a human judge (or jury) be unable to distinguish the human's performance from the machine's. We believe that the existence of an objective grading scheme to decide intelligence in the WSC is equivalent to saying that a machine has been built that can decide whether another machine's performance on a multiple choice exam is intelligent or not.

There's no such machine! Unless, of course, someone has created one that can decide when another machine is intelligent.

It is interesting that Turing alternately suggested that judges or juries would decide the question of intelligence. There are at least two theories of why juries exist in the legal system [1]. One is that juries of peers tempered the decisions of judges, in the same sense the introduction of the House of Commons tempered decisions of the House of Lords.

The other theory, which [1] argues should be *central*, might be characterized as saying that the idea of justice ultimately resides in the minds of humans.

This brings us back to our earlier statement, where we stated that the Turing test transcends science. Let us be clear that we do not intend to enter the realm of the supernatural when we say this; it is only that in the Knowledge Representation community, it is appropriate to think of science in terms of the formal logical frameworks articulated, for example, by Kyburg [3]. But consensus on what science is has not been achieved there either – there are also frameworks going back to Popper and Quine, and many variations since.

4 CONCLUSIONS

The theme of this conference is *AI and Human Values*, and this work suggests that no computer has been able to impersonate a human and sustain the illusion for a reasonable length of time (despite some claims that the test has been passed), and that the ability to decide the success of such an impersonation remains a uniquely human task, even if computers are succeeding at various benchmarks perhaps sooner than expected.

This paper is a shortened version of [7]; this research was supported by the University of Saskatchewan.

REFERENCES

[1] R.P. Burns, The History and Theory of the American Jury. *California Law Review* 83(6):1477-1494, (1995).

[2] D. C. Dennett, "Darwin's dangerous idea." *The Sciences*, 35.3, 34-40, (1995).

[3] H.E. Kyburg, Jr., *Theory and Measurement*, Cambridge University Press (2009)

[4] H.J. Levesque, 'Is it Enough to get the Behaviour Right?', *Proceedings of IJCAI-09, Pasadena, CA*, Morgan Kaufmann, 1439:1444, (2009).

[5] H.J. Levesque, 'The Winograd Schema Challenge', *Logical Formalizations of Commonsense Reasoning, 2011 AAAI Spring Symposium, TR SS-11-06*, (2011).

[6] H.J. Levesque, 'On our best behaviour', *Artificial Intelligence* 212(1): 27-35, (2014).

[7] E. Neufeld and S. Finnestad, 'The Post-Modern Homunculus', *Proceedings of EGPAI,* to appear.

[8] J. Searle, 'Minds, Brains and Programs', *Behavioral and Brain Sciences*, 3: 417–57, (1980).

[9] J. Searle, 'Why Dualism (and Materialism) Fail to Account for Consciousness', in R. E. Lee, ed., Questioning *Nineteenth Century Assumptions about Knowledge, III: Dualism.* NY: SUNY Press, 2010.

[10] A.M. Turing, "Computing machinery and intelligence." *Mind*, 433-460, (1950).

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

doi:10.3233/978-1-61499-672-9-1672

Symmetric Multi-Aspect Evaluation of Comments
Extended Abstract

Theodore Patkos [1] and **Giorgos Flouris** [2] and **Antonis Bikakis** [3]

1 Introduction

As we explained in [3], most online systems for the Social Web, such as social media, online discussion forums, news sites and product review sites, offer limited support in identifying the helpful comments among the credible ones. We thus proposed the *multi-Dimensional Comment Evaluation* (mDiCE) framework, which relies on methodologies from the field of Computational Argumentation to clearly distinguish between the acceptance (credibility) and the quality (helpfulness) of arguments.

In this paper, we extend this framework and introduce the *symmetric multi-Dimensional Comment Evaluation* (s-mDiCE) framework, which has a more intuitive behavior and a wider scope, aiming to cover also the goal-oriented debates found in decision support systems or in debate portals, e.g., in active citizenship portals. s-mDiCE combines features, such as voting on arguments, expert rating, supporting and attacking arguments, in a unified and adaptable framework that can guarantee intuitive behavior for the user or the moderator who wish to spot important opinions or to rank them for easier processing.

2 Formalization of the s-mDiCE Framework

Our proposal is a generic formal framework that enables the evaluation of thestrength of arguments considering one or more aspects.

Definition 1. *An s-mDiCE (symmetric multi-Dimensional Comment Evaluation) framework is an (N+1)-tuple* $\langle \mathcal{A}, \mathcal{D}^*_{d1}, \ldots, \mathcal{D}^*_{dN} \rangle$*, where* $\mathcal{A}$ *is a finite set of arguments and* $\mathcal{D}^*_{d1}, \ldots, \mathcal{D}^*_{dN}$ *are aspects (dimensions), under which an argument is evaluated.*

Depending on the domain of interest, different aspects can be defined, such as relevancy, reliability, objectivity etc. The ultimate objective of the s-mDiCE framework is the calculation of the quality and acceptance score of each argument (we use the interval $\mathbb{I} = [0, 1]$ as the range of these functions), using the different aspects as different "dimensions" for calculating said scores.

Definition 2. *An aspect* $\mathcal{D}^*_x$ *corresponding to an argument set* $\mathcal{A}$ *is a 5-tuple* $\langle \mathcal{R}^{supp}_x, \mathcal{R}^{att}_x, BS_x, V^+_x, V^-_x \rangle$*, where* $\mathcal{R}^{supp}_x \subseteq \mathcal{A} \times \mathcal{A}$ *is a binary acyclic support relation on* $\mathcal{A}$*,* $\mathcal{R}^{att}_x \subseteq \mathcal{A} \times \mathcal{A}$ *is a binary acyclic attack relation on* $\mathcal{A}$*, and* $BS_x : \mathcal{A} \to \mathbb{I}$*,* $V^+_x : \mathcal{A} \to \mathbb{N}^0$ *and* $V^-_x : \mathcal{A} \to \mathbb{N}^0$ *are total functions mapping each argument to a basic score, a number of positive and a number of negative votes relative to this aspect, respectively.*

[1] FORTH-ICS, Greece, email: patkos@ics.forth.gr
[2] FORTH-ICS, Greece, email: fgeo@ics.forth.gr
[3] University College London, UK, email: a.bikakis@ucl.ac.uk

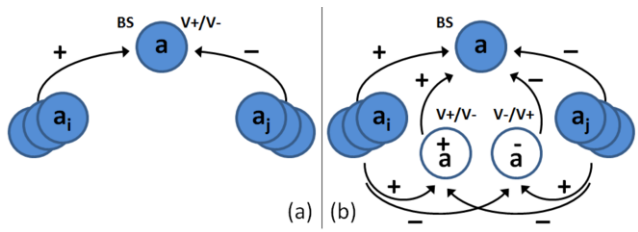


Figure 1. (a) A debate graph with votes, base score and user-generated supporting and attacking arguments, (b) its transformation with black nodes.

The current definition generalizes the one introduced in [3], incorporating also the notion of a *base score* (or intrinsic strength) BS_x, which is often used in decision-making systems (e.g., in [1, 4]) to capture an expert's initial rating over an opinion, before any debate has taken place. This parameter offers an one-time estimation for an argument; other users may then affect the final score of the opinion positively or negatively through their arguments or votes. In other systems, the base score may obtain a more personalized flavor, representing for instance the trust that a user attributes to the user who issues an argument regardless of its content.

Discussion forums and review sites on the Web, on the other hand, rely on a different rating scheme, enabling their users to express their (unjustified) stance towards comments through various voting mechanisms, such as positive/negative votes, like/dislike counters, star-based rating mechanisms etc. Many methods have been suggested to aggregate such ratings having different degrees of confidence.

Some systems treat votes as a base score, yet the conceptual difference between the two is important. Our proposed framework distinguishes between static base scores and dynamic ratings. It also considers the credibility that votes carry, given their the interplay of the underlying supporting or attacking arguments.

In s-mDiCE, we reduce votes to the argument level and convert them to arguments by assuming that they express an opinion in favor or against the target argument. As these opinions carry no content on their own, we call them *supporting* and *attacking blank arguments*. We can estimate the credibility of these opinions more accurately by adding additional attacks (supports) from arguments that support (resp. attack) the target argument to its attacking blank argument, and additional supports (resp. attacks) to its supporting blank argument. Fig. 1 presents the simple argumentation graph that shows these interactions.

Before formally defining blank arguments, some convenient notation is needed. We let $\bar{\mathcal{A}}$ denote the set of user-generated arguments and use $\mathring{\mathcal{A}}$ to refer to the set of blank arguments of an s-

mDiCE framework $\mathcal{F}$, such that $\mathcal{A} = \mathring{\mathcal{A}} \cup \widetilde{\mathcal{A}}$. Moreover, given an aspect $\mathcal{D}^*_x = \langle \mathcal{R}^{supp}_x, \mathcal{R}^{att}_x, BS_x, V^+_x, V^-_x \rangle$, we define the set of direct supporters of an argument $a \in \mathcal{A}$ as $\mathcal{R}^+_x(a) = \{a_i : (a_i, a) \in \mathcal{R}^{supp}_x\}$. Similarly, the set of direct attackers of a is defined as $\mathcal{R}^-_x(a) = \{a_i : (a_i, a) \in \mathcal{R}^{att}_x\}$.

Definition 3. *Let $\mathcal{F}$ be an s-mDiCE framework and $\mathcal{D}^*_x = \langle \mathcal{R}^{supp}_x, \mathcal{R}^{att}_x, BS_x, V^+_x, V^-_x \rangle$ be an aspect of $\mathcal{F}$. For each argument $a \in \widetilde{\mathcal{A}}$, we define two new arguments $\overset{+}{a}$ and $\overline{a}$, called the supporting and attacking blank argument of a respectively, such that*

- $(\overset{+}{a}, a) \in \mathcal{R}^{supp}_x$,
- $V^+_x(\overset{+}{a}) = V^+_x(a)$, $V^-_x(\overset{+}{a}) = V^-_x(a)$,
- *for all* $(a_i, a) \in \mathcal{R}^{supp}_x$ *it also holds that* $(a_i, \overset{+}{a}) \in \mathcal{R}^{supp}_x$,
- *for all* $(a_j, a) \in \mathcal{R}^{att}_x$ *it also holds that* $(a_j, \overset{+}{a}) \in \mathcal{R}^{att}_x$, *and similarly*
- $(\overline{a}, a) \in \mathcal{R}^{att}_x$,
- $V^+_x(\overline{a}) = V^-_x(a)$, $V^-_x(\overline{a}) = V^+_x(a)$,
- *for all* $(a_i, a) \in \mathcal{R}^{supp}_x$ *it also holds that* $(a_i, \overline{a}) \in \mathcal{R}^{att}_x$,
- *for all* $(a_j, a) \in \mathcal{R}^{att}_x$ *it also holds that* $(a_j, \overline{a}) \in \mathcal{R}^{supp}_x$.

There is a inherent symmetry in the model, which is the main differentiation to mDiCE, where the blank argument is only used as a means to express the positive stance of votes towards an argument (e.g., its social support [2]). We next introduce a set of generic functions that help assess the different quantities affecting the strength of an argument.

Definition 4. *The generic score function $g^{vot} : \mathbb{N}^0 \times \mathbb{N}^0 \to \mathbb{I}$ aggregates the positive and negative votes into a single strength score.*

Definition 5. *The generic score function $g^{set} : (\mathbb{N}^0)^{\mathbb{I}} \to \mathbb{I}$ aggregates the strength of a set of supporting or attacking arguments into a single strength score.*

We apply these functions, in order to estimate both the acceptance and the quality score of an argument. For the former, we need to define $s^{dlg}_x()$, which reflects the credibility of an argument by combining the strength of its supporting and attacking arguments, i.e., the outcome of the dialogue that it generated on a given aspect. For the latter, we define $s^{cng}_x()$ to characterize the congruence strength, i.e., the degree of people's compliance with an argument with respect to its clarity and justification.

Definition 6. *Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{D}^*_{d1}, ..., \mathcal{D}^*_{dN} \rangle$ be an s-mDiCE framework and $\mathcal{D}^*_x = \langle \mathcal{R}^{supp}_x, \mathcal{R}^{att}_x, BS_x, V^+_x, V^-_x \rangle$ be an aspect of $\mathcal{F}$. The dialogue strength $s^{dlg}_x : \mathcal{A} \to \mathbb{I}$ of an argument $a \in \mathcal{A}$ over aspect $\mathcal{D}^*_x$ is given by*

$$s^{dlg}_x(a) = g^{dlg}(G^x_1(a), g^{set}(\{s^{dlg}_x(a_i) : a_i \in \mathcal{R}^+_x(a)\}), g^{set}(\{s^{dlg}_x(a_j) : a_j \in \mathcal{R}^-_x(a)\})) \quad (1)$$

where

$$G^x_1(a) = \begin{cases} BS_x(a) & \text{, if } a \in \widetilde{\mathcal{A}} \\ g^{vot}(V^+_x(a), V^-_x(a)) & \text{, if } a \in \mathring{\mathcal{A}} \end{cases}$$

and function $g^{dlg} : \mathbb{I} \times \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ is a generic score function valuating the dialogue strength of an argument for a given aspect, considering the aggregation of the strength of the votes or base score, the supporting and the attacking arguments.

Definition 7. *Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{D}^*_{d1}, ..., \mathcal{D}^*_{dN} \rangle$ be an s-mDiCE framework and $\mathcal{D}^*_x = \langle \mathcal{R}^{supp}_x, \mathcal{R}^{att}_x, BS_x, V^+_x, V^-_x \rangle$ be an aspect of $\mathcal{F}$. The congruence strength $s^{cng}_x : \mathcal{A} \to \mathbb{I}$ of an argument $a \in \mathcal{A}$ over aspect $\mathcal{D}^*_x$ is given by*

$$s^{cng}_x(a) = g^{cng}(G^x_2(BS_x(a), g^{vot}(V^+_x(a), V^-_x(a))), g^{set}(\{s^{dlg}_x(a_i) : a_i \in \mathcal{R}^+_x(a) \cap \widetilde{\mathcal{A}}\}), g^{set}(\{s^{dlg}_x(a_j) : a_j \in \mathcal{R}^-_x(a) \cap \widetilde{\mathcal{A}}\})) \quad (2)$$

where $G^x_2 : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ is a generic function combining the strength of the base score of an argument and its votes, and $g^{cng} : \mathbb{I} \times \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ is a generic score function valuating the congruence score of an argument, considering the aggregation of the strength of the votes, the supporting and the attacking arguments.

Notice that $g^{cng}()$ only considers the strength of user-generated arguments; in a sense, it calculates the strength of the supporting blank argument, even though one may choose to instantiate it in a different way. For example, typically, $g^{cng}(x_v, x_s, x_a)$ should lay more emphasis on x_v and x_a, increasing on x_v and decreasing on x_a, as, arguably, the "ideal" comment would attract only positive votes and no supporting arguments. That is, in an ideal setting, supporting arguments are only asserted to add information or to explain better the opinion stated, thus denoting a sense of dissatisfaction related to the quality of the target argument. However, we keep the function generic to allow its instantiation to vary from system to system.

Finally, by considering the strength of all aspects defined within a particular s-mDiCE framework, the main scores (quality, acceptance) of an argument can be determined.

Definition 8. *Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{D}^*_{d1}, ..., \mathcal{D}^*_{dtN} \rangle$ be an s-mDiCE framework. The quality and acceptance scores of an argument $a \in \mathcal{A}$ is given by the functions $\mathcal{QUA} : \mathcal{A} \to \mathbb{I}$ and $\mathcal{ACC} : \mathcal{A} \to \mathbb{I}$, respectively, which aggregate the strength of its aspects, such that*

$$\mathcal{QUA}(a) = g^{\mathcal{QUA}}(s^{cng}_{d1}(a), ..., s^{cng}_{dN}(a)) \quad (3)$$

$$\mathcal{ACC}(a) = g^{\mathcal{ACC}}(s^{dlg}_{d1}(a), ..., s^{dlg}_{dN}(a)) \quad (4)$$

with $g^{\mathcal{QUA}}, g^{\mathcal{ACC}} : \mathbb{I}^N \to \mathbb{I}$.

As is obvious from the above, our framework is generic enough to allow many different types of functions to be defined. It is therefore important to define adequate properties for such functions, which would guarantee a "reasonable" behaviour for the task at hand. Defining and proving these properties for specific instantiations of the model is part of our future work.

REFERENCES

[1] Pietro Baroni, Marco Romano, Francesca Toni, Marco Aurisicchio, and Giorgio Bertanza, 'Automatic evaluation of design alternatives with quantitative argumentation', *Argument & Computation*, **6**(1), 24–49, (2015).

[2] João Leite and João Martins, 'Social abstract argumentation', in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pp. 2287–2292. AAAI Press, (2011).

[3] Theodore Patkos, Antonis Bikakis, and Giorgos Flouris, 'A multi-aspect evaluation framework for comments on the social web', in *In Proceedings of the 15^{th} International Conference on Principles of Knowledge Representation and Reasoning (KR-16)*, (2016).

[4] Antonio Rago, Francesca Toni, Marco Aurisicchio, and Pietro Baroni, 'Discontinuity-free decision support with quantitative argumentation debates', in *Proceedings of the 15^{th} International Conference on Principles of Knowledge Representation and Reasoning (KR-16)*, (2016).

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

doi:10.3233/978-1-61499-672-9-1674

An Efficient and Expressive Similarity Measure for Relational Clustering Using Neighbourhood Trees

Sebastijan Dumančić and **Hendrik Blockeel**[1]

Abstract. Clustering is an underspecified task: there are no universal criteria for what makes a good clustering. This is especially true for relational data, where similarity can be based on the features of individuals, the relationships between them, or a mix of both. Existing methods for relational clustering have strong and often implicit biases in this respect. In this paper, we introduce a novel similarity measure for relational data. It is the first measure to incorporate a wide variety of types of similarity, including similarity of attributes, similarity of relational context, and proximity in a hypergraph. We experimentally evaluate how using this similarity affects the quality of clustering on very different types of datasets. The experiments demonstrate that (a) using this similarity in standard clustering methods consistently gives good results, whereas other measures work well only on datasets that match their bias; and (b) on most datasets, the novel similarity outperforms even the best among the existing ones.

1 Introduction

In relational learning, the data set contains instances with relationships between them. Standard learning methods typically assume data are i.i.d. (drawn independently from the same population) and ignore the information in these relationships. Relational learning methods do exploit that information, and this often results in better performance. Much research in relational learning focuses on supervised learning [2] or probabilistic graphical models [4]. Clustering, however, has received less attention in the relational context.

Clustering is an underspecified learning task: there is no universal criterion for what makes a good clustering, thus it is inherently subjective. This is known for i.i.d. data [3], and even more true for relational data. Different methods for relational clustering have very different biases, which are often left implicit; for instance, some methods represent the relational information as a graph (which means they assume a single binary relation) and assume that similarity refers to proximity in the graph, whereas other methods take the relational database stance, assuming typed objects that may participate in multiple, possibly non-binary, relationships.

In this paper, we propose a very versatile framework for clustering relational data. It views a relational dataset as a hypergraph with typed vertices, typed hyperedges, and attributes associated to the vertices. This view is very similar to the viewpoint of relational databases or predicate logic. The task we consider, is: cluster the vertices of one particular type. What distinguishes our approach from other approaches is that the concept of similarity used here is very broad. It can take into account attribute similarity, similarity of the relations an object participates in (including roles and multiplicity), similarity of the neighbourhood (in terms of attributes, relationships, or vertex identity), and interconnectivity or graph proximity of the objects being compared. We experimentally show that this framework for clustering is highly expressive and that this expressiveness is relevant, in the sense that on a number of relational datasets, the clusters identified by this approach coincide better with predefined classes than those of existing approaches.

[1] KU Leuven, Belgium, email: {sebastijan.dumancic, hendrik.blockeel}@cs.kuleuven.be

2 Type-based clustering over neighbourhood trees

2.1 Hypergraph Representation

Relational learning encompasses multiple paradigms. Among the most common ones are the *graph* view, where the relationships among instances are represented by a graph, and the *predicate logic* or equivalently *relational database* view, which typically assumes the data to be stored in multiple relations, or in a knowledge base with multiple predicates. Though these are in principle equally expressive, in practice the bias of learning systems differs strongly depending on which view they take. For instance, *shortest path distance* as a similarity measure is much more common in the graph view than in the relational database view. In the purely logical representation, however, no distinction is made between the constants that identify a domain object, and constants that represent the value of one of its features. Identifiers have no inherent meaning, as opposed to feature values.

In this work, we introduce a new view that combines elements of both. This view essentially starts out from the predicate logic view, but changes the representation from a purely logical one to a hypergraph representation. Formally, the data structure that we assume in this paper is a typed, labelled hypergraph $H = (V, E, \tau, \lambda)$ with V being a set of vertices, and E a set of hyperedges; each hyperedge is an ordered set of vertices. The type function τ assigns a type to each vertex and hyperedge. The set of all vertex types is denoted as T_V, whereas T_E denotes the set of all hyperedge types. A set of attributes $A(t)$ is associated with each $t \in T_V$. The labelling function λ assigns to each vertex a vector of values, one for each attribute of $A(\tau(v))$. If $a \in A(\tau(v))$, we denote $a(v)$ the value of a in v.

Consider a knowledge base, with its Herbrand universe (the set of all constants) partitioned into domain constants and feature values. From this we can infer a typed, labelled hypergraph as follows. The set of vertices V equals the set of domain constants. We assume all constants are typed. Each vertex has a type $t(v)$ that equals the type of the domain constant. For each fact $\mathtt{p(d, v_1, v_2, \ldots, v_n)}$, with d being a domain constant of type p and v_i feature values, there

is a vertex d with type p and attribute vector $(v_1, \ldots, v_m)$. For each fact `r(d1, d2, ..., dm)`, with d_i domain constants, there is a hyperedge $(d_1, d_2, \ldots, d_m)$ with type r. Hyperedges are ordered sets, and do not have attributes[2].

The clustering task we consider is the following: given a vertex type $t \in T_V$, partition the vertices of this type into clusters such that vertices in the same cluster tend to be similar, and vertices in different clusters dissimilar, for some subjective notion of similarity. In practice, it is of course not possible to use a subjective notion; one uses a well-defined similarity function, which hopefully in practice approximates well the subjective notion that the user has in mind. To be able to capture several interpretations of relational similarity, such as attribute or neighbourhood similarity, we represent each vertex with a *neighbourhood tree* - a structure that effectively describe a vertex and its neighbourhood.

2.2 Neighbourhood tree

Consider a vertex v. A neighbourhood tree aims to compactly represent the neighbourhood of the vertex v and all relationships it forms with other vertices, and it is defined as follows. For every hyperedge E in which v participates, add a directed edge from v to each vertex $v' \in E$. Label each vertex with its attribute vector. Label the edge with the hyperedge type and the position of v in the hyperedge (recall that hyperedges are ordered sets). The vertices thus added are said to be at depth 1. If there are multiple hyperedges connecting vertices v and v', v' is added each time it is encountered. Repeat this procedure for each v' on depth 1. The vertices thus added are at depth 2. Continue this procedure up to some predefined depth d. The root element is never added to the subsequent levels.

2.3 Similarity measure

Given the definition above, comparing two vertices is achieved by comparing their neighbourhood trees. The main motivation behind the proposed similarity measure is to capture the distribution of different elements in the neighbourhood of a vertex, the elements being attributes of vertices in the neighbourhood, their identities and hyperedges. Given that in general a vertex participates in a non-fixed number of relations, where neighbours are described by (often overlapping) sets of attributes and their relations to other vertices, distributions are necessary to reliably model the neighbourhood of a vertex. Furthermore, comparing distributions avoids issues arising when comparing the vertices with very different number of neighbours (for example, properly normalizing for such scenario).

The similarity measure we propose relies on the similarity measure between multisets extracted from the corresponding neighbourhood tree. The multiset can be seen as features of the neighbourhood trees. For any neighbourhood tree g, let $B_l(g)$ be the multiset of vertices at depth l in g, and $B_{l,t}(g)$ the multiset of vertices of type t at depth l in g. All the vertices in $B_{l,t}$ have the same attributes, and each vertex assigns one value to each attribute; thus, for each attribute a, a multiset of values $B_{l,t,a}(g)$ is obtained. Let $B_{l,e}(g)$ be the multiset of labels of edges originating at vertices at depth l. Let $\mathcal{N}$ be the set of all neighbourhood trees corresponding to the vertices of interest in a hypergraph.

This way, the similarity is reduce to comparing different multisets. In principle, any measure over multisets of elements can be used, however, in our experiments, we chose to use the χ^2-distance between multisets [5], which is defined as:

$$d(A, B) = \sum_{x \in A \cup B} \frac{(f_A(x) - f_B(x))^2}{f_A(x) + f_B(x)} \quad (1)$$

where A and B are multisets and $f_S(x)$ is the relative frequency of element x in multiset S (e.g., for $A = \{a, b, b, c\}$, $f_A(a) = 0.25$ and $f_A(b) = 0.5$).

The final similarity measure consists of a linear combination of different interpretations of similarity, reflected in the content of the multisets. Concretely, the similarity measure is a composition of components reflecting:

1. attributes of the root vertices,
2. attributes of the neighbouring vertices,
3. proximity of the vertices,
4. identity of the neighbouring vertices,
5. distribution of hyperedge types in a neighbourhood.

Each component is weighted by the corresponding weight w_i. These weights allow one to formulate an interpretation of the similarity between relational objects.

This formulation is somewhat similar to the *multi-view clustering* [1], with each of the components forming a different view on data. However, there is one important fundamental difference: multi-view clustering methods want to find clusters that are good in each view separately, whereas our components do not represent different views on the data, but different potential biases, which jointly contribute to the similarity measure.

2.4 Results

We compared the proposed similarity measure against a wide range of existing relational clustering approaches and graph kernels on five datasets. The proposed similarity measure was used in conjunction with spectral and hierarchical clustering algorithms. We found that, on each separate dataset, our approach performs at least as well as the best competitor, and it is the only approach that achieves good results on all datasets. Furthermore, the results suggest that decoupling different sources of similarity into a linear combination helps to identify relevant information and reduce the effect of noise.

Acknowledgements

Research supported by Research Fund KU Leuven (GOA/13/010)

REFERENCES

[1] Steffen Bickel and Tobias Scheffer, 'Multi-view clustering', in *Proceedings of the Fourth IEEE International Conference on Data Mining*, ICDM '04, pp. 19–26, Washington, DC, USA, (2004). IEEE Computer Society.

[2] Luc De Raedt, *Logical and relational learning*, Cognitive Technologies, Springer, 2008.

[3] Vladimir Estivill-Castro, 'Why so many clustering algorithms: A position paper', *SIGKDD Explor. Newsl.*, **4**(1), 65–75, (June 2002).

[4] Lise Getoor and Ben Taskar, *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2007.

[5] Haifeng Zhao, Antonio Robles-Kelly, and Jun Zhou, 'On the use of the chi-squared distance for the structured learning of graph embeddings', in *Proceedings of the 2011 International Conference on Digital Image Computing: Techniques and Applications*, DICTA '11, pp. 422–428, Washington, DC, USA, (2011). IEEE Computer Society.

[2] A fact of the form `p(d1, ..., dn, v1, ..., vm)` can always be written as `r(d1, ..., dn, h)` and `q(h,v1, ..., vm)`, where **h** denotes a *hyperedge identity*

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

doi:10.3233/978-1-61499-672-9-1676

On Inconsistency Measuring and Resolving

Said Jabbour [1]

Abstract. In dealing with the field of inconsistency, measuring and resolving conflicts are two interesting concepts allowing to make a better reasoning. In this paper, we address these two essential questions. Firstly, we propose a new framework for inconsistency handling based on independent sets of MUSes. In fact, we propose to rank the knowledge bases using a sequence of number that represents independent sets of MUSes. Such sequence is able to catch more finely the structure of MUSes. Moreover, an inconsistency measure is instantiated based on such sequence. Secondly, we address the question of inconsistency resolving through hitting set deletion. In this case we show the interesting points of our new approach which selects hitting sets covering minimum subset of MUSes. Indeed, a framework is defined for the computation of such hitting sets. Furthermore, a derived underlying decision problem is defined. A special case of such decision problem is known to be defined in the literature as SIM-UNSAT and its complexity is still open.

1 Preliminaries

Throughout this paper a finite propositional language $\mathcal{L}$ is assumed. . Let $(a, b, \ldots)$ be a set of propositional symbols and $(\alpha, \beta, \gamma, \ldots)$ be a set of formulas from $\mathcal{L}$. The symbol $\bot$ denotes the constant true and $\top$ its negation.

A Knowledge base K is a finite set of propositional formulas. K is *inconsistent* if $K \vdash \bot$, where $\vdash$ is the classical inference (consequence) relation. In this case, the inconsistency is generally caused by subsets of K representing minimal conflicts and called Minimal Inconsistent (Unsatisfiable) Subsets (Definition 1).

Definition 1. *M is a* Minimal Inconsistent Subset *(MUS) of K iff $M \vdash \bot$ and $\forall M' \subsetneq M$, $M' \nvdash \bot$. The set of all minimal inconsistent subsets of K is denoted $MUSes(K)$.*

Definition 2 (Hitting Set)**.** *H is a hitting set of a collection of sets F if $\forall S \in F$, $H \cap S \neq \emptyset$. A hitting set H is irreducible if there is no other hitting set H', s.t. $H' \subset H$. HitS(F) denotes the set of hitting sets of F.*

Inconsistency measures I associate non-negative real number to knowledge bases. These measures are considered as inconsistency degree functions. Well-founded properties have been proposed [3, 4, 1] allowing to build strong frameworks to reason with inconsistency.

We introduce a non exhaustive list of such properties which were introduced since many years:

1. *Consistency:* $I(K) = 0$ iff K is consistent.
2. *Monotonicity:* if $K \subseteq K'$, then $I(K) \leq I(K')$.
3. *Independence:* $I(K \cup \{\alpha\}) = I(K)$ if $\alpha \in free(K \cup \{\alpha\})$.
4. *MinInc:* $I(M) = 1$ if $M \in MUSes(K)$.
5. *Additivity:* $I(K_1 \cup \ldots \cup K_n) = \Sigma_{i=1}^{n} I(K_i)$ if $MUSes(K_1 \cup \ldots \cup K_n) = MUSes(K_1) \uplus \ldots \uplus MISes(K_n)$, where $\uplus$ is the multi-set union over sets.
6. *Ind-Additivity:* $I(K_1 \cup \ldots \cup K_n) = \Sigma_{i=1}^{n} I(K_i)$ if $\{K_1, \ldots, k_n\}$ are independent knowledge bases ($MUSes(K_1 \cup \ldots \cup K_n) = \biguplus_{i=1}^{n} MUSes(K_i)$, and $unfree(K_i) \cap unfree(K_j) = \emptyset$, for all $i \neq j$).
7. *Super Additivity:* If $K_1 \cap K_2 = \emptyset$, $I(K_1) + I(K_2) \leq I(K_1 \cup K_2)$

[1] CRIL, Univ. Artois & CNRS, F-62300 Lens, France, email: {jabbour@cril.fr}

2 Independent Set Based Inconsistency Ranking

In this section, we propose to deal with structure based inconsistency measures throughout Independent set of MUSes.

Definition 3 (Independent Set of MUSes)**.** *Let $S \subseteq MUSes(K)$. S is an independent set of $MUSes(K)$ if for all $M, M' \in S$ s.t. $M \neq M'$ we have $M \cap M' = \emptyset$.*

Independent Set of MUSes can be seen as the classical independent Set defined in graph theory, where each vertex represents a MUS and an edge is added between two vertices namely M_1 and M_2, if M_1 and M_2 share a formula.

In order to catch as widely as possible the structure of MUSes, all set of independent sets are considered and represented as a sequence.

Definition 4 (Independent MUSes Sequence)**.** *We define the inconsistency sequence associated to K as $im_s(K) = \langle \delta_0 \ldots \delta_{m_K} \rangle$ where δ^k is the number of independent sets of size k and m_K the maximum cardinality value of independent sets of MUSes of K.*

Example 1. *Let $K_1 = \{p \wedge q, \neg q, \neg p, p \wedge s, p \wedge r, \neg r\}$ and $K_2 = \{\neg p, p \wedge q, p \wedge r, \neg q, \neg r, q \vee r\}$. We have $im_s(K_1) = \langle 1\ 5\ 5\ 1 \rangle$ and $im_s(K_2) = \langle 1\ 5\ 5 \rangle$.*

In order to compare knowledge bases, it will be convenient to assume an ordering relation $\preceq$ over their associated numeric sequences.

Definition 5. *Let $u = \langle u_1 \ldots u_n \rangle, v = \langle v_1 \ldots v_m \rangle \in \mathbb{N}^n \times \mathbb{N}^m$ be two sequences. Then, u is less preferred than v, denoted by $u \preceq v$, iff $u = v$ or $n < m$ or there exists $k \leq m$ s.t. $u_k < v_k$ and $u_{k'} = v_{k'}$ for all $k' > k$. Furthermore, $u \prec v$ iff $u \preceq v$ and $u \neq v$.*

Now, we can compare knowledge bases according to their inconsistency sequence as defined below.

Definition 6. *K is less inconsistent than K' iff $im_s(K_2) \preceq im_s(K_1)$.*

According to Definition 6, the relation $\preceq$ privileges knowledge bases having sparse set of MUSes.

Example 2. *Let us consider the knowledge bases of Example 1. We have:* $im_s(K_2) \preceq im_s(K_1)$.

Let us provide some results for particular knowledge bases.

- If MUSes(K) are pairwise disjoints, then $im_s(K) = \langle \binom{n}{0} \binom{n}{1} \ldots \binom{n}{n} \rangle = \langle 1 \ n \ \ldots \frac{n!}{k!(n-k)!} \ldots 1 \rangle$ and it is the maximum possible sequence according to $\preceq$ order over sequences when n is fixed.
- $im_s(K)$ reachs its minimum value for MUSes that pairwise intersect. This is the case if there exists $S \subset K$ s.t. $M_i \cap M_j = S$ for all $i \neq j$ and in this case we have $im_s(K) = \langle 1 \ n \rangle$.

In the sequel, we associate to a knowledge base K the polynomial as stands in Definition 7.

Definition 7. *Let K be a knowledge base s.t., $im_s(K) = \langle a_0 \ldots a_n \rangle$. We define the polynomial function associated to K as: $\Lambda_K(x) = \sum_{i=0}^{n} a_k x^i$*

We define f^s as the function that associates to each polynomial $p(x) = \sum_{i=0}^{n} a_i x^i$, the sequence of its factors $\langle a_0 \ \ldots \ a_n \rangle$.

According to Definition 7, we have $im_s(K) = f^s(\Lambda_K)$

Now, we discuss the behavior of im_s when merging independent knowledge bases.

Proposition 1. *Let K_1 and K_2 be two independent knowledge bases, then $im_s(K_1 \cup K_2) = f^s(\Lambda_{K_1} \times \Lambda_{K_2})$*

According to Proposition 1, if K_1 and K_2 are independent, the factors of $im_s(K_1 \cup K_2)$ resulting by multiplying the polynomial (with integers factors) of K_1 and K_2.

Proposition 2. *im_s satisfies the monotony and super-additivity.*

In the sequel, we derive a new inconsistency measure based on im_s sequence.

Definition 8. *Let $im_s(K) = \langle a_0 \ a_1 \ \ldots a_n \rangle$. We define the inconsistency measure of K as: $Inc_{is}(K) = log(\sum_{k=0}^{n} a_k)$*

Proposition 3. *Inc_{is} satisfies the Consistency, Independence, MinInc, Monotony, and Ind-Additivity. Furthermore, if MUSes(K) are pairwise disjoint, then $Inc_{is}(K) = n$.*

By using logarithm function, the multiplication of im_s over independent knowledge bases is translated into additivity of Inc_{is}.

Let us now provide a lower bound of Inc_{is}.

Proposition 4. $m_K \leq Inc_{is}(K)$

3 Inconsistency Resolving

Measuring inconsistency aims to have an idea on how conflicts are structured inside a knowledge base. Often, such conflicts have to be solved. Indeed, recovering consistency by formulas deletion is one of the most used approach. Usually, the goal is to derive a consistent base by performing the minimal change in the original knowledge base. However, removing an arbitrary hitting set of minimum size can lead to the suppression of useful information. To illustrate such a case, let us consider the knowledge base $K = \{p, \neg p \wedge r, \neg r \wedge q, \neg q\}$ where MUSes are respectively $M_1 = \{p, \neg p \wedge r\}$, $M_2 = \{\neg p \wedge r, \neg r \wedge q\}$, and $M_3 = \{\neg r \wedge q, \neg q\}$.

The minimum cardinality hitting sets are $H_1 = \{p, \neg r \wedge q\}$, $H_2 = \{\neg p \wedge r, \neg q\}$, and $H_3 = \{\neg p \wedge r, \neg r \wedge q\}$. Removing each hitting set give rise to a different consistent knowledge base. However, among these hitting sets, H_3 is a special one since removing its formulas have the following results (1) The resolution of the conflict M_2 two times (2) The complete deletion of M_2, and therefore any information about r is lost. In this particular case, H_1 and H_2 are more suitable for conflicts resolution in K. Using H_3 to resolve inconsistency is clearly a bad choice for the reasons cited above and since there exists other alternatives to resolve conflict in this particular base allowing to keep more information of the initial base.

In the sequel, we tackle the problem of inconsistency resolving through formulas deletion. Our preferred repairs are those aiming to avoid as possible complete deletion of MUSes of K.

Let us first start by defining the cover of a hitting setting as stands below.

Definition 9. *Let H a hitting set of a knowledge K. We define the cover of H as: $C_{mus}(H) = \{M \in MUSes(K) \mid M \subseteq H\}$*

H is said to cover each MUS of $C_{mus}(H)$ and $C_{mus}(H)$ represents the set of MUSes(K) covered by H. Let us note that $0 \leq |C_{mus}(H)| \leq |MUSes(K)|$.

Example 3. *Let $K = \{\neg t, \neg s \wedge t, p \wedge s, \neg s, \neg p, p \wedge q, p \wedge r\}$. $H = \{\neg s \wedge t, p \wedge s, \neg p\}$ is a hitting set and $C_{mus}(H) = \{M_1, M_3\}$ where $M_1 = \{\neg s \wedge t, p \wedge s\}$ and $M_2 = \{p \wedge s, \neg p\}$.*

In order to select best hitting sets, we define an ordering relation over the hitting sets of K as stands before.

Definition 10. *Let K be a knowledge base and $H, H' \in HitS(MUSes(K))$. $H \leq_c H'$ iff $|C_{mus}(H)| \leq |C_{mus}(H')|$*

Now we introduce our hitting set based inconsistency resolving.

Definition 11. *Let K be a knowledge base. Minimum MUS Cover (SHitCoverMin in short) is defined as the problem of finding a smallest hitting set that covers the minimum number of MUSes.*

Solutions of SHitCoverMin can be represented as $min(\leq_s, min(\leq_c, HitS(MUSes(K))))$, where $H \leq_s H'$ for $H, H' \in HitS(MUSes(K))$ iff $|H| \leq |H'|$ and are regarded in priority to restore consistency.

Example 4. *Let us reconsider the Example 3. The hitting set $H = \{\neg p, \neg s \wedge t, \neg s\}$ is a solution of SHitCoverMin problem.*

Let us first start by considering an underlying decision problem associated to $min(\leq_c, HitS(MUSes(K)))$ noted *HitCoverEMUS(k, MUSes(K))*.

Definition 12. *HitCoverEMUS(k, MUSes(K)) is the problem of deciding if there exists a hitting set covering exactly k MUSes of K.*

Proposition 5. *HitCoverEMUS(0, MUSes(K)) is computationally equivalent to SIM-UNSAT [2].*

Note that HitCoverEMUS, HitCoverEMUS, and SHitCoverMin problems can be encoded as SAT and MaxSAT problems.

REFERENCES

[1] Philippe Besnard, 'Revisiting postulates for inconsistency measures', in *JELIA*, pp. 383–396, (2014).

[2] Thomas Eiter, 'Open problems in satisfiability', in *http://www.ece.uc.edu/ franco/Sat-workshop/sat-workshop-open-problems.html*, (1996).

[3] Anthony Hunter and Sébastien Konieczny, 'On the measure of conflicts: Shapley inconsistency values', *Artif. Intell.*, **174**(14), 1007–1026, (2010).

[4] Said Jabbour, Yue Ma, and Badran Raddaoui, 'Inconsistency measurement thanks to MUS-decomposition', in *AAMAS*, pp. 877–884, (2014).

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

doi:10.3233/978-1-61499-672-9-1678

A Fuzzy Semantic CEP Model for Situation Identification in Smart Homes

Amina Jarraya [1] and **Nathan Ramoly** [2 4] and **Amel Bouzeghoub**[2] and **Khedija Arour**[3] and **Amel Borgi**[1] and **Béatrice Finance**[4]

Abstract. Uncertainty is an essential issue for smart home applications. Events generated from sensors can be outdated, inaccurate, imprecise or in contradiction with other ones. These unreliable data can lead to dysfunction in smart home applications. To tackle these challenges, we propose a new model named FSCEP (Fuzzy Semantic Complex Event Processing) that integrates fuzzy logic paradigm, semantic features through an ontology and traditional CEP. We confronted FSCEP with other works tackling uncertainty for CEP and experimented it through simulation with early but promising result.

1 Introduction

Event-driven systems are highly depending on the quality of detection and processing of events. CEP is considered to be a key tool to analyze events from multiple sensors in order to generate high-level events (complex event).

In everyday life environment, in particular smart homes, dealing with uncertainty in CEP is a serious issue. In fact, we should expect sensors to provide non-perfect data. In the literature, uncertainty has multiple dimensions [8]. In this work, we aim to deal with four of them: freshness, precision, accuracy and contradiction. We propose to enrich the traditional CEP with semantic and fuzzy properties. Hence, the new model is able to provide complex events compliant with uncertainty and usable for various applications, in particular, activity recognition.

To illustrate our contribution, we describe a short use case scenario. Let us consider a user living in a smart home equipped with multiple sensors. A smart application aims to recognize the current activity of the user and requires the user's location. However, the user is detected in multiple rooms due to his/her movement or unreliable sensors. With a standard CEP approach, only the most probable value is kept. However, with this solution, the selected value may still not match the real one, leading to a misunderstanding of the activity. With our approach, events are analyzed and one fuzzy semantic complex event is provided with trust weights for each possible interpretation. The activity recognition process can then use these information to provide a more reliable result.

This paper introduces a new model, its implementation and a short state of the art.

[1] LIPAH, Université de Tunis El Manar, 1068, Tunis, Tunisia, email: firstname.surname@telecom-sudparis.eu
[2] SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay, Evry, France, email: firstname.surname@telecom-sudparis.eu
[3] Université de Carthage, 1054, Tunis, Tunisia, email: Khedija.Arour@issatm.rnu.tn
[4] DAVID, University of Versailles Saint-Quentin-en-Yvelines, Versailles, France, email: beatrice.finance@uvsq.fr

2 FSCEP model

As stated before, to overcome uncertainty challenges for event processing in smart homes, we propose a Fuzzy Semantic Complex Event Processing (FSCEP) model that combines traditional CEP, semantic features through a domain ontology, and the fuzzy logic paradigm.

2.1 Architecture

As expressed in figure 1, FSCEP is a three steps process.

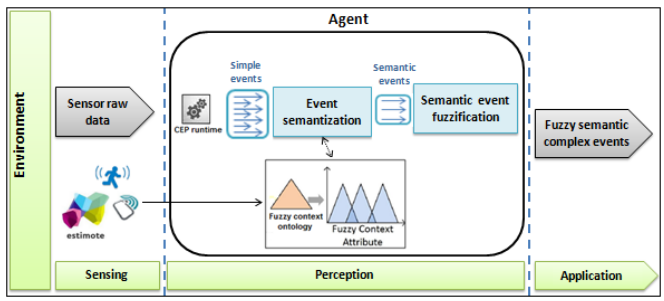


Figure 1. FSCEP approach for environmental perception

Firstly, the **sensing** phase deals with acquiring context data from physical sensors as simple events. Sensors are various and provide heterogeneous data, such as cameras or motion sensors. The data can be pushed by the sensors or pulled by the system. Both approaches are tackled by FSCEP: pushed data are treated when received while pulled data are asked periodically.

Secondly, the **perception** phase consists in analyzing simple events in order to transform them into higher level fuzzy semantic complex event (FSCE). Perception is the heart of our work and it has three steps that are detailed in the next section.

Thirdly, once the FSCE is computed, it is provided to an **application**. This application will then take advantage of the semantic and fuzziness of the provided event. In our global work, we aim to provide a situation recognition system based on a fuzzy ontology: the FSCEP would enrich the ontology on whom a recognition algorithm would be applied.

2.2 Perceiving

Our main contribution lies in the perception layer. The perception is put into action through multiple agents. Each agent is able to handle the whole perception process and each agent focuses on one type of context data. For instance, one agent handles user's location while

another focuses on the temperature. Once processed, agents send generated FSCEs to the application. The perception proceeds by a three-steps process: event processing, semantization and fuzzification.

Event processing - Simple events are gathered, filtered and batched using a standard CEP, such as ESPER. The CEP is provided with a user query, each agent has a different one. This query defines the duration of acquisition (time window), what type of events to gather (example: location) and conditions to filter these events. The output of this step is a batch of filtered events.

Semantization - At this step, events only carries sensors data. This phase aims to enrich events by using a domain ontology to make them more compliant with the upcoming phases. Semantized events are for example enhanced with sensors' confidences values or devices' positions. In this process, events are also modeled as RDF graphs, that is to say a set of triples, such as: $(camera1, hasTrust, "0.8")$. This allows a better integration and interoperability of events.

Fuzzification - A batch of semantic events is analyzed as a whole to compute a single FSCE. We define a dedicated membership function MF that calculates trust weights for each possible values from a batch of semantic events. The resulting FSCE carries those weights, for example: $FSCE_{(presence)} = \{[livingroom; 0.75], [office; 0.25]\}$. In our work, we have defined MF as a weighted mean: it is the average occurrence of a possible value (ex: living room or office) weighted by the confidence values associated to each events. It is possible to substitute it with other functions.

In the end, the FSCE is provided to the application. It can easily be integrated in an ontology, thanks to its RDF format, and can be processed in a fuzzy inference engine.

3 Implementation and evaluation

We evaluated FSCEP through a simulation. Our prototype is implemented in Java and is agent oriented (Jade[5]). We use a famous CEP, ESPER [4] combined with the Jena[6] framework to handle ontologies. The background knowledge is carried by a context ontology inspired from [6]. The fuzzy logic is integrated through annotations. The simulation was set up in Freedomotic[7], a development framework used for managing smart spaces.

We simulated the case of activity recognition in a smart home. The application aims to identify what activity the user is doing and requires the user's location to do so. User position is acquired through multiple sensors, however the interpretation is uncertain. Our approach is able to overcome uncertainty and to provide fuzzy events that will later enable a better and more reliable activity recognition. We plan to evaluate our approach in a physical and real context.

4 State of the art

Although researches on CEP mainly focus on performance [9, 2], uncertainty was tackled by several approaches. Most approaches rely on probabilities to overcome uncertainty, in particular accuracy. To the best of our knowledge, there is no work that uses fuzzy logic. Compared to probabilistic approaches, fuzzy logic is more suitable to tackle imprecision and contradiction: they deal with different uncertainty dimensions. Of course, fuzzy logic and probability do not have the exclusivity and other techniques are used. Let us have a short review of the literature. First, Wasserkrug et al. [7] addressed two types of uncertainty in CEP: the imprecision of events and the uncertain relations between events. The authors defined a probability space and relied on dynamically generated Bayesian Networks to compute the relevant probabilities. Artikis et al. [1] identified five essential uncertainty dimensions in CEP, such as incomplete data or imprecision. They proposed multiple ideas to solve these challenges, but none of them uses fuzzy logic. More recently, Cugola et al. [3] proposed CEP2U (Complex Event Processing under Uncertainty). They tackle accuracy over events and rules by using theory of probability and Bayesian Networks. Finally, Lee et al. [5] aimed to tackle the problem of dynamic changes in the environment domain and/or expert mistakes. This can be seen as a problem of freshness. They have proposed a tool known as Sequence Clustering-based Automated Rule Generation (SCARG) that generates rules by analyzing decision-making history.

These approaches tackle either fewer or different uncertainty dimensions than our FSCEP. Some of them are complementary and compatible with our approach.

5 Conclusion

In this work, we propose a novel combination between CEP, ontology and fuzzy logic to tackle four essential uncertainty dimensions in smart home applications. Our first experiments shows encouraging results but we aim to deepen our tests and use-cases to validate our approach. Finally, several improvements are scheduled as future works. This includes using machine learning to set up sensors confidence values and making our approach compatible with open and unknown environments.

[5] http://jade.tilab.com/
[6] https://jena.apache.org/
[7] http://freedomotic.com/

REFERENCES

[1] Alexander Artikis, Opher Etzion, Zohar Feldman, and Fabiana Fournier, 'Event processing under uncertainty', in *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, pp. 32–43. ACM, (2012).

[2] Lars Brenna, Alan Demers, Johannes Gehrke, Mingsheng Hong, Joel Ossher, Biswanath Panda, Mirek Riedewald, Mohit Thatte, and Walker White, 'Cayuga: a high-performance event processing engine', in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pp. 1100–1102. ACM, (2007).

[3] Gianpaolo Cugola, Alessandro Margara, Matteo Matteucci, and Giordano Tamburrelli, 'Introducing uncertainty in complex event processing: model, implementation, and validation', *Computing*, **97**(2), 103–144, (2015).

[4] Espertech event stream intelligence. Espercomplex event processing, 2010.

[5] O-Joun Lee and Jai E Jung, 'Sequence clustering-based automated rule generation for adaptive complex event processing', *Future Generation Computer Systems*, (2016).

[6] Natalia Díaz Rodríguez, Manuel P Cuéllar, Johan Lilius, and Miguel Delgado Calvo-Flores, 'A fuzzy ontology for semantic modelling and recognition of human behaviour', *Knowledge-Based Systems*, **66**, 46–60, (2014).

[7] Segev Wasserkrug, Avigdor Gal, and Opher Etzion, 'A model for reasoning with uncertain rules in event composition systems', *arXiv preprint arXiv:1207.1427*, (2012).

[8] Juan Ye, Simon Dobson, and Susan McKeever, 'Situation identification techniques in pervasive computing: A review', *Pervasive and mobile computing*, **8**(1), 36–66, (2012).

[9] Haopeng Zhang, Yanlei Diao, and Neil Immerman, 'On complexity and optimization of expensive queries in complex event processing', in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 217–228. ACM, (2014).

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

doi:10.3233/978-1-61499-672-9-1680

Multi-Context Systems in Time

Stefania Costantini and **Andrea Formisano** [1]

Abstract. In this paper we consider how to enhance flexibility and generality in Multi-Context Systems (MCS) by considering that contexts can evolve over time, that bridge-rule application can be proactive (according to a context's specific choice), and not instantaneous but requiring an execution mechanism. We introduce bridge-rule patterns to make bridge-rules parametric w.r.t. the involved contexts.

1 BRIDGE RULES AND MCS

Multi-Context Systems (MCSs) have been proposed in Artificial Intelligence and Knowledge Representation to model information exchange among heterogeneous sources. MCSs are defined so as to drop the assumption of making such sources in some sense homogeneous: rather, the approach deals explicitly with their different representation languages and semantics. Heterogeneous "contexts" (also called "sources", or "modules") interact through special inter-context *bridge rules*. The reason why MCSs are particularly interesting is that they aim at modeling in a formal way real applications requiring access to sources distributed, for instance, on the web. In view of such practical applications it is important to notice that, being logic-based, contexts may encompass logical agents, to which MCSs have in fact already been extended (cf. [2, 3]).

We refer the reader to [1, 5], and the references therein, for the formal definition of basic notions and properties concerning MCSs and *managed MCSs* (for short, mMCSs), such as context, bridge rule, belief state, management function, equilibrium, etc.

2 PROPOSED EXTENSIONS

2.1 Update Operators and Timed Equilibria

Let us assume a discrete, linear model of time. States $t_1, t_2, \ldots$ can be seen as time instants (or 'time points') in abstract terms. Moreover, we assume that each context is subject, at each time point, to a (possibly empty) finite update. Thus, for mMCS $M = (C_1, \ldots, C_n)$ let $\Pi_T = \langle \Pi_T^1, \ldots \Pi_T^n \rangle$ be a tuple composed of the finite updates performed to each context at time T, where, for $1 \leq i \leq n$, Π_T^i is the update to context C_i (possibly including the set *Ops* of sensor inputs of [1]). Let $\Pi = \Pi_1, \Pi_2, \ldots$ be a sequence of such updates performed at time instants $t_1, t_2, \ldots$ Let us assume that each context copes with updates in its own particular way, so let $\mathcal{U} = \{\mathcal{U}_1, \ldots, \mathcal{U}_n\}$ be the tuple composed of the *update operators* $\mathcal{U}_i$s that modules C_is employ for incorporating the new information. Let, morever, the *update base* $uops_i$ be a set of update operations which are admitted on context C_i. Then we have: $\mathcal{U}_i : 2^{uops_i} \times KB_i \rightarrow 2^{KB_i} \setminus \{\emptyset\}$, where KB_i is the set of knowledge bases pertaining to C_i. Consequently, we allow contexts' knowledge bases and belief states to evolve in time.

Definition 1 *Given context $C_i = (c_i; L_i; kb_i; br_i; OP_i; mng_i)$ as originally defined in mMCS, we define the corresponding* timed *context w.r.t. belief state S as follows:*
$C_i^0 = (c_i; L_i; kb_i^0; br_i; OP_i; mng_i; uops_i; \mathcal{U}_i)$
$C_i^{T+1} = (c_i; L_i; kb_i^{T+1}; br_i; OP_i; mng_i; uops_i; \mathcal{U}_i)$,
where $kb_i^0 = kb_i$ and $kb_i^{T+1} = mng_i(app(S^T), \mathcal{U}_i(\Pi_T^i, kb_i^T))$, with Π_T^i being the update performed on C_i^T, and $app(\cdot)$ determines the set of bridge rules that are applicable in a belief state.
Timed context C_i^T will also be referred to as "context C_i at time T".

Definition 2 *Let $M = (C_1^T, \ldots, C_n^T)$ be an mMCS at time T. A* timed *belief state at time T is a tuple $S^T = (S_1^T, \ldots, S_n^T)$ where each S_i^T is a possible set of consequences of C_i^T's knowledge base.*

The timed belief state S^0 will possibly be an equilibrium, according to original mMCS definition. Later on, however, transition from a timed belief state to the next one, and consequently the definition of an equilibrium, is determined both by the update operators and by the application of bridge rules. Therefore:

Definition 3 *A timed belief state of mMCS M at time $T+1$ is a timed equilibrium iff, for $1 \leq i \leq n$ it holds that $S_i^{T+1} \in ACC_i(kb_i^{T+1})$, where $ACC_i(\cdot)$ is a function which computes the semantics of C_i.*

The meaning is that a timed equilibrium is now a data state which encompasses bridge rules applicability on the updated contexts' knowledge bases. As seen in Def. 1, bridge-rule applicability is checked on belief state S^T, but bridge rules are applied (and their results incorporated by the management function) on the knowledge base resulting from the update. The enhancement w.r.t. [1] is that the management function now resumes its original role concerning bridge rules, while the update operator copes with updates. So, we relax the limitation that each rule involving an update should be considered to be a bridge rule, and that updates should consist of (the combination and elaboration of) simple atoms occurring in bridge bodies. Our approach in fact allows update operators to consider and incorporate any piece of knowledge. Moreover, we make time explicit thus showing the timed evolution of contexts and equilibria.

2.2 Bridge Rule Grounding and Activation

In the original definition of mMCSs, bridge rules are by definition ground and their application is reactive. However, according to a context's own logic, other patterns of application might in principle be defined. In particular, we admit non-ground bridge rules that might be grounded over all terms that can be built over the signature of every context's underlying logic. This because mMCSs admit

[1] DISIM – Università di L'Aquila, Italy, email: stefania.costantini@univaq.it and DMI–Università di Perugia, Italy, email: formis@dmi.unipg.it. Research partially supporded by GNCS and FCRPG.2016.0105.021 project.

"value invention", i.e., via the results of bridge-rules application, beliefs (and their arguments) are propagated among contexts; so, via a bridge rule a context may obtain a result involving constants and terms previously not occurring therein.

Definition 4 *Let r be a non-ground bridge rule occurring in context C_i of a given mMCS M with (timed) belief state S^T. A ground instance ρ of r w.r.t. S^T is obtained by substituting every variable occurring in r via ground terms occurring in S^T.*

As concerns proactive activation, let, for each context C_i, $H(T,i) = \{h|\text{there exists rule } r \in br_i \text{ with head } h \text{ at time } T\}$, and let $tr_i^T : KB_i \rightarrow 2^{H(T,i)}$ be a *timed triggering function*, specifying which rules are triggered at each time T, by performing some reasoning over the present knowledge base in KB_i of C_i. Then, a bridge rule r of context C_i is *triggered at time* T iff $r \in tr_i^T(kb_i^T)$. Note that, a (ground instance of) a bridge rule can be triggered at a time T', but can become applicable at some later time T. Thus, any bridge rule which has been triggered in actual terms remains in predicate for applicability, which occurs whenever its body should be entailed by some future data state. The definition of timed equilibrium remains unchanged, save for modified bridge-rule applicability. However, the added expressivity is remarkable, as with our solution a context is not just the passive recipient of new information, but rather can reason about which bridge rules to potentially apply at each stage. Moreover, in practical applications, the grounding of literals in bridge rules can be computed at run-time, when the rule is actually applied.

2.3 Bridge-Rules Patterns

In [4] we have proposed, for logical agents, *bridge-rule patterns* which are enhanced bridge rules of the form

$$s \leftarrow (\mathcal{C}_1{:}p_1), \ldots, (\mathcal{C}_j{:}p_j), not\,(\mathcal{C}_{j+1}{:}p_{j+1}), \ldots, not\,(\mathcal{C}_m{:}p_m).$$

where each $\mathcal{C}_d$ can be either a constant (i.e., a context name, as in usual bridge rules) or a *context designator* (which is a term built up from fresh function symbol and constants. Intuitively, a context designator indicates a specific kind of context, and can be substituted by a context name. New bridge rules can thus be obtained by replacing, in a bridge-rule pattern, context designators via actual context names. So, contexts will now evolve also in the sense that they may increase their set of bridge rules by exploiting bridge-rule patterns. Note that the context names to substitute to a context designator are established by suitable reasoning performed according to context's knowledge base. All other previously-introduced notions (equilibria, bridge-rule triggering and applicability, etc.) remain unchanged. Notice that bridge-rule patterns instantiation corresponds to specializing rules with respect to the context which is deemed more suitable for acquiring some specific information at a certain stage of a context's operation. This evaluation is performed via specific predicates (cf. [5]) so as to take several factors into account, among which, for instance, trust and preferences. Also, this enhancement goes in the direction of *dynamic* mMCS, where contexts can either join or leave the system during its operation, while rule applicability may depend upon the presence in the system of suitable contexts.

3 COMPLEXITY ISSUES

In general, the property that we may wish to check is whether a specific belief of our interest will eventually occur at some stage in one (or all) timed equilibria of a given mMCS. The formal definition is the following.

Definition 5 *The problem $Q^{\exists}$ (respectively $Q^{\forall}$), consists in deciding whether, for a given mMCS M under a sequence $\Pi = \Pi_1, \Pi_2, \ldots$ of updates performed at time instants $t_1, t_2, \ldots, t$, and for a context C_i of M and a belief b_i for C_i, it holds that $b_i \in S_i^{t'}$ for some (respectively for all) timed equilibria $S^{t'}$ at time $t' \leq t$.*

The system's *context complexity* (see [6]) depends upon the logics of the contexts composing M. Then, in general, the problems $Q^{\exists}$ and $Q^{\forall}$ are undecidable for infinite update sequences, because contexts' logics can in general simulate a Turing Machine and such problems reduce to the halting problem. Complexity results can however be obtained under some restrictions. For instance, if we assume that all contexts C_i's knowledge bases and belief states are finite at any stage, all update operators $\mathcal{U}_i$, management functions and triggering functions are computable in polynomial time, and that the set of bridge-rule patterns is empty and all bridge rules are ground, it is easy to show that for finite update sequences the context complexity determines the complexity of $Q^{\exists}$ and, complementarily, that of $Q^{\forall}$.

Reconsidering bridge rules patterns, and assuming that substitutions for each context designator can be completed in polynomial time, given a set of bridge-rule patterns of size $\hat{c}$, the size of the set of its valid instances is in general single exponential in $\hat{c}$. The same holds in principle for the grounding of (non-ground) bridge rules.

4 CONCLUDING REMARKS

In this paper (see also [5]) we have discussed and extended MCSs, which are a general and powerful framework for modeling systems composed by several heterogeneous and possibly distributed sources. We have in particular considered that such systems may evolve in time in consequence of updates of several kinds, and that bridge rules get actually instantiated at run-time, and that their actual execution in a distributed setting cannot be instantaneous, but rather may involve some delay. We introduced bridge-rule patterns to make bridge rules parametrical w.r.t. the queried contexts.

We intend to employ the proposed features in the implementation, that will start in the near future, of a smart Cyber-Physical System for e-health, which will be developed in cooperation with medical oncology doctors, and will be tested and applied in the monitoring of oncology patients with co-morbidities. Participation to this project will allow us to perform also practical experiments to assess the performance of this kind of systems, and to identify possible limitations and/or further aspects that can be subject to improvements.

REFERENCES

[1] G. Brewka, S. Ellmauthaler, and J. Pührer, 'Multi-context systems for reactive reasoning in dynamic environments', in *Proc. of ECAI 2014*, ed., Torsten Schaub. IJCAI/AAAI, (2014).

[2] S. Costantini, 'Knowledge acquisition via non-monotonic reasoning in distributed heterogeneous environments', in *Proc. of LPNMR-13*, eds., M.Truszczyński, G.Ianni, and F.Calimeri, volume 9345 of *LNCS*. Springer, (2015).

[3] S. Costantini and G. De Gasperis, 'Exchanging data and ontological definitions in multi-agent-contexts systems', in *RuleML Challenge track*, eds., A.Paschke, P.Fodor, A.Giurca, and T.Kliegr, CEUR, (2015).

[4] S. Costantini and A. Formisano, 'Augmenting agent computational environments with quantitative reasoning modules and customizable bridge rules', in *Proc. of AAMAS Workshop, EMAS 2016, Proceedings*, eds., M.Baldoni J.P.Müller, I.Nunes, and R.Zalila-Wenkstern, (2016).

[5] S. Costantini and A. Formisano. Multi-context systems in time, 2016. Full version. `www.dmi.unipg.it/formis/MCSIT.pdf`.

[6] T. Eiter, M. Fink, P. Schüller, and A. Weinzierl, 'Finding explanations of inconsistency in multi-context systems', in *Proc. of KR 2010*, eds., F. Lin, U. Sattler, and M. Truszczynski. AAAI Press, (2010).

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

doi:10.3233/978-1-61499-672-9-1682

Speech Emotion Recognition Using Voiced Segment Selection Algorithm

Yu Gu[1] and **Eric Postma** [2] and **Hai-Xiang Lin**[3] and **Jaap van den Herik** [4]

Abstract. Speech emotion recognition (SER) poses one of the major challenges in human-machine interaction. We propose a new algorithm, the Voiced Segment Selection (VSS) algorithm, which can produce an accurate segmentation of speech signals. The VSS algorithm deals with the voiced signal segment as the texture image processing feature which is different from the traditional method. It uses the Log-Gabor filters to extract the voiced and unvoiced features from spectrogram to make the classification. The finding shows that the VSS method is a more accurate algorithm for voiced segment detection. Therefore, it has potential to improve performance of emotion recognition from speech.

1 Introduction

A speech always consists of voiced and unvoiced parts. When vocal chords vibrate to pronounce vowels, voiced segments are generated. In contrast to unvoiced segments, voiced segments show periodic and prosodic signals. Unvoiced segments show irregular signals, generated by the influence of narrow vocal tract. Emotion is an important component of the information contained in a speech. Emotional information in speech is represented in a variety of prosodic types and is also mainly contained in the voiced parts. That is why researchers always focus on the voiced parts of a speech in emotion recognition.

The outline of the paper is as follows. Section 2 describes the methodology of VSS method in detail. In Section 3, the set-up of the comparative evaluation of the VSS method is described. Experimental results and discussions are presented in Section 4. Finally, Section 5 provides concluding remarks.

2 Formulation of VSS Method

There are two key components in the VSS method which are the spectrogram and the log-Gabor filter.

The VSS method deals with the voiced activity detection as a classification issue. In contrast to existing machine learning algorithm for VAD which normally used the acoustic features or statistical features for the classification, we propose a novel kind of feature which is extracted by log-Gabor filter from the spectrogarm for this issue. We use two-dimensional Gabor filters which are locally corresponding to the orientations of energy bands in the spectrogram. Therefore, by convolving the spectrogram with a Gabor filter of a given spatial frequency and orientation, the convolved spectrogram represents spectro-temporal patterns with the associated spatial frequency (width) and orientation, respectively. We use the log-Gabor filters with combination of scale and orientation for convolving whole spectrogram to obtain the texture information. Because all the texture patterns are generated by voiced parts in the spectrogram, these convolution results from the filters are sensitive to the voiced segment for speech signal. Each combination of different scale and orientation will result in one filter-image. The energy of each filter-iamge will be averaged by the time sequence and be stored as a vector data. And finally, all the vectors which calculated by each filters will be assemble as a matrix. The Support vector machine will be used to classify the voiced and unvoiced segment by using the convolution feature matrix from spectrogram.

3 Experiment of the VSS method

This section describes the implementation which is used in the experiments and evaluation details.

The evaluation of the VSS method was performed on two corpora: the Mandarin Affective Speech (MAS) corpus (MAS, 2007) and the Berlin Database of Emotional Speech (Emo-DB, 2007). More details about the corpus can be found via the Linguistic Data Consortium website[5].

The experimental procedure consisted of three steps: (i) spectrogram generation, (ii) spectrogram convolution using Log-Gabor filters, (iii) voiced and unvoiced segment classification. In the following, the details of each of these steps will be outlined.

Table 1. The value of parameters for Log-Gabor filter

name of parameter	value of parameter
spectrogram time resolution t	0
scale N_s and orientation N_o	12 12
segmented Patch	8
minwavelength	3
mult	1.35
sigmaOnf	0.8

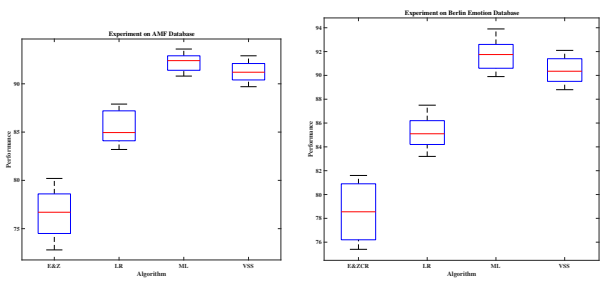
(i) The speech signal was visualized in spectrogram. Each auditory signal (utterance) was transformed into a spectrogram using Matlab's spectral analysis function. (ii) Each spectrogram is convolved by a Log-Gabor filter with a number different parameter values of scale

[1] Tilburg center of Communication and cognition, Tilburg University, Tilburg, Netherlands, email: y.gu_1@uvt.nl
[2] Tilburg center of Communication and cognition, Tilburg University, Tilburg, Netherlands
[3] Institute of Applied Mathematics, Delft University of Technology, Delft, Netherlands
[4] Leiden Institute of Advanced Computer Science, Leiden University, Leiden, Netherlands

[5] http://catalog.ldc.upenn.edu/LDC2007S09

and orientation. N_s and N_o denote the scale and the orientation respectively. Therefore, if the number of scales and the orientations are equal to N_s and N_0 respectively, then, $N_s * N_o$ bank filter images are generated. The parameters of log-Gabor filters as shown in Table 1 will be applied to make the convolution to the spectrogram. The convolution values within each image is averaged by the time sequence and store as a vector. Thus, each vector will have $1 * 512$ value. (iii) Step (ii) will produce $N_s * N_o$ convolution image, and the total number of the speech record is N_t. Therefore $N_s * N_o * N_t$ vectors which we could achieve after the whole convolution. Combining the energy value vectors into matrices features which we can use for training an SVM classifier.

The performance evaluation consists of two parts: a comparison in accuracy of voiced part detection between the VSS method exiting voiced detection methods and the performance in emotion recognition by the VSS method. First, we compare the VSS method with another prevalent VAD method in the voiced segment detection. Three major algorithms were duplicated: the a) The Energy and ZRC method, b) the statistical likelihood ratio method [1] and c) the Deep belief network method [2].



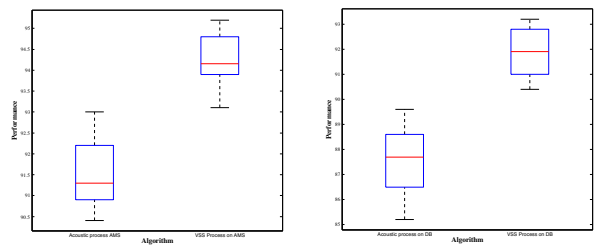
(a) AMS database (b) Berlin database

Figure 1. Comparison of the voiced part accuracy performances obtained on both database.

Figure 1 show the comparison of the voice accuracy between the three major methods and our VSS method by using the AMS and DB database respectively. We can clearly observe that our method was significantly better than the traditional ZCR and LR method. There are three possible explanations are as follows. Firstly, the spectrogram contains various kinds of most occurring acoustic phenomena, such as hamonicity, formants, vertical edges which are all the crucial characters for indicating the voice parts. And the formant and vertical edges are exactly the start and end point for the voice signal. Therefore, it can detect all acoustic phenomena which in other words to recognize the voice speech in a robust manner. Secondly, traditional methods have limitation due to manually set thresholds, the value of threshold was not always optimized. Compare to that, the SVM has a strong ability for the high dimensional feature learning that for the voice and unvoiced discrimination which can be less influence than manually and more accurate. Last reason for the higher accuracy is due to that, the log-Gabor filters has strong ability to distinguish the phonetic phenomena from noise background in spectrogram. Even through the noise in speech which can bring a huge number of corresponded peaks to the spectrogram. The reason which can be explained this is because the output of Log-Gabor filters is achieved from integrating the entire 2-D spectrogram information, which make the 2-D filterbank more robust ability to the noise. Moreover, the DBN method slightly outperforms the VSS method. However, for the deep learning it requires huge computation consumption. Therefore, we conclude that the VSS method is a useful and practical method to improve the voice detection accuracy than the traditional method.

4 Experiment of Speech Emotion Recognition

In the speech emotion recognition experiment, the acoustic features were extracted from the voiced segments based on the VSS method. The performance of speech emotion recognition was also compared with and without VSS procedure before the feature extraction. To avoid over-fitting due to the PCA and the SVM parameter optimization, the evaluation was performed using a nested 10×10-fold cross validation,.



(a) AMS database (b) Berlin database

Figure 2. Comparison of the recognition performances obtained from both database.

The classification was applied in both corpus of MAS and EmoDB. Figure 2 visualizes the classification performances with and without VSS before feature extraction. Both of the features were base on the state-on-art baseline. The comparative evaluation of employing VSS method against without VSS method. It demonstrates that the performance of speech emotion recognition with VSS method led to a non-overlapping improvement. The results showed that VSS method for voiced detection is more effective in extracting features, because most of the acoustic features are related to the voiced parts.

5 Conclusions

We proposed a novel VSS algorithm which uses the log-Gabor filter to extract the features from the spectrogram for the voiced segment classification. Experiment results showed that the VSS algorithm can be a useful and practical method for the voiced activity detection. Furthermore, the performance in speech emotion recognition shows that the classification rate is improved for both Chinese and German speech. It convinced us the VSS is an useful compliment for the speech emotion recognition.

REFERENCES

[1] Youngjoo Suh and Hoirin Kim, 'Multiple acoustic model-based discriminative likelihood ratio weighting for voice activity detection.', *IEEE Signal Process. Lett.*, **19**(8), 507–510, (2012).

[2] Xiao-Lei Zhang and Ji Wu, 'Deep belief networks based voice activity detection', *Audio, Speech, and Language Processing, IEEE Transactions on*, **21**(4), 697–710, (2013).

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

doi:10.3233/978-1-61499-672-9-1684

Scalable Exact MAP Inference in Graphical Models

Radu Marinescu and **Akihiro Kishimoto** and **Adi Botea**[1]

Abstract. This paper presents parallel dovetailing in a distributed-memory environment for exact MAP inference in graphical models. Parallel dovetailing is a simple procedure which performs multiple searches in parallel with different parameter configurations. We evaluate empirically the performance of parallel dovetailing with three state-of-the-art AND/OR search algorithms in solving various MAP inference benchmarks. Our results clearly show that parallel dovetailing is effective, yielding considerable speedups and improving the solving abilities of these state-of-the-art baseline methods.

1 INTRODUCTION

Graphical models provide a powerful framework for reasoning about conditional dependency structures over many variables. The Maximum a Posteriori (MAP) task, which asks for the mode of the joint probability distribution defined by the graphical model, arises in many applications and is often solved efficiently with search methods. Current state-of-the-art exact algorithms for MAP inference traverse either depth-first or best-first an AND/OR search space while being guided by admissible heuristics based on the mini-bucket partitioning [1, 7, 4]. Parallelizing these algorithms is difficult, especially in distributed-memory environments, due to parallel overheads.

In this paper, we apply the so-called *parallel dovetailing* [9] to parallelize *any* AND/OR search-based exact MAP inference algorithm in a distributed-memory environment. Parallel dovetailing is a simple scheme that executes many search algorithms with different configurations in parallel. In order to exploit search diversity, our scheme generates many different problem reformulations as inputs. Then, each search algorithm attempts to solve an assigned, reformulated problem individually without sharing any information with the others. If any of the search algorithms returns a solution, then that solution is the optimal one and the other algorithms can safely be terminated. We evaluate empirically the performance of parallel dovetailing for depth-first and best-first AND/OR search on solving various MAP inference benchmarks. Our experimental results show that parallel dovetailing achieves significant improvement to all of these algorithms, often yielding super-linear speedups especially in solving problem instances with a large branching factor.

2 BACKGROUND

A *graphical model* is a tuple $\mathcal{M} = \langle \mathbf{X}, \mathbf{D}, \mathbf{F} \rangle$, where $\mathbf{X} = \{X_i : i \in V\}$ is a set of variables indexed by set V and $\mathbf{D} = \{D_i : i \in V\}$ is the set of their finite domains of values. $\mathbf{F} = \{\psi_\alpha : \alpha \in F\}$ is a collection of discrete positive real-valued local functions defined on subsets of variables, where $F \subseteq 2^V$, and $\alpha \subseteq V$. The graphical model $\mathcal{M}$ defines a factorized probability distribution on $\mathbf{X}$, $p(\mathbf{x}) = \frac{1}{Z} \prod_{\alpha \in F} \psi_\alpha(\mathbf{x}_\alpha)$, where $Z = \sum_{\mathbf{x}} \prod_{\alpha \in F} \psi_\alpha(\mathbf{x}_\alpha)$ is called the *partition function* and normalizes the probability. The MAP task looks for a complete assignment to the variables that has the highest probability (i.e., a mode of the joint probability), namely to find $\mathbf{x}^* = \text{argmax}_{\mathbf{x}}\, p(\mathbf{x}) = \text{argmax}_{\mathbf{x}} \prod_{\alpha \in F} \psi_\alpha(\mathbf{x}_\alpha)$.

Significant recent improvements in search for MAP inference have been achieved by using AND/OR search spaces, which often capture problem structure far better than standard OR search methods [1]. The AND/OR search space is defined relative to a *pseudo tree* of the primal graph, which captures problem decomposition. The current state-of-the-art for exact MAP inference is based either on depth-first or best-first exploration of the AND/OR search space, guided by mini-bucket heuristics enhanced with a variational cost-shifting mechanism [7, 8, 3]. Algorithms like AOBB and AOBF have shown robust performance on various benchmarks. More recently, a recursive best-first AND/OR search approach (RBFAOO) that can operate within restricted space has emerged as one of the best-performing exact search algorithms [4]. A parallel version for shared-memory environments called SPRBFAOO has also been proposed in [5].

3 DOVETAILING AND/OR SEARCH

We introduce a simple *dovetailing* strategy to enable distributed MAP inference. Given a graphical model $\mathcal{M}$ and m processing cores, the basic idea is to run in parallel m instances of a MAP algorithm a, one for each core, such that each instance solves the same model $\mathcal{M}$ but uses a different parameter configuration θ_i, where $1 \leq i \leq m$. The communication is limited to broadcasting messages signaling that $\mathcal{M}$ has been solved by a core and the other cores should stop running their assigned algorithm instance. Clearly, many of the properties of dovetailing relate to the properties of its underlying MAP inference algorithm a. For example, if algorithm a guarantees optimality of the solution, then dovetailing is optimal as well.

Extending dovetailing to AND/OR search spaces can be done by using *any* of the exact AND/OR search algorithms introduced in recent years such as AOBB, RBFAOO and SPRBFAOO, respectively. We use the pseudo tree as the parameter configuration θ_i, and therefore generate m distinct pseudo trees using a randomized *min-fill* heuristic [6]. This is because the performance of the AND/OR search algorithms is strongly influenced by the quality of the guiding pseudo tree that determines variable ordering [7]. Furthermore, this choice applies with no change to multiple AND/OR search methods, as opposed to having to find a different set of parameters for every baseline algorithm considered.

4 EXPERIMENTAL RESULTS

Our benchmark problems[2] include three sets of instances from genetic linkage analysis (denoted `pedigree` with 22 instances) [2],

[1] IBM Research – Ireland, email: radu.marinescu@ie.ibm.com

[2] *http://graphmod.ics.uci.edu*

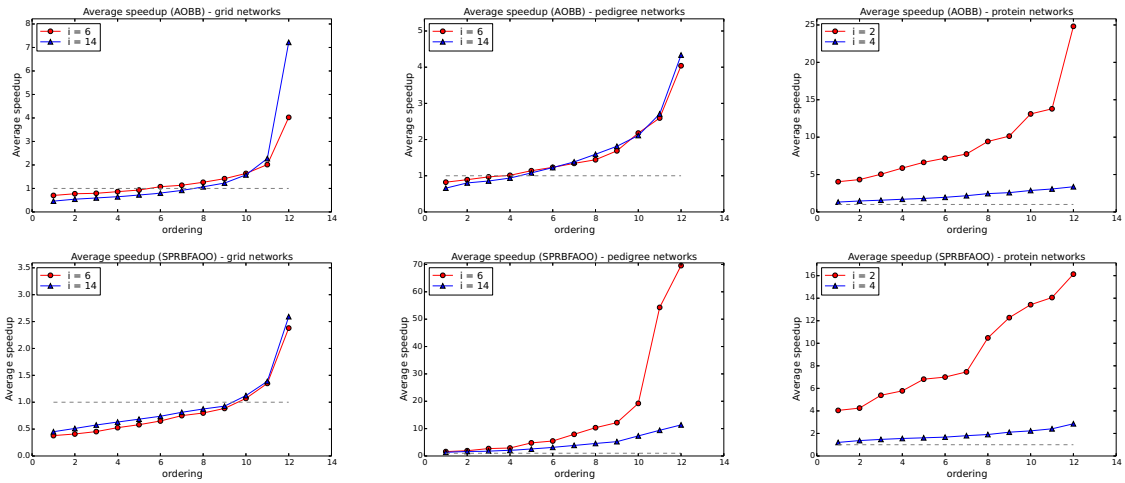


Figure 1. Average speedups. Top: AOBB; bottom: SPRBFAOO. Left: `grid` domain; middle column: `pedigree` domain; right: `protein` domain.

grid networks (denoted `grid` with 32 instances) and protein side-chain interaction networks (denoted `protein` with 240 instances) [10]. Following [5], all algorithms use the pre-compiled *mini-bucket heuristics* [7], with two distinct i-bound values for each domain: $i \in \{6, 14\}$ in `pedigree` and `grid`, and $i \in \{2, 4\}$ in `protein`.

For each problem instance, we consider 12 new variable orderings, in addition to the default minfill ordering reported in [5]. When applied to a serial algorithm, such as AOBB and RBFAOO, dovetailing is run on 12 CPU cores. For the parallel, shared-memory SPRBFAOO, dovetailing requires 144 cores in total. Each instance of SPRBFAOO runs on 12 cores, and there is one instance for each variable ordering considered in the dovetailing scheme.

Figure 1 plots the average speedups as a function of the variable orderings considered in the dovetailing solver (AOBB and SPRBFAOO). In case of a timeout, the cutoff time of 7,200 seconds is used as the running time. For a given combination of a domain, an algorithm and an i-bound, the speedup obtained with dovetailing is the largest value for the curve at hand. Indeed, as soon as the run with the best performing variable ordering completes, an optimal solution has been found and all other runs can be halted. We can see that our dovetailing approach leads to a significant variability of the performance across all variable orderings, which in turn translates into significant speedups (e.g., super-linear speedups on the `protein` domain).

Algorithm: **AOBB**					
instance	baseline	dovetail	instance	baseline	dovetail
Domain: `protein`					
	$i = 2$			$i = 4$	
pdb1aho	81.49	**0.49**	pdb1b0b	>7200	**2060.44**
pdb1c9x	>7200	**47.28**	pdb1b8e	4448.31	**676.90**
pdb1co6	>7200	**8.56**	pdb1jbe	2019.72	**121.08**
Algorithm: **RBFAOO**					
instance	baseline	dovetail	instance	baseline	dovetail
Domain: `protein`					
	$i = 2$			$i = 4$	
pdb1aac	46.61	**2.28**	pdb1bgc	>7200	**1984.16**
pdb1atg	700.69	**28.68**	pdb1b0b	>7200	**1668.23**
pdb1bkb	>7200	**351.09**	pdb1bkb	1239.16	**307.43**
Algorithm: **SPRBFAOO**					
instance	baseline	dovetail	instance	baseline	dovetail
Domain: `protein`					
	$i = 2$			$i = 4$	
pdb1atg	121.93	**6.32**	pdb1ail	490.02	**54.41**
pdb1bkb	2417.90	**59.76**	pdb1b0b	>7200	**258.72**
pdb1c9x	>7200	**7.76**	pdb1i5g	634.17	**132.74**

Table 1. CPU time (in seconds) for a subset of difficult instances from the `protein` domain. Timeout cases are written as ">7200." Best performance points are shown in bold.

Table 1 shows the CPU time in seconds obtained on a subset of difficult instances from the `protein` domain. The results show consistent savings achieved with the dovetailing strategy. The largest speedup exceeds a factor of 1443, on instance pedigree33 in the `pedigree` domain for $i = 6$.

5 CONCLUSION

Exact MAP inference in graphical models is an important problem that arises in many applications. However, despite recent progress in optimal solving approaches, such as search-based algorithms, many problem instances remain difficult. In this paper we presented a parallelization strategy based on dovetailing. The approach is simple, easy to implement, and it is not specific to a particular baseline solver. In dovetailing, several instances of a given baseline algorithm run in parallel each one with a different configuration of the parameter settings. Results on three popular benchmark domains demonstrate the benefits of this approach which often leads to substantial speedups.

REFERENCES

[1] R. Dechter and R. Mateescu, 'AND/OR search spaces for graphical models', *Artificial Intelligence*, **171**(2-3), 73–106, (2007).
[2] M. Fishelson and D. Geiger, 'Exact genetic linkage computations for general pedigrees', *Bioinformatics*, **18(1)**, 189–198, (2002).
[3] A. Ihler, N. Flerova, R. Dechter, and L. Otten, 'Join-graph based cost-shifting schemes', in *UAI*, pp. 397–406, (2012).
[4] A. Kishimoto and R. Marinescu, 'Recursive best-first AND/OR search for optimization in graphical models', in *UAI*, pp. 400–409, (2014).
[5] A. Kishimoto, R. Marinescu, and A. Botea, 'Parallel recursive best-first AND/OR search for exact map inference in graphical models', in *NIPS*, 928–936, Curran Associates, Inc., (2015).
[6] U. Kjærulff, 'Triangulation of graph-based algorithms giving small total space.', *Technical Report, University of Aalborg, Denmark*, (1990).
[7] R. Marinescu and R. Dechter, 'AND/OR branch-and-bound search for combinatorial optimization in graphical models', *Artificial Intelligence*, **173**(16-17), 1457–1491, (2009).
[8] R. Marinescu and R. Dechter, 'Memory intensive AND/OR search for combinatorial optimization in graphical models', *Artificial Intelligence*, **173**(16-17), 1492–1524, (2009).
[9] R. A. Valenzano, N. R. Sturtevant, J. Schaeffer, K. Buro, and A. Kishimoto, 'Simultaneously searching with multiple settings: An alternative to parameter tuning for suboptimal single-agent search algorithms', in *ICAPS*, pp. 177–184, (2010).
[10] C. Yanover, O. Schueler-Furman, and Y. Weiss, 'Minimizing and learning energy functions for side-chain prediction', *Journal of Computational Biology*, **15**(7), 899–911, (2008).

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

doi:10.3233/978-1-61499-672-9-1686

Hiding Actions in Concurrent Games

Vadim Malvone [1] and **Aniello Murano**[1] and **Loredana Sorrentino**[1]

Abstract. We study a class of determined two-player reachability games, played by *Player*$_0$ and *Player*$_1$ under imperfect information. Precisely, we consider the case in which *Player*$_0$ wins the game if *Player*$_1$ cannot prevent him from reaching a target state. We show that the problem of deciding such a game is EXPTIME-COMPLETE.

1 Introduction

In this paper we consider two-player reachability games ($2CRGI$), played by *Player*$_0$ and *Player*$_1$, where both players can have imperfect information about the actions taken by the other player. Some states of the game arena are set as *target*. We consider the game "optimistically" from the viewpoint of *Player*$_0$. Precisely, *Player*$_0$ *wins* the game if *Player*$_1$ does not have a strategy, while adhering to his observability, to prevent the former to reach a target state. Solving the game amounts to check whether *Player*$_0$ wins the game. Notably, $2CRGI$ result to be determined, *i.e.* in every state, either *Player*$_0$ has a wins the game or not [3]. We recall that two-player turn-based games are always determined, while concurrent games are generally not [1].

Technically, to solve a $2CRGI$ G we look for trees (representing *Player*$_1$ strategies) that, at each node and for every possible action taken by *Player*$_0$, collect the best possible counter-actions of *Player*$_1$, chosen under the visibility constraint. Such a tree is considered "blocking" whenever it contains only paths along which no target state is met. Then, we say that *Player*$_0$ wins the game if no such a tree exists. Otherwise, we say that *Player*$_0$ loses the game. We build in linear time an alternating tree automaton A collecting all such trees and thus reduce the problem of solving G to checking for the emptiness of A. As the latter can be checked in exponential time [8] and solving two-player turn-based games with imperfect information is EXPTIME-HARD [11], we finally get that the problem of solving $2CRGI$ is EXPTIME-COMPLETE.

2 Game definition

We model the game by means of a classic *concurrent game structure* [2] augmented with a set of *target states* and a set of *equivalence relations* over actions, one for each player. Precisely, for a *Player*$_i$ and two actions a and b, if $a \cong_i b$ we say that a and b are *indistinguishable* to *Player*$_i$. The formal definition of our game model follows.

Definition 2.1 *A* concurrent two-player reachability game with imperfect information *($2CRGI$) is a tuple* $G \triangleq <\mathrm{St}, s_I, \mathrm{P}, \mathrm{Ac}, \mathsf{tr}, \mathrm{W}, \cong>$ *where* St *is a finite non empty set of* states, $s_I \in \mathrm{St}$ *is a designated* initial state, $\mathrm{P} \triangleq \{$*Player*$_0$, *Player*$_1\}$ *is the set of players,* $\mathrm{Ac} \triangleq \mathrm{Ac}_0 \cup \mathrm{Ac}_1$ *is the set of actions. We assume that* $\mathrm{Ac}_0 \cap \mathrm{Ac}_1 = \emptyset$. W *is a set of target states,* $\mathsf{tr} : \mathrm{St} \times (\mathrm{Ac}_0 \times \mathrm{Ac}_1) \to \mathrm{St}$ *is a* transition function *mapping a tuple of one state and two actions to one state, and* $\cong \triangleq \cong_0 \cup \cong_1$ *is a set of equivalence relations on* Ac.

By $[\mathrm{Ac}_i]$ we denote the subset of *Player*$_i$ actions that are distinguishable to *Player*$_{1-i}$. If two actions are indistinguishable then also the states reached by using tr are so. A relation $\cong_i$ is called an *identity* if $a \cong_i b$ iff $a = b$. A $2CRGI$ has perfect information if $\cong$ contains only identity relations (so, we drop I from the acronym).

To give the semantics of $2CRGI$s, we now introduce some basic concepts such as track, strategy, and play.

A *track* is a finite sequence of states $\rho \in \mathrm{St}^*$ such that, for all $k < |\rho|$, there are two actions $a_0 \in \mathrm{Ac}_0$ and $a_1 \in \mathrm{Ac}_1$ such that $(\rho)_{k+1} = \mathsf{tr}((\rho)_k, a_0, a_1)$, where $(\rho)_k$ denotes the k-st element of ρ. For a track ρ, by $\rho_{\leq k}$ we denote the prefix track $(\rho)_0 ... (\rho)_k$. By $\mathrm{Trk} \subseteq \mathrm{St}^*$, we denote the set of tracks over St. For simplicity, we assume that Trk contains only tracks starting at the initial state s_I.

A *strategy* represents a scheme for a player containing a precise choice of actions along an interaction with the other player. It is given as a function over tracks. Formally, a *strategy* for *Player*$_i$ in a $2CRGI$ is a partial function $\sigma_i : \mathrm{Trk} \rightharpoonup \mathrm{Ac}_i$ that maps each track to an action. A strategy σ_i is *total* if it is defined on all tracks in Trk. A strategy is *uniform* if it adheres on the visibility of the players. Thus uniform strategies are based on observable actions. In the rest of the paper we only refer to uniform strategies.

The composition of strategies, one for each player in the game, induces a computation called *play*. More precisely, assume that *Player*$_0$ and *Player*$_1$ take strategies σ_0 and σ_1, respectively. Their composition *induces* a play ρ such that $(\rho)_0 = s_I$ and for each $k \geq 0$ we have that $(\rho)_{k+1} = \mathsf{tr}((\rho)_k, \sigma_0(\rho_{\leq k}), \sigma_1(\rho_{\leq k}))$, for all $k \in N$.

We have now all the ingredients to properly define how to interpret a game and thus to establish who wins it. The winning condition we set is based on the reachability condition viewed "optimistically" for *Player*$_0$. Precisely, *Player*$_0$ wins the game if *Player*$_1$ does not have a strategy to prevent him to reach a target state. Dually, we have that *Player*$_1$ wins the game (and thus *Player*$_0$ loses it) if, for each strategy σ_0 of *Player*$_0$, there exists a strategy σ_1 for *Player*$_1$, that can force the induced play to never reach a target state in W. The latter is the one we will use in the rest of the paper. Deciding the winner of a game by looking at the strength of the adversary is quite common in game theory applied to open system verification, largely investigated in the two-player turn-based setting. See [5–7, 9] for an argument.

It is worth noting that the winning condition we consider preserves the *determinacy* property [3], even under imperfect information [2]. The automata-based solution we propose in the next section will strongly

[1] Università degli Studi di Napoli Federico II, Italy.
Contact author: Aniello Murano, mail:{murano@na.infn.it}.

[2] For the sake of clarity, we recall that in the plain two-player reachability winning condition *Player*$_0$ wins the game if he has a strategy such that for all strategies of *Player*$_1$ the resulting induced play has at least one state in W. This definition does not guarantee the determinacy property in the concurrent setting. To be convinced considering the classic two-player concurrent *matching bit* game where no one of the player wins the game.

benefit from this fact.

To properly formalize the winning condition we use, we introduce a tree structure machinery that we call *blocking tree*. For the lack of space, we only give the main concepts about trees and refer to [12] for a proper definition. Let Υ be a set. An Υ-*tree* is a prefix closed subset $T \subseteq \Upsilon^*$. The elements of T are called *nodes* and the empty word ε is the *root* of T. Given a node $v = y \cdot x$, with $y \in \Upsilon^*$ and $x \in \Upsilon$, we define $prf(v)$ to be y and $last(v)$ to be x. For an alphabet Σ, a Σ-labeled Υ-tree is a pair $<T, V>$ where T is an Υ-tree and $V : T \to \Sigma$ maps each node of T to a symbol in Σ.

Definition 2.2 *Given a 2CRGI G, a* blocking tree *for Player$_i$ is a $\{\top, \bot\}$-labeled $([\mathrm{Ac}_0] \times [\mathrm{Ac}_1])$-tree $<T, V>$ with $T \subset ([\mathrm{Ac}_0] \times [\mathrm{Ac}_1])^*$ and V as follows: (i) $V(\varepsilon) = \top$; (ii) for all $v \in T$ let $\rho = (\rho)_0 ... (\rho)_{|v|-1}$ be a track from s_I such that for each $k < |v|$ we have that $(\rho)_k = tr((\rho)_{k-1}, last(v_{\leq k})(i), last(v_{\leq k})(1-i))$. If $last(v)(1-i) = \sigma(\rho)$ then $V(v) = \top$, otherwise $V(v) = \bot$.*

Directly from the definition of blocking tree, the definition of winning condition follows.

Definition 2.3 *Let G be a 2CRGI and $\mathrm{W} \subseteq \mathrm{St}$ a set of target states. Player$_0$ wins the game G, under the reachability condition, if there is no blocking tree for Player$_0$.*

3 Automata theoretic solution

For the solution side, we use an automata-approach via *alternating tree automata* (*ATA*). Specifically, an *ATA* is a tuple $A \triangleq < \Sigma, D, Q, q_0, \delta, F >$, where Σ is the alphabet, D is a finite set of directions, Q is the set of states, $q_0 \in Q$ is the initial state, $\delta : Q \times \Sigma \to \mathcal{B}^+(D \times Q)$ is the transition function, where $\mathcal{B}^+(D \times Q)$ is the set of all positive Boolean combinations of pairs (d, q) with d direction and q state, and $F \subseteq Q$ is the set of accepting states. An *ATA* A recognizes (finite) trees by means of runs. For a Σ-labeled tree $<T, V>$, with $T = D^*$, a run is a $(D^* \times Q)$-labeled N-tree $<T_r, r>$ such that the root is labeled with (ε, q_0) and the labels of each node and its successors satisfy the transition relation. A run is *accepting* if all its leaves are labeled with accepting states. An input tree is accepted if it admits an accepting run. By $L(A)$ we denote the set of trees accepted by A. We say that A is not empty if $L(A) \neq \emptyset$.

The solution we propose is to read a $\{\top, \bot\}$-labeled full $([\mathrm{Ac}_0] \times \mathrm{Ac}_1)$-tree such that more copies of the automaton are sent to the same directions along the class of equivalence over $[\mathrm{Ac}_0]$. These trees are taken with depth greater than the number of states; so if no state in W is reached in $|\mathrm{St}|$ step, then there is a loop over the states in the game model that forbids to reach states in W.

Theorem 1 *Given a 2CRGI G the problem of deciding whether Player$_0$ wins the game is in* ExpTime-complete.

Proof sketch. Let G be a *2CRGI*. For the lower bound, we recall that two-player turn-based games with imperfect information is ExpTime-hard [11]. For the upper bound, we can build in linear time an automaton accepting all blocking trees for *Player*$_0$. The automaton uses as set of states $Q = \mathrm{St} \times \mathrm{St} \times \{\top, \bot\} \times \{0, 1\}$ and alphabet $\Sigma = \{\top, \bot\}$. Note that, we use in Q a duplication of game states as we want to remember the game state associated to the parent node while traversing the tree. For the initial state we set $q_0 = (s_I, s_I, \top, 0)$, *i.e.* for simplicity the parent game state associated to the root of the tree is the game state itself. The flag $f \in \{0, 1\}$ indicates whether along a path we have entered a target state, in that case we move f from 0 to 1. The transition relation is defined as: $\delta((p, q, \top, f), \top) = \bigwedge_{a_0 \in \mathrm{Ac}_0} \bigwedge_{a_1 \in \mathrm{Ac}_1} (d, (q, q', t', f'))$, $\delta((p, q, t', f), \bot) = true$, $\delta((p, q, \bot, f), \top) = false$; where $q' = tr(q, a_0, a_1)$, $t' \in \{\top, \bot\}$, $d = [\mathrm{Ac}_0] \times [\mathrm{Ac}_1]$, and $f' = 1$ if $q' \in \mathrm{W}$ otherwise $f' = f$. The set of accepted states is $\mathrm{F} = \{(p, q, t, f) : p, q \in \mathrm{St} \wedge t = \top \wedge f = 0\}$. Recall that an input tree is accepted if there exists a run whose leaves are all labeled with accepting states. In our setting this means that an input tree simulates a blocking tree for *Player*$_0$. So, if the automaton is empty then *Player*$_0$ wins the game, *i.e.*, does not exists a blocking tree for him. The required computational complexity of the solution follows by considering that: *(i)* the size of the automaton is polynomial in the size of the game, *(ii)* to check its emptiness can be performed in exponential time [4, 8].□

4 Discussion and future work

Game theory is a useful framework largely applied in AI [13]. Worth of mentioning are the contributions in open-system verification, where the goal is to check whether a system can avoid to reach a bad state no matter how the unpredictable environment behaves, while acting under perfect or imperfect information [7, 9]. Along this line of research, we have considered in this paper the case in which the two players, *Player*$_0$ and *Player*$_1$, move concurrently, under partial visibility. The reachability condition is interpreted by looking at the ability of *Player*$_1$ to prevent *Player*$_0$ to reach a target state. In this case *Player*$_0$ loses the game (and wins otherwise). We have proved that this can be checked in ExpTime and showed that this result is tight.

Our game setting is very useful in multi-agent open-system verification. In particular, it would be useful to consider the case of multiple players along with some reacher acceptance condition, such as a logic for the strategic reasoning (ATL* [2], Strategy Logic [10], and like). Of course one has to consider some restrictions as the imperfect information immediately let the decision problem to jump to nonelementary or even undecidability.

References

[1] L. De Alfaro and T.A. Henzinger, 'Concurrent omega-regular games', in *LCS'00*, pp. 141–154. IEEE, (2000).

[2] R. Alur, T.A. Henzinger, and O. Kupferman, 'Alternating-Time Temporal Logic.', *JACM*, **49**(5), 672–713, (2002).

[3] J.R. Buchi and L.H. Landweber, *Solving sequential conditions by finite-state strategies*, Springer, 1990.

[4] E.A. Emerson and C.S. Jutla, 'The Complexity of Tree Automata and Logics of Programs (Extended Abstract).', in *FOCS'88*, pp. 328–337. IEEE Computer Society, (1988).

[5] W. Jamroga and A. Murano, 'On Module Checking and Strategies.', in *AAMAS'14*, pp. 701–708. IFAAMAS, (2014).

[6] W. Jamroga and A. Murano, 'Module checking of strategic ability', in *AAMAS'15*, pp. 227–235. IFAAMAS, (2015).

[7] O. Kupferman and M. Y. Vardi, 'Module checking revisited', in *CAV'97*, volume 1254 of *LNCS*, pp. 36–47. Springer, (1997).

[8] O. Kupferman, M.Y. Vardi, and P. Wolper, 'An Automata Theoretic Approach to Branching-Time Model Checking.', *JACM*, **47**(2), 312–360, (2000).

[9] O. Kupferman, M.Y. Vardi, and P. Wolper, 'Module Checking.', *IC*, **164**(2), 322–344, (2001).

[10] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi, 'Reasoning About Strategies: On the Model-Checking Problem.', *TOCL*, **15**(4), 34:1–42, (2014).

[11] John H. Reif, 'The complexity of two-player games of incomplete information', *J. Comput. Syst. Sci.*, **29**(2), 274–301, (1984).

[12] Wolfgang Thomas, 'Infinite trees and automaton definable relations over omega-words', 263–277, (1990).

[13] M. Wooldridge, *An Introduction to Multi Agent Systems*, John Wiley & Sons, 2002.

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

doi:10.3233/978-1-61499-672-9-1688

Shape Invariant Formulation for Change Point Models in Multiple Dimensions

Jesse Glass[1] and **Zoran Obradovic**[1]

Abstract.

Artificial intelligence has achieved superhuman performance in a variety of tasks. Unfortunately, this is often done without interpretable methods. In medicine, it is not sufficient to have an algorithm with maximum accuracy. The methods must be evaluated by experts. Our motivating task is to detect features of the eye using unlabeled data. We detect features using a Bayesian changepoint model. Changepoint detection can perform object recognition when applied to two dimensional feature spaces such as images. We present a formula for detecting any shape with the changepoint model. We then extend it to multiple features in order to capture the color of an object. The work presented here has the ability to incorporate prior information, the ability to handle images of varying granularity, can provide confidence estimates on object features in an image. It will serve as the foundation for a classification method with interpretable results.

1 INTRODUCTION, BACKGROUND, & RELATED WORK

The task of predicting the severity of diabetic retinopathy using images of the retina was a recent Kaggle competition. A significant majority of the top 100 performers used Deep Learning approaches. This approach has several drawbacks. When applying Convolutional Neural Networks, the pixel sizes of the images must be the same. This requires that many of the images are downgraded, which results in information loss. Our approach can learn on varying granularities which potentially enables us to transfer knowledge learned on more detailed images to the less detailed ones.

Another drawback with Deep Learning Models, is a lack of interpretability. Our model provides the location, color, and size of an object with probability estimates for any combination. Detecting objects such as the basal ganglion helps with the prediction task of labeling veins because vascular growth stems out from the basal ganglion. Similarly detecting the macula helps during diagnosis because stage 2 retinopathy is most often diagnosed by determining whether a hemorrhage is blocking light from entering the macula. Importantly, the accuracy of the detection of a macula and hemorrhage can be easily verified by physicians.

The Gamma-Poisson conjugate prior has a closed form Bayesian posterior for the changepoint problem. The Poisson distribution is an obvious choice for image pixel value prediction since a Poisson distribution outputs a natural number and images are coded with integers from 0 to 255 on three color channels.

Bayesian models also give us a unique opportunity to incorporate prior information. Then evaluate/update that prior information. The images in this data set have a great variety some are dark, some have glare, some have artifacts. A well designed model will not require preprocessing to handle these.

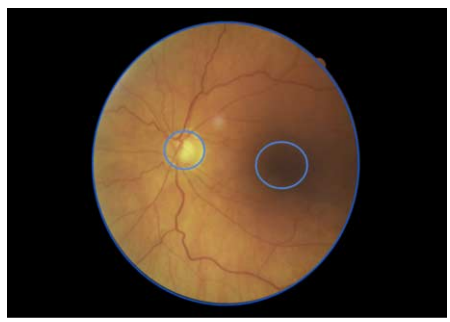


Figure 1. Retina (whole), Optic Nerve (light), Macula (dark)

Bayesian methods can often seem too specific for adaptation to multiple purposes. [1] derived an exact solution for multiple features for changepoints along a single dimension. [2] used heuristics to model a single feature, with the potential to model stacked signals. Bayesian models also have a much better ability to learn confidence. Heuristics can work but often fail miserably in outlier cases. So, we developed a fully Bayesian extension of both of the discussed models.

2 METHOD

Given the following Gamma priors on our average values and Poisson priors on our observations we will develop a Gibbs sampling algorithm for multiple features on arbitrary shapes. We allow different objects within the image (segments) and the different color channels to have different priors (a,b). The notation is: S, shape; f, feature (RGB layer); ij, position in image. The generative model is as follows:

$$\lambda_i^{(f)} \sim Gamma(a,b)$$

$$x_{ij}^{(f)} \sim \begin{cases} Poisson(x_{ij}^{(f)}|\lambda_1^{(f)}) & i,j \notin S \\ Poisson(x_{ij}^{(f)}|\lambda_2^{(f)}) & i,j \in S \end{cases}$$

The problem of inferring the posterior over the latent variables S, λ_1, λ_2 can be solved via Bayes theorem. The full joint distribution

$$p(\vec{\lambda}_1,\vec{\lambda}_2,S|\boldsymbol{X}) \propto p(\boldsymbol{X}|\vec{\lambda}_1)p(\boldsymbol{X}|\vec{\lambda}_2)p(\vec{\lambda}_1)p(\vec{\lambda}_2)p(S)$$
$$\propto \prod_{f=1}^{M}(\prod_{ij\notin S} p(x_{ij}^{(f)}|\lambda_1^{(f)}) \prod_{ij\in S} p(x_{ij}^{(f)}|\lambda_2^{(f)})p(\lambda_1^{(f)})p(\lambda_2^{(f)}))p(S).$$

The probability of the shape depends on the number of features required for the shape. We define a square by its start and end points.

[1] Temple University, Center for Data Analytics and Biomedical Informatics, email: tud25892@temple.edu, zoran.obradovic@temple.edu

We define a circle by is center, (x,y), and radius, r.

$$p(S) = p(x)p(y)p(r)$$

We see that each neighboring x_{ij} is independent of its neighbors given that it is drawn from a known underlying gamma process. This enables us to plug known prior distributions into the above formula.

Obtain the posterior conditionals for each latent variable by collecting the terms in the full joint distribution that include the latent variable of interest. Derive the posterior conditionals for each of the variables S, λ_1, λ_2 by selecting relevant components from the full joint distribution. When calculating the conditional probability we can marginalize constants out of $p(\lambda_1|\cdot)$.

$$p(\lambda_1^{(f)}|S, \vec{\lambda}_1^{-f}, \vec{\lambda}_2, \boldsymbol{X}) \propto Gamma(a + \sum_{ij \in S}(x_{ij}^f), \sum_{ij \in S}(1) + b)$$

$$p(\lambda_2^{(f)}|S, \vec{\lambda}_2^{-f}, \vec{\lambda}_1, \boldsymbol{X}) \propto Gamma(a + \sum_{ij \notin S}(x_{ij}^f), \sum_{ij \notin S}(1) + b)$$

We can see that each $\lambda^{(f)}$ is independent of one another. Calculating the Multinomial Distribution of $p(S|\vec{\lambda_1}, \vec{\lambda}_2, \boldsymbol{X})$ is a bit more complex, and must be done separately for each component of S. We defined the properties of a circle as two center points and a radius. They all have the similar formulations. We present the multinomial formula for the probability of the radius below.

$$\begin{aligned} log(p(r|\vec{\lambda_1}, \vec{\lambda}_2, \boldsymbol{X})) =^+ & \sum_{f=1}^{M}(log(\lambda_1^{(f)}) \sum_{ij \in S} x_{ij}^{(f)}) + \\ & - M \cdot \sum_{ij \in S}(1) \cdot \sum_{f=1}^{M} \lambda_1^{(f)} + \\ & + \sum_{f=1}^{M}(log(\lambda_2^{(f)}) \sum_{ij \notin S} x_{ij}^{(f)}) - M \cdot \sum_{ij \notin S}(1) \cdot \sum_{f=1}^{M} \lambda_2^{(f)}. \end{aligned}$$

After calculating each possible solution, we take the exponent and then normalize.

3 EXPERIMENTS & RESULTS

A set of synthetic experiments were devised in increasing complexity to examine this approaches ability to handle the data. First, we display a change point algorithm run for one and two change points on 1-D data. Then, the models for those are extended to the two dimensional case - the square. Then, we change the shape to a circle. Last, three features are incorporated endowing objects in the image with a color. For all of the synthetic experiments, the latent values of the objects were learned without mistakes in 100% of trials.

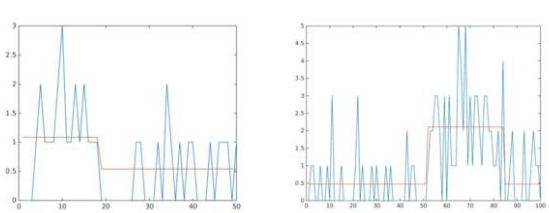


Figure 2. Single Change Point (left) & Segment Shift (right)

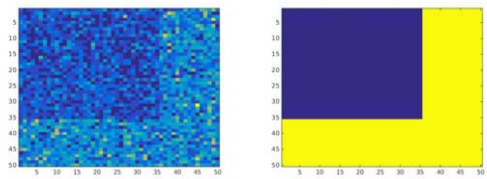


Figure 3. Two Dimensional Change Point

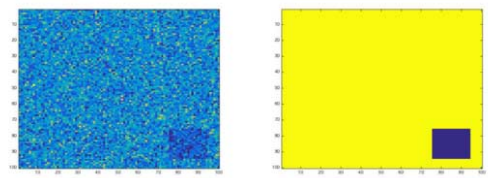


Figure 4. Two Dimensional Segment Shift (a.k.a. Square Detection)

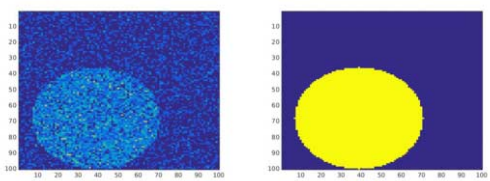


Figure 5. Two Dimensional Circle Detection

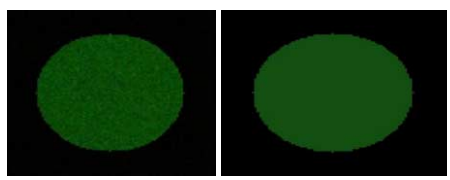


Figure 6. Color Circle Detection

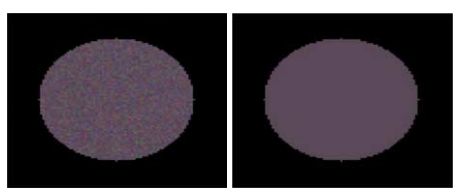


Figure 7. Color Circle Detection

4 CONCLUSION

[1] and [2] are both one dimensional solutions. [2] attempts to handle overlapped signals using heuristics while [1] develops an exact solution for a similar problem but does not handle stacked signals. The next step in this work is to handle stacked signals. Once we can identify the key circular features of the eye, we can attempt an approach for the vasculature and detecting hemorrhages.

ACKNOWLEDGEMENTS

This research was supported in part by DARPA grant FA9550-12-1-0406 negotiated by AFOSR, NSF BIGDATA grant 14476570, ONR grant N00014-15-1-2729 and SNSF Joint Research project (SCOPES), ID: IZ73Z0_152415.

REFERENCES

[1] Paul Fearnhead, 'Exact and efficient bayesian inference for multiple changepoint problems', in *Statistical Computing*, pp. 203–213, (2006).

[2] W.R. Lai, M.D. Johnson, R. Kucherlapati, and P.J. Park, 'Comparative analysis of algorithms for identifying amplifications and deletions in array cgh data', in *Bioinformatics*, pp. 3763–3770, (2005).

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

doi:10.3233/978-1-61499-672-9-1690

Shaping Proto-Value Functions Using Rewards

Raj Kumar Maity[1], **Chandrashekar Lakshminarayanan**[1], **Sindhu Padakandla**[1], and **Shalabh Bhatnagar**[1]

Abstract. In reinforcement learning (RL), an important sub-problem is learning the value function, which is chiefly influenced by the architecture used to represent value functions. is often expressed as a linear combination of a pre-selected set of basis functions. These basis functions are either selected in an ad-hoc manner or are tailored to the RL task using the domain knowledge. Selecting basis functions in an ad-hoc manner does not give a good approximation of value function while choosing functions using domain knowledge introduces dependency on the task. Thus, a desirable scenario is to have a method to choose basis functions that are task independent, but which also provide a good approximation for the value function. In this paper, we propose a novel task-independent basis function construction method that uses the topology of the underlying state space and the reward structure to build the reward-based Proto Value Functions (RPVFs). The approach we propose gives good approximation for the value function and enhanced learning performance. The performance is demonstrated via experiments on grid-world tasks.

1 INTRODUCTION

Reinforcement learning (RL) is a sequential decision making paradigm to solve adaptive control problems under model uncertainties. RL problems are cast in the framework of Markov decision processes (MDPs). In the MDP setting, dynamics of the underlying environment evolves within a set of states called the state-space, and the agent performs actions to control the state of the system. Depending on the state and the action the agent performs in that state, the agent receives a reward. The agent aims to maximize the infinite sum of discounted rewards obtained as a result of its actions. An action selection mechanism is known as a policy and the agent aims to learn an optimal policy. To learn the optimal policy, the agent needs to be able to evaluate a policy. The value function J_u corresponding to a given policy u, is a map from the state space to real numbers, and captures the total discounted reward that agent collects by following the policy u. Using the value function, the agent can improve its policy and hence learning the value function in an efficient manner assumes importance.

The value function representation has to be *compact* (i.e., it should be easy to compute and store). Usually, the value function is represented as a linear combination of basis functions, which are generally chosen using task-specific knowledge and when there is no such task-specific information, primitive hand-coded functions ([3, 4]) are used. However, the primary objective is to formulate a method that selects a compact representation for effectively approximating the value function with little or no information about the task. Hand-coded basis functions do not usually capture the connectivity information which can be exploited to optimally control the underlying MDP. Thus, there is a need for algorithms which build and optimize basis functions by leveraging the geometry of the state space and the reward (or cost) structure. Aiming at this problem, we develop the Shaping-based Representation Policy Iteration (SRPI) algorithm.

In this paper, we present the reward-shaping based graph Laplacian framework. Immediate rewards indicate beneficial transitions. Using the knowledge of transitions which yield good and bad rewards, we construct the graph underlying the MDP and the Laplacian by appropriately shaping (see [1]) the rewards. We also incorporate risk-sensitivity [2] for rewards in the construction of the Laplacian. Thus, the graph so constructed is *sensitive to immediate rewards*. Spectral analysis of the Laplacian gives us the required basis functions. Through this construction, we show that the learning agent is prevented from taking actions which lead it to unfavourable states. Simulation results and experiments show that the error in value function approximation is minimized when SRPI is used as compared to previous approaches [6, 7].

[1] Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore, India, Email: {rkmaityju,chandrurec5,sindhupr}@gmail.com

2 REWARD-SHAPING BASED GRAPH LAPLACIAN FRAMEWORK

Let $G = (E, V)$ denote a graph with edge set E and vertex set V. W is a weight matrix for G. Let D denote the diagonal matrix whose diagonal entries are row sums of W. For a fixed policy, we can construct the matrices W, D for a MDP. To construct the weight matrix, we obtain trajectory samples. From the samples, we also get the immediate rewards which can be used to capture preferred connectivity. While in the case of goal oriented tasks, the immediate rewards are 0 for non-goal states, it might not hold true for MDPs with a general reward structure. The agent's decision can be influenced by shaping the immediate rewards. The shaped rewards must preserve the optimal policy structure which can be guaranteed only if the reward shaping functions satisfy certain conditions and are potential functions (see [1]).

We denote the shaped reward for state transition from a state s to s' as $\bar{R}(s, s'), \forall s, s' \in S$. Using the shaped rewards $\bar{R}$, we define the weight matrix W_R in a manner which prevents the agent from choosing unfavourable states. The weight matrix so defined captures the nearness of states which is affected by the underlying reward structure. Let $W_R = (w_r(s, s'), s \in S, s' \in S)$, where

$$w_r(s, s') = \frac{\exp^{\beta \bar{R}(s')}}{\sum_{s'' \in S} \exp^{\beta \bar{R}(s'')}}. \quad (1)$$

In (1), $\beta > 0$ is a positive constant that models affinity. The parameter β allows one to incorporate aversion ($\beta < 0$) or preference ($\beta > 0$) to risk. Using the weight matrix W_R, we compute the matrix D, find the combinatorial and normalized Laplacian operators and the diffusion matrix (see [6]). The basis functions for the state-action value function are selected by analyzing the spectra of the random walk diffusion matrix D_R. We choose the top m eigen values of D_R and the corresponding eigen vectors to form the matrix Ψ. The i^{th} row of the matrix $\Psi_{|V| \times m}$, i.e., $\psi(i)$, $i \in 1, 2, \ldots, |V|$, is the feature vector of state s_i, that is in the vertex set V. Hence, $\psi(s_i) = \psi(i)$. The state-action feature matrix Φ is obtained as indicated in [4]. Let $\phi(s_i, a)$ be the feature vector corresponding to the state-action pair (s_i, a). Thus the action-value function $Q^\pi(s, a)$ for a policy π is approximated as:

$$Q^\pi(s, a) = \sum_{i=1}^{k} \phi_i(s, a) w^\pi(i), \quad (2)$$

where $w^\pi \in \mathrm{R}^k$ is the weight vector. The SRPI algorithm is given in Algorithm 1, wherein the approximate value function is evaluated using the LSTDQ algorithm [4] (a variant of LSTD algorithm). The integer t used in policy iteration in SRPI is chosen large enough to ensure convergence.

Algorithm 1 SRPI

1: Collect a set of (state,action,reward) samples $\mathcal{D}_S$ using a random policy π_0
2: Construct W_R and D_R
3: Generate eigen vectors of D_R and construct Φ
4: **for** $i = 0, 1, 2, \ldots, t-1$ **do**
5: $\quad w^{\pi_i} = LSTDQ(\mathcal{D}_S, \pi_i, k)$, where $\hat{Q}^{\pi_i} = \Phi w^{\pi_i}$
6: $\quad$ Set $\pi_{i+1}(s) = \arg\max_{a \in A}(\hat{Q}^{\pi_i}(s,a)), \forall s \in S$
7: **end for**
8: Return π_t

3 EXPERIMENTS

We consider an instance of goal-based MDP (see Table 1), which is a $N \times N$ grid. The state space is given by $S = \{s = (x, y), x = 1, \ldots, N, y = 1, \ldots, N\}$, where $(1, 1)$ denotes the *bottom-left* cell and (N, N) the *top-right* cell. The allowable actions are to move *up, down, right* or *left*. (N, N) is the goal-state. The agent receives a reward of 10 on reaching the goal state and actions in the intermediate states do not yield any reward. It is evident that the learning process can be sped up if the agent is rewarded for those actions that take it either *up* or *right*.

				G
		↑		
	←		→	
		↓		

5	10	15	20	25
4	9	14	19	24
3	8	13	18	23
2	7	12	17	22
1	6	11	16	21

Table 1: On the left is the grid world task with a reward of 10 in the goal-state **G**. On the right is the potential reward shaping function.

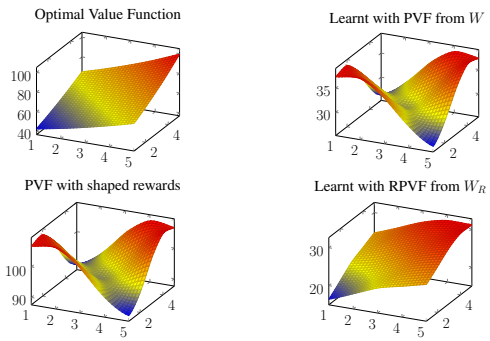


Figure 1: Shows the optimal value function (left, top row) and the learned value functions (other three) for the grid world domain in Table 1. Notice that learning using the RPVF (right, bottom row) is better than the learning using PVF with/without reward shaping.

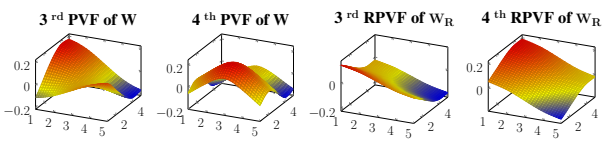


Figure 2: Comparing the profiles of eigen functions of W and W_R. The first two eigen functions of W and W_R were identical and hence not presented.

We consider a grid for $N = 5$. The optimal value function is shown in the top most plot of Fig. 1. We chose 4 eigen functions corresponding to 4 largest eigen values of the diffusion matrix D_R constructed from the usual adjacency matrix W. We ran the RPI algorithm [6] with the diffusion matrix constructed from A and the result is shown (top-right) in Fig. 1. Notice that the value function learnt by the RPI algorithm does not quite resemble the profile of the optimal value function and consequently resulted only in a moderately good policy. We evaluated the policy π_W returned by RPI and it turned out that $\sum_{s \in S} J_{\pi_W}(s) = 1132$ as opposed to $\sum_{s \in S} J^*(s) = 1887$. Further, we also ran the RPI algorithm by retaining the diffusion matrix, but provided additional reward shaping feedback using the shaping function $\tilde{R}$ during the learning phase (see Table 1). Even in this case the profile of the learnt value function did not change, and the resulting policy performed only moderately. The SRPI algorithm was run for this experiment by using the shaped rewards for constructing the weight matrix W_R. In this case, the profile of the learnt value function resembles the optimal value function. Further, we also observed that the policy π_{W_R} returned by SRPI in this case performed better than π_W (with/without reward shaping feedback), i.e., $\sum_{s \in S} J_{\pi_{W_R}}(s) = 1660$. The reason why the RPVFs perform better than the PVFs can be explained by looking at the corresponding eigen functions (see Fig. 2). The PVFs and RPVFs corresponding to the first two largest eigen values were the same. However, the third and fourth eigen functions differed (see Fig. 2). We can see from (Figures 1 and 2), that this difference in eigen function shows up in the difference in the profiles of the corresponding learnt value functions.

4 CONCLUSION

This paper presents a new reward transformation based graph Laplacian framework for basis representation, which can be used to find a *compact* representation for the value function. The performance of SRPI is compared with RPI for the grid world problem. It is shown that the RPVFs capture the neighbourhood information well when compared to PVFs. The experimental results show that the SRPI algorithm finds better basis compared to RPI. It is also observed that similarity matrices other than the diffusion matrix can be used to generate the basis. Reward transformation is shown to aid learning, and the SRPI algorithm is seen to outperform the policy learnt using PVFs. An extended version of this work is available at [5].

REFERENCES

[1] Tim Brys, Anna Harutyunyan, Matthew E Taylor, and Ann Nowé, 'Policy transfer using reward shaping', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 181–188, (2015).
[2] Stefano P. Coraluppi and Steven I. Marcus, 'Risk-sensitive and minimax control of discrete-time, finite-state markov decision processes', *Automatica*, **35**(2), 301 – 309, (1999).
[3] G. Konidaris, S. Osentoski, and P. S. Thomas, 'Value function approximation in reinforcement learning using the Fourier basis.', in *AAAI*, (2011).
[4] M. G. Lagoudakis and R. Parr, 'Least-squares policy iteration', *Journal of Machine Learning Research*, **4**, 1107–1149, (2003).
[5] Raj K M, Chandrashekar L, S Padakandla, and Shalabh B, 'Shaping proto value functions using rewards'. `http://stochastic.csa.iisc.ernet.in/www/research/files/IISc-CSA-SSL-TR-2016-2.pdf`.
[6] S. S. Mahadevan and M. Maggioni, 'Proto-value functions: A Laplacian framework for learning representation and control in Markov decision Processes.', *Journal of Machine Learning Research*, **8**(16), 2169–2231, (2007).
[7] X. Xu, Z. Huang, D. Graves, and W. Pedrycz, 'A clustering-based graph laplacian framework for value function approximation in reinforcement learning', *IEEE Transactions on Cybernetics*, 2613–2625, (2014).

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

doi:10.3233/978-1-61499-672-9-1692

Heuristic Constraint Answer Set Programming

Erich C. Teppan[1] and **Gerhard Friedrich**[1]

Abstract. Constraint answer set programming (CASP) is a family of hybrid approaches integrating answer set programming (ASP) and constraint programming (CP). These hybrid approaches have already proven to be very successful in various domains. In this paper we present first evaluation results for the CASP solver ASCASS, which provides novel methods for defining and exploiting problem-dependent search heuristics. Beyond the possibility of using already built-in problem-independent heuristics, ASCASS allows on the ASP level the definition of problem-dependent variable selection, value selection and pruning strategies, which guide the search of the CP solver. The proof-of-concept evaluation was carried out on benchmark instances of the real world Partner Units Problem (PUP). Due to a sophisticated heuristic, which cannot be represented by other ASP or CASP solvers, ASCASS shows superior performance.

1 Introduction

During the last decade, Answer Set Programming (ASP) under the stable model semantics [5] has evolved to an extremely powerful approach for solving combinatorial problems. Especially conflict-driven search mechanisms contribute to the high performance of state-of-the-art solvers [4]. Furthermore, ASP provides superior problem encoding capabilities as ASP is strongly declarative in nature and even provides language features, which go beyond first order.

However, the expressive power on the one hand and the potent conflict-driven search approach on the other hand do not come for free. Current ASP solvers employing conflict-driven search transform the higher-order problem representation to propositional logic. This transformation (called grounding) constitutes the space bottleneck of current ASP systems. Once the grounding step is completed, conflict-driven search in combination with state-of-the-art look-back heuristics like VSIDS and restarts [6] typically shows superior performance compared to other search approaches. Yet, grounding is not possible for many industrial-sized problem instances.

One approach that emerged also out of the need of easing the grounding was Constraint Answer Set Programming (CASP) [8, 2]. CASP can be seen as a hybrid approach extending ASP by Constraint Programming (CP) features. Conceptually, it is very close to satisfiability (SAT) modulo theory approaches, which integrate first-order formulas with additional background theories such as real numbers or integers [9].

For combining ASP and CP there are basically two approaches. First, solvers like Clingcon [8] are based on the extension of the ASP input language in order to support the formulation of constraints. A different approach has been introduced by the Ezcsp solver [2] where ASP and CP are not integrated into one language. ASP rather acts as a specification language for Constraint Satisfaction Problems (CSPs). The main idea is that answer sets constitute CSP encodings, which are used as input for a CP solver.

For certain classes of problems like industrial-sized scheduling CASP was already successfully applied [3]. Especially search problems with large variable domains often profit from the CASP representation due to the alleviation of the grounding bottleneck [7].

Of course, the complexity of problem solving does not vanish by easing the grounding bottleneck but it is rather shifted from grounding in ASP to search in CP. In the context of CASP, often a majority of the solution calculation is done by the CP solver. Hence, the applied search strategies on the CP level play a crucial role for the successful application of CASP in real-world problem domains. However, up to now there was no focus on the development of sophisticated features for expressing and exploiting search strategies in CASP solvers. Consequently, the means for expressing and exploiting search strategies on the CP level are rather limited.

Clingcon[2] and Ezcsp[3] provide a set of built-in problem-independent strategies depending on the underlying CP solver. Clearly, for many real-world problem domains problem-independent strategies are not sufficient and problem-dependent heuristics are needed. Any problem-dependent heuristic on the CP level basically consists of three components:

1. a problem-dependent variable selection strategy
2. a problem-dependent value selection strategy
3. a problem-dependent pruning strategy

In EZCSP, problem-dependent variable selection strategies are already supported by a special label-ordering predicate. What is missing is the possibility of expressing custom value ordering and pruning strategies.

2 A Simple Constraint Answer Set Solver

ASCASS[4] is a finite discrete CASP solver following the approach of Ezcsp, i.e. the input language is pure ASP and the answer sets encode CSPs. Answer set production (grounding and solving) is done by Clingo[5], which is currently one of the most powerful ASP systems. The input language is the ASP standard ASP-Core-2[6].

After answer set solving, a produced answer set is handed over to a parsing module that extracts the facts, which encode the CSP and search directives. This information is used to instantiate a corresponding CSP in the CP solver and perform search conforming to the given search directives. Currently, Jacop[7] is used within ASCASS as a CP solver. In case that the CSP could not be solved by

[1] Universität Klagenfurt, Austria, email: firstname.lastname@aau.at

[2] www.cs.uni-potsdam.de/clingcon/
[3] mbal.tk/ezcsp/index.html
[4] http://isbi.aau.at/hint/ascass
[5] sourceforge.net/projects/potassco/files/clingo
[6] www.mat.unical.it/aspcomp2013/files/ASP-CORE-2.03b.pdf
[7] jacop.osolpro.com

the CP solver or a timeout occurred (defined by the special predicate $csptimeout(\Delta)$), the process continues with the next answer set, until a solution is found, or there are no more answer sets. The empty CSP (i.e. when there is not a single CSP variable) is always satisfiable and possesses the empty CSP solution.

ASCASS combines and extends the heuristic possibilities of state-of-the-art CASP solvers and makes them completely available on the problem encoding level. Beyond the usage of built-in strategies, ASCASS provides powerful constructs for the formulation and exploitation of problem-dependent heuristics consisting of variable selection, value selection and pruning strategies.

3 The Partner Units Problem

We want to exemplify the expressive power of ASCASS with respect to the partner units problem (PUP) out of three reasons.

1. The PUP is a real world combinatorial problem with many different application domains [1].
2. The PUP is one of the hardest benchmark problems participating in the ASP competitions[8].
3. There exists an effective and non-trivial problem-dependent heuristic to solve the PUP.

The PUP originates in the domain of railway safety systems. One of the problems in this domain is to make sure that certain rail tracks are not occupied by a train/wagon before another train enters this track. The signals for the corresponding occupancy indicators are calculated by special processing units based on the input of several observing sensors. The problem consists in connecting sensors/indicators with units and units with other units such that all communication requirements are fulfilled and the number of maximal connectors of the devices is not exceeded.

The state-of-the-art heuristic for solving the PUP is the QuickPup heuristic proposed in [10]. QuickPup can be expressed quite naturally in ASCASS but, to the best of our knowledge, cannot be expressed in any other ASP or CASP approach.

4 Evaluation

We tested the ASP solver Clingo 4 and the CASP solvers ASCASS, Clingcon and Ezcsp on the PUP benchmark suite used in [1][9]. Clingo was tested using the PUP encoding proposed in [1]. The tests were run on a 3.2 Ghz machine with 64 GByte of RAM, assuring that the grounding bottleneck does not play a role for the tested instances and performance can be attributed to the search phase.

In the Clingcon model, problem-dependent CSP variable selection, value selection or pruning strategies cannot be exploited. For Ezcsp, it is possible to express the topological variable orderings similar to ASCASS. However, there are no means for pruning search or problem-dependent value strategies.

Table 1 depicts how many instances of each type in the benchmark suite could be solved by the different approaches within a 1000 seconds time frame. Clingo using VSIDS heuristic peformed very well on the benchmark suite showing once again that the conflict-driven search techniques employed by Clingo are quite powerful. Also Ezcsp was able to solve some instances. Using other built-in heuristics did not result in better performance. Clingcon was not able to solve a single instance. In the contrary, ASCASS was able to solve all but one instances within time limits. We want to point out that only optimal solutions (i.e. minimum number of units) were allowed for easing the grounding bottleneck of conventional ASP. Increasing the number of allowed units in a solution would increase grounding size for ASP significantly. In the cases of ASCASS and Ezcsp, increasing the number of allowed units would not affect the grounding size as the number of allowed units is captured by the upper bounds of the CSP variables.

	#	Clingo	ASCASS	Clingcon	Ezcsp
double(IUCAP=2)	10	2	10	0	2
doublev(IUCAP=2)	6	3	6	0	0
triple(IUCAP=2)	3	2	3	0	2
triple(IUCAP=4)	7	6	6	0	3
grid(IUCAP=4)	10	10	10	0	0
total	36	23	35	0	7

Table 1. Solved instances whithin 1000 seconds

Furthermore, we want to make clear that the superior performance of ASCASS can be attributed to the inclusion of the QuickPup strategies. This was crosschecked by removing the heuristic parts from the ASCASS problem encodings. It is to be noted that QuickPup originally was designed for producing only near-optimal solutions. However, the concepts of QuickPup obviously also work well for finding optimal solutions.

[8] Further information can be found at www.mat.unical.it/aspcomp2014/

[9] Encodings and benchmark instances can be found at http://isbi.aau.at/hint/ascass

REFERENCES

[1] M. Aschinger, C. Drescher, G. Friedrich, G. Gottlob, P. Jeavons, A. Ryabokon, and E. Thorstensen, 'Optimization methods for the partner units problem', in *CPAIOR'11*, pp. 4–19, Berlin, Heidelberg, (2011). Springer-Verlag.

[2] M. Balduccini, 'Representing constraint satisfaction problems in answer set programming', in *ICLP09 Workshop on Answer Set Programming and Other Computing Paradigms (ASPOCP'09)*, (2009).

[3] Marcello Balduccini, 'Industrial-size scheduling with asp+cp', in *Proceedings of the 11th Int. Conf. on Logic Programming and Nonmonotonic Reasoning*, LPNMR'11, pp. 284–296, Berlin, Heidelberg, (2011). Springer-Verlag.

[4] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub, *Answer Set Solving in Practice*, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool Publishers, 2012.

[5] Michael Gelfond and Vladimir Lifschitz, 'The stable model semantics for logic programming', in *Proceedings of the Fifth Int. Conf. and Symp. of Logic Progr. (ICLP'88)*, eds., R. Kowalski and K. Bowen, pp. 1070 – 1080. MIT Press, (1988).

[6] Matthew D. T. Lewis, Tobias Schubert, and Bernd W. Becker, 'Speedup techniques utilized in modern sat solvers', in *Proceedings of the 8th Int. Conf. on Theory and Applications of Satisfiability Testing*, SAT'05, pp. 437–443, Berlin, Heidelberg, (2005). Springer-Verlag.

[7] Yuliya Lierler, Shaden Smith, Miroslaw Truszczynski, and Alex Westlund, 'Weighted-sequence problem: Asp vs casp and declarative vs problem-oriented solving', in *Proceedings of the 14th Int. Conf. on Practical Aspects of Declarative Languages*, PADL'12, pp. 63–77, Berlin, Heidelberg, (2012). Springer-Verlag.

[8] Max Ostrowski and Torsten Schaub, 'Asp modulo csp: The clingcon system', *Theory Pract. Log. Program.*, **12**(4-5), 485–503, (September 2012).

[9] Roberto Sebastiani, 'Lazy satisfiability modulo theories', *J. on Satisfiability, Boolean Modeling and Computation*, **3**, 141–224, (2007).

[10] Erich Christian Teppan, Gerhard Friedrich, and Andreas Falkner, 'Quickpup: A heuristic backtracking algorithm for the partner units configuration problem.', in *International Conference on Innovative Applications of AI (IAAI'12)*, pp. 2329–2334. AAAI, (2012).

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

doi:10.3233/978-1-61499-672-9-1694

Non-Deterministic Planning with Numeric Uncertainty

Liana Marinescu and **Andrew Coles** [1]

Abstract. Uncertainty arises in many compelling real-world applications of planning. There is a large body of work on propositional uncertainty where actions have non-deterministic outcomes. However handling numeric uncertainty has been given less consideration. In this paper, we present a novel offline policy-building approach for problems with numeric uncertainty. In particular, inspired by the planner PRP, we define a numeric constraint representation that captures only relevant numeric information, supporting a more compact policy representation. We also show how numeric dead ends can be generalised to avoid redundant search. Empirical results show we can substantially reduce the time taken to build a policy.

1 Introduction and Background

Planning accounting for uncertainty is relevant to many interesting real-world problems, for instance where the dynamics of the environment make it difficult to plan on the basis of actions having a single, predictable outcome. In this work we look at *fully observable non-deterministic* (FOND) planning, i.e. applying an action has several possible outcomes, but we can observe which occurs. The task of planning here is to find a policy (a mapping from states to actions) that dictates what to do in each state reachable from the initial state.

Additionally, within this setting, we allow numeric effects to have *continuous* uncertainty; specifically, we allow Gaussian-distributed (independent) effects on variables. Each effect is of the form $v\ op\ \mathcal{N}(\mathbf{w}\cdot\mathbf{v}+k, \sigma^2)$, where v is some state variable; $op \in \{+=, =\}$; $\mathbf{w}\cdot\mathbf{v}$ is a weighted sum of state variables; $k \in \Re$; and $\sigma^2 \in \Re$ is the variance of the effect. In each state we store the mean and variance of each variable: v and $\sigma^2(v)$. For a numeric precondition (or goal) of the form $\mathbf{w}\cdot\mathbf{v} \geq c$, we first compute the variance affecting it (1). The precondition is then true with confidence θ *iff* (2) is satisfied (where Φ is the Gaussian cumulative distribution function).

$$\sigma^2(\mathbf{w}\cdot\mathbf{v}) = \textstyle\sum_{w\cdot v\in\mathbf{w}\cdot\mathbf{v}} w^2\cdot\sigma^2(v) \quad (1)$$
$$\mathbf{w}\cdot\mathbf{v} \geq c + \sigma(\mathbf{w}\cdot\mathbf{v})\cdot\Phi^{-1}(\theta) \quad (2)$$

Effectively, this computes an *offset* we need to add onto the precondition so that, even accounting for uncertainty, it remains true e.g. 99% of the time for $\theta = 0.99$. The task of planning is then to find a policy such that at each point, the preconditions of the action to be applied (or the goals) are true with a given confidence level θ.

Our work builds upon two recent strands of research: a heuristic for planning with continuous numeric uncertainty [5]; and a planner for propositional FOND domains, PRP [8, 7]. In PRP, policies are found by making repeated calls to a deterministic planning kernel, which finds *weak plans* that work assuming the action outcome can be chosen. To then obtain a policy that covers the other outcomes, this process is recursively applied for the other states that could be reached. Key to the success of PRP are two ideas:

[1] Department of Informatics, King's College London, UK
email: firstname.lastname@kcl.ac.uk

First, the policy maps *partial states* (rather than fully-specified states) to actions. These partial states are found by *regression*: every time a weak plan is added to the policy, the goals are regressed through it step-by-step, and at each point a partial-state–action pair is added to the policy. The benefit of regression is that it collects the *weakest preconditions* needed for the tail of the plan to succeed – if a state in a weak plan contains literals that are irrelevant to the remaining steps, these literals will not be included in the partial state.

Second, for states that cannot reach the goal, *dead end generalisation* is used. In the propositional case, this amounts to a sensitivity analysis to check which literals in a state are causing it to be a dead end. If a dead end state S contains f, and evaluating $S \cup \{\neg f\}$ with the Relaxed Planning Graph (RPG) heuristic [4] indicates S is still a dead end, then the truth value of f is irrelevant and can be ignored. Any state found during policy-building that matches a known dead end can then be discarded, avoiding wasted search effort.

Adapting these ideas to the continuous numeric case requires a planning kernel (we use our recent work [5]) and suitable definitions of regression and dead end generalisation. These form the next two sections of this paper, followed by an empirical evaluation.

2 Regression for Numeric Constraints

As mentioned above, in PRP, the policy maps partial states to actions; these partial states are found by regressing through the weak plans given by the planning kernel. Regression begins with the goal state and steps through the actions in reverse order. From a partial state ps', regressing through an action a with propositional preconditions $Pre(a)$ and add-effects $Eff^+(a)$ yields a new partial state:

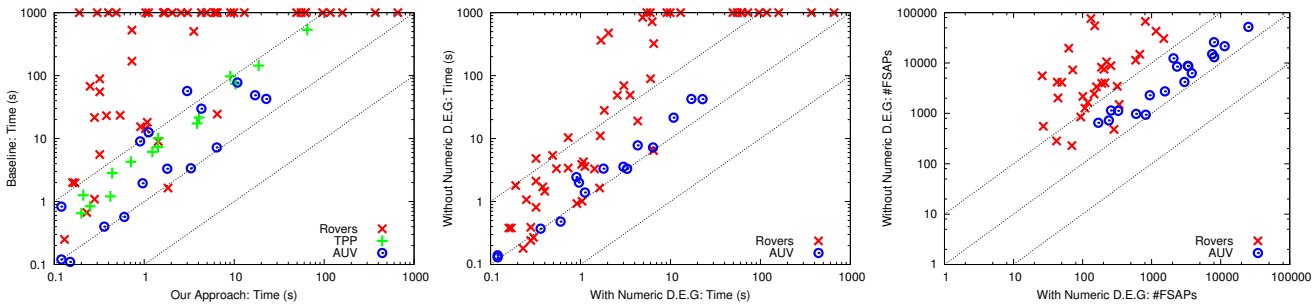
$$ps = (ps' \setminus Eff^+(a)) \cup Pre(a) \quad (3)$$

The policy would then record that a should be applied in any state S where $S \vdash ps$. In effect, ps constrains S in a way that ensures action a and subsequent actions in the weak plan are applicable. To apply this same reasoning for the numeric case, we now derive an analogous constraint representation for *numeric partial states*.

Suppose in state S, a variable v has value $S[v]$ and variance $S[\sigma^2(v)]$. Applying an action a with an uncertain numeric effect $v{+}{=}\mathcal{N}(p, q)$ (where $p, q \in \Re$) reaches a state S', where:

$$S'[v] = S[v] + p \qquad S'[\sigma^2(v)] = S[\sigma^2(v)] + q \quad (4)$$

If we have a precondition $(v \geq c)$ that must be true in S', then $(S'[v] \geq c)$ must be true with confidence θ allowing for variance $S'[\sigma^2(v)]$. By replacing S' with S according to (4), we can rewrite this as $(S[v] + p \geq c)$ allowing for variance $(S[\sigma^2(v)] + q)$. Writing the precondition in this form allows us to derive numeric regression for increase effects: for $(v \geq c)$ to hold in the state *after* a, then $(S[v] + p \geq c)$ must hold allowing for variance $(S[\sigma^2(v)] + q)$ in the state *before* a. Each partial state ps in the policy records numeric constraints in this form. When checking if $S \vdash ps$, these constraints



(a) Time with/without both new techniques (b) Time with/without dead end generalisation (c) FSAPs with/without dead end generalisation

Figure 1. Tests comparing our approach (X axis) to a baseline (Y axis). For the baseline in all cases, only propositional dead end generalisation is used. For the Time to Solve baseline (a), additionally, only propositional regression is used. Tests that timed out are plotted at $t = 1000$. All axes are log scaled.

are evaluated alongside the propositions recorded in (3).

Analogously, applying an assignment effect $v = \mathcal{N}(p, q)$ reaches:

$$S'[v] = p \qquad S'[\sigma^2(v)] = q \qquad (5)$$

Regression in this case is trivial, as there is no reference to S at all. If a precondition $(v \geq c)$ must be true in S', then $(p \geq c)$ must be true with confidence θ allowing for variance q.

The above can be generalised to cases where the precondition refers to a weighted sum of variables, rather than a single variable v; and where p is a weighted sum of variables rather than a constant. Our approach is based on the following intuition (for full details, see [6]). Suppose we have a condition $\mathbf{w} \cdot \mathbf{v} \geq c$ with a non-zero weight on the affected variable v. Regressing this condition does one of three things depending on the type of effect:

- Increase/decrease on v: we update c and the weights in $\mathbf{w} \cdot \mathbf{v}$ according to the variables and values p in the effect. We separately record any extra variance q that now needs to be accounted for.
- Assignment to v: we update c and the weights in $\mathbf{w} \cdot \mathbf{v}$ according to the variables and values p in the effect. We then set the weight on v to zero, effectively removing it from the condition, as regressing through further effects on v would not affect this condition.
- Assignment to $\sigma^2(v)$: for the purpose of evaluating this condition, from this point on we fix $\sigma^2(v)$ to the value q. (NB regressing through further effects on v would not change this variance.)

3 Numeric Dead End Generalisation

In PRP, when building a policy, forbidden state–action pairs (FSAPs) are recorded each time a dead end is found. As noted earlier, the dead end is generalised by using the propositional RPG heuristic, to better allow it to prune fruitless search branches during policy-building.

For the numeric case (with uncertainty), in a dead end state S we store the mean v and variance $\sigma^2(v)$ of each variable. We generalise S by finding a range of values for each variable's mean and variance such that S is still a dead end. This ties in nicely with the structure of the metric RPG heuristic [3], which underlies the heuristic we use in this work [5]. Within the metric RPG, the values of numeric variables are relaxed so they lie in a range rather than taking on fixed values. The upper and lower bounds on each value are then used to optimistically determine whether preconditions are true. Ordinarily, at the start of heuristic computation, the upper and lower bounds on each variable are set to the value the variable takes in the state being evaluated (in our case, the original dead end S). For generalising numeric dead ends as below, we exploit this mechanic to expand the range of still-dead-end values for each variable.

Suppose S is a dead end, and $S[v] = k$; i.e. initially $S[v] \in [k, k]$. We perform interval halving on the upper bound of v in the range $[k, \infty]$, using our heuristic [5], to find the largest v for which S is still a dead end. Likewise, we reduce the lower bound on v using interval halving in the range $[-\infty, k]$. If the revised bounds on v after interval halving are $[-\infty, \infty]$, it means the value of v is irrelevant to whether S is a dead end. Otherwise, for finite bounds, we have expanded the range of still-dead-end values of $S[v]$. This process is repeated greedily, updating S at each step, for each variable and variance.

4 Evaluation

To evaluate our techniques, we compare them to a configuration of our planner where only *propositional* regression and dead end generalisation are used (the values of numeric variables are assumed to be exactly those from the states in the weak plan, or from the dead end, respectively). We evaluate on three domains: Rovers and AUV[2][1]; and a 'flat tyre' variant of TPP [2]. Summary results are in Figure 1a, showing a dramatic reduction in the time taken to solve problems. TPP does not have dead ends, so it serves to highlight the benefits of numeric regression. In Figures 1b and 1c, we test the benefits of numeric dead end generalisation (both X and Y use numeric regression), in the two domains with dead ends. There is again an improvement in time taken (Figure 1b). This is correlated with a reduction in the number of forbidden-state–action pairs (Figure 1c): the lower the number, the more 'general' the dead ends, and the faster the planner.

REFERENCES

[1] A. J. Coles, 'Opportunistic Branched Plans to Maximise Utility in the Presence of Resource Uncertainty', in *Proc. ECAI*, (2012).
[2] A. Gerevini, D. Long, P. Haslum, A. Saetti, and Y. Dimopoulos, 'Deterministic Planning in the Fifth International Planning Competition: PDDL3 and Experimental Evaluation of the Planners', *AIJ*, (2009).
[3] J. Hoffmann, 'The Metric-FF Planning System: Translating Ignoring Delete Lists to Numeric State Variables', *JAIR*, (2003).
[4] J. Hoffmann and B. Nebel, 'The FF planning system: Fast plan generation through heuristic search', *JAIR*, (2001).
[5] L.E. Marinescu and A.I. Coles, 'Heuristic guidance for forward-chaining planning with numeric uncertainty', in *Proc. ICAPS*, (2016).
[6] L.E. Marinescu and A.I. Coles, 'Non-deterministic planning with numeric uncertainty', Technical report, King's College London, (2016).
[7] C. Muise, V. Belle, and S.A. McIlraith, 'Computing contingent plans via fully observable non-deterministic planning', in *Proc. AAAI*, (2014).
[8] C. Muise, S.A. McIlraith, and J.C. Beck, 'Improved non-deterministic planning by exploiting state relevance', in *Proc. ICAPS*, (2012).

[2] We use a variant in which the AUV (Boaty McBoatface) must satisfy a prescribed number of goals, rather than considering oversubscription.

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

doi:10.3233/978-1-61499-672-9-1696

How Good Is Predictive Routing in the Online Version of the Braess Paradox?

László Z. Varga [1]

Abstract. With the online routing game model we can investigate the online routing problem, where each subsequent agent of the traffic flow may select different route based on real-time data. Recent investigations proved that if the agents of such system use the selfish shortest path search strategy, then in some situations sometimes the multi-agent system may be worse off with real-time data than without real-time data, even if anticipatory techniques are applied to predict the future state of the environment. We investigate the online Braess paradox, where each subsequent agent of the traffic flow may select different route, using anticipatory techniques.

1 Introduction

Things are connected to the internet in the hope that we can derive economic benefit from analysing the generated data streams in several application areas. The critical challenge is using the generated big amount of real-time data in a way that the overall distributed adaptive human-agent collective benefits from the instant availability of information. Recent mobile phone applications, like Google Maps and Waze, collect real-time and detailed information, including for example current travel time on road sections, in order to optimize travel routes. With this technical progress, the former equilibrium models [10] may not describe those traffic flows, where each subsequent agent of the traffic flow makes autonomous and rational decision based on the real-time traffic situation.

In this paper we claim that there is guaranteed benefit of real-time data in that version of the Braess paradox [1], where traffic participants exploit cooperation through intention-sharing-based prediction. We also claim that the maximum travel times after some transitional time are almost the same as in the classic Braess paradox.

2 Related Work

From game theory point of view [3], the routing problem is a network with source routing, where end users simultaneously choose a full route to their destination and the traffic is routed in a congestion sensitive manner [6]. A *flow distribution is optimal*, if it minimizes the cost of the total traffic flow over all possible flow distributions. A flow distribution is an *equilibrium flow distribution*, if none of the actors can change its traffic flow distribution among its possible paths to decrease its cost. It is proven [6] that every nonatomic routing problem *has at least one equilibrium flow distribution* and all equilibrium flow distributions have the same total cost. The *price of anarchy* is the ratio between the cost of an equilibrium flow distribution and the optimal flow distribution. If the cost functions are linear functions of the traffic flow, then the *price of anarchy has an upper bound* [6].

[1] ELTE University, Hungary, email: lzvarga@inf.elte.hu

In order to be able to investigate the effect of real-time data on the routing problem formally, the *online routing game* model was developed [7], and later refined [8]. In order to measure the effect of real-time data, the *benefit of online real-time data* concept was defined [7]. The agents are happy with real-time data, if the benefit value is below 1. If the benefit value is greater than 1, then it is in fact a ”price” like the price of anarchy. Three types (worst/average/best) of benefit of online real-time data are needed in case an equilibrium traffic distribution cannot be achieved.

The class of *simple naive* (SN) online routing games [7] is the model of current commercial vehicle navigation systems, which select the shortest route, with taking into account the real-time information available at decision time. It is proved [7], that in SN online routing games, equilibrium is not guaranteed, ”single flow intensification” is possible and the worst case benefit of online real-time data is not guaranteed to be below 1.

Anticipation [5] is essential to be able to behave proactively, and adapt to changing situations. Several design patterns were studied [11] to help the systematic design of self-organising emergent systems, that show anticipatory behaviour. The anticipatory vehicle routing simulation system of [2] uses delegate multi-agent systems to forecast future traffic conditions, based on the broadcast intention of the agents. This forecast information is then used by the agents to make better routing decisions than without forecast information.

In order to formally underpin the empirical conjectures, the class of *simple naive intention propagation* (SNIP) online routing games was defined and investigated in [8]. The findings are, that SNIP online routing games may also have ”single flow intensification” (although in a ”lighter” way), the worst case benefit of online real-time data may be above 1, and the traffic may fluctuate.

Because SNIP online routing games are very promising, and previous formal investigations found problems only with the simultaneous decision making, we advance the state of the art by proving properties of the SNIP online routing game on the Braess paradox, where there is a single incoming flow.

3 Online Routing Games

In order to get a detailed description of the online routing game model, the reader is referred to the open access paper [8].

The classic routing game [6] is a non-cooperative normal form game, where the decision making is on the flow level. The online routing game model resembles to the routing game model in the concepts of flow, cost, and graph edges.

Because the classic game types do not really fit the online routing problem, the approach of online mechanisms [4] is applied in the online routing game model. The sequential decision making of the agents of a flow is modelled with the sequences of time periods and

decisions. A T time unit is introduced in the online routing game model, because the costs of the edges depend on the number of agents entering the edge in a time unit.

The class of SNIP online routing games [8] abstracts away from the anticipatory design patterns, and models the mechanisms of these design patterns with the ability of the agents to spread their intentions truthfully, and the ability of the infrastructure to compute the future state of the system as a result of the intended actions of the agents. The agents select the route with the shortest travel time among the predicted travel times on the different routes (as predicted from the propagated intentions). The predicted travel times carry a kind of aggregated information about the previous decisions of the agents.

4 The Braess Network

We are investigating the Braess paradox in the road network G (shown in Figure 1). The total traffic flow is $r = (r_1)$, with only one flow on trip P_1 from source vertex v_0 to target vertex v_3. The cost function of the road network is c, with $c_{e_1}(x(t)) = l_1 + x(t) \div d$, $c_{e_2}(x(t)) = l_2$, $c_{e_3}(x(t)) = l_3$, $c_{e_4}(x(t)) = l_2$, $c_{e_5}(x(t)) = l_1 + x(t) \div d$ (where $x(t)$ is the traffic flow, i.e. total number of agents entering an edge from time period $t - T$ to time period t, and l_1, l_2, l_3, d are positive constants, and $l_2 - l_3 - l_1 > 0$).

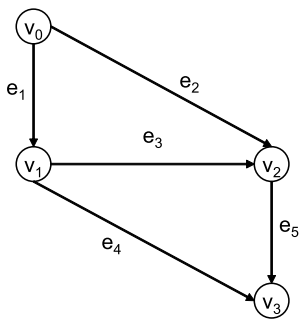


Figure 1. The investigated Braess network

In the equilibrium model if the traffic is low, then all the traffic goes on path p_2. The equilibrium cost is $c_{eq1} = 2 \times l_1 + 2 \times r_1 \div d + l_3$.

If the flow is above $(l_2 - l_3 - l_1) \times d \times 2 \div 3$ and below $(l_2 - l_3 - l_1) \times d$, then all the traffic goes on path p_2, and the anarchy has price. The equilibrium travel cost is again c_{eq1}.

If the flow is above $(l_2 - l_3 - l_1) \times d$ and below $(l_2 - l_3 - l_1) \times d \times 2$, then the traffic is divided among the paths p_1, p_2 and p_3, and the anarchy has price. The equilibrium travel cost is $c_{eq2} = 2 \times l_2 - l_3$.

If r_1 is above $(l_2 - l_3 - l_1) \times d \times 2$, then there is no traffic on the central path p_2, and half of the traffic goes on path p_1, and half of the traffic goes on path p_3. The equilibrium travel cost is $c_{eq3} = l_1 + r_1 \div 2 \div d + l_2$.

5 How Good is Predictive Routing?

Let us take the SNIP online routing game over the above Braess network, and call it B_1. This is the online version of the Braess paradox with anticipatory routing. The graph G and cost function c of B_1 is defined above in Section 4. The SNIP route selection strategy of the agents is described in Section 3. The total flow $r = r_1$ is from source vertex v_0 to target vertex v_3. The minimum following distance gap_e on all edges is the constant $g \geq 0$. The time unit T of B_1 can be any valid value, because it is not relevant in the theorems below.

Theorem 1 *The worst case benefit of online real-time data has a value less than or equal to* $1 + (1 \div d + g) \div (2 \times l_2 - l_3)$ *in the SNIP online routing game* B_1.

The proof and discussion of Theorem 1 is in [9].

Theorem 2 *The travel times in the SNIP online routing game* B_1 *after some time are at most the same as in the routing game model plus* $1 \div d$.

The proof is omitted here. The Braess network in [7] has the values $l_1 = 1, l_2 = 15, l_3 = 7.5$ and $d = 10$. With these values, c_{eq2} of the routing game model has the value 22.5 and the additional time in the SNIP online routing game model is 0.1. The additional time is due to the atomic nature of the flows.

Note that Theorem 2 seems to indicate that the worst case benefit of online real-time data is much better than the result from Theorem 1. For example if $r_1 > (l_2 - l_3 - l_1) \times d \times 2 + 2$, then Theorem 2 shows that the travel time c_{eq3} is around half of $c_{p2oracle} = 2 \times l_1 + l_3 + 2 \times r_1 \div d$ from Theorem 1. There is no contradiction. The value $1 + (1 \div d + g) \div (2 \times l_2 - l_3)$ of the worst case benefit of online real-time data may be achieved at the flow value $r_1 = (l_2 - l_3 - l_1) \times d$ or somewhat above.

6 Conclusion

The conjecture is that if simultaneous decision making is prevented, then intention-propagation-based prediction can limit the fluctuation in the multi-agent system. The results show that the coordination established by the intention-propagation-based prediction among the agents entering the network in a sequence is good enough to reduce the excessive swing of the system caused by the utilization of real-time information. There is convergence to the equilibrium, but the equilibrium may have to pay the price of anarchy.

REFERENCES

[1] D. Braess, 'Uber ein paradoxon der verkehrsplanung', *Unternehmensforschung*, **12**(1), 258–268, (1968).
[2] R. Claes, T. Holvoet, and D. Weyns, 'A decentralized approach for anticipatory vehicle routing using delegate multi-agent systems', *IEEE Transactions on Intelligent Transportation Systems*, **12**(2), 364–373, (2011).
[3] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, New York, NY, USA, 2007.
[4] David C. Parkes, *Online mechanisms*, 411–439. in [3].
[5] Robert Rosen, *Anticipatory Systems: Philosophical, Mathematical, and Methodological Foundations (IFSR International Series on Systems Science and Engineering)*, Pergamon Press, 1985.
[6] Tim Roughgarden, *Routing games*, 461–486. in [3].
[7] László Z. Varga, 'Online routing games and the benefit of online data', Proc. 8th Int. Workshop on Agents in Traffic and Transportation, at AAMAS 2014, May 5-6, 2014, Paris, France, 88–95, (2014).
[8] László Z. Varga, 'On intention-propagation-based prediction in autonomously self-adapting navigation', *Scalable Computing: Practice and Experience*, **16**(3), 221–232, (2015).
[9] László Z. Varga, 'Benefit of online real-time data in the braess paradox with anticipatory routing', Proc. 2nd Workshop on Distributed Adaptive Systems at 2016 IEEE International Conference on Autonomic Computing, July 19 2016, Wurzburg, Germany, accepted, to appear, (2016).
[10] John Glen Wardrop, 'Some theoretical aspects of road traffic research', *Proceedings of the Institution of Civil Engineers, Part II*, **1**(36), 352–378, (1952).
[11] Tom De Wolf and Tom Holvoet, 'Design patterns for decentralised coordination in self-organising emergent systems', in *Engineering Self-Organising Systems*, volume 4335 of *Lecture Notes in Computer Science*, 28–49, (2007).

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

doi:10.3233/978-1-61499-672-9-1698

Delete-Free Reachability Analysis for Temporal and Hierarchical Planning

Arthur Bit-Monnot[1] and **David E. Smith**[2] and **Minh Do**[2]

Abstract. Reachability analysis is a crucial part of the heuristic computation for many state of the art classical and temporal planners. In this paper, we study the difficulty that arises in assessing the reachability of actions in planning problems containing sets of interdependent actions, notably including problems with required concurrency as well as hierarchical planning problems. We show the limitation of state-of-the-art techniques and propose a new method suitable for both temporal and hierarchical planning problems. Our proposal is evaluated on FAPE, a constraint-based temporal planner.[3]

1 Introduction

Reachability analysis is crucial in computing heuristics guiding many classical and temporal planners. This is typically done by relaxing the action delete lists and constructing the reachability graph. This graph is then used as a basis to extract a relaxed plan, which serves as a non-admissible heuristic estimate of the actual plan reaching the goals from the current state.

Temporal planning poses some additional challenges for reachability analysis as heuristics should not only estimate the total cost but also the earliest time at which goals can be achieved. This can be accomplished on the reachability graph by labeling: (1) propositions with the minimum time of the effects that can achieve them; and (2) actions with the maximum time of the propositions they require as conditions. Since the reachability graph construction process progresses as time increases, when all *start* conditions are reachable, a given action a is eligible to be added to the graph. However, there is the additional problem that a's end conditions must also be reachable, although they do not need to be reachable until the end time of a. To see why this is a problem for the conventional way of building the reachability graph, consider the two actions in Figure 1: action B achieves the end condition for action A, but requires a start effect of A before it can start. Thus, B cannot start before A, but A cannot end until after B has ended. This means that A is not fully reachable until B is reachable, but B is not reachable unless A is reachable. Whether this turns out to be possible depends on whether B fits inside of A. In this example, the reasoning is simple enough, but more generally, B might be a complex chain of actions.

Planners such as POPF [2] address this problem by splitting durative actions into instantaneous start and end events, and forcing a time delay between the start and end events. In our example, the start of A would be reachable, leading to the start of B being reachable, which leads to the end of B being reachable, and finally the end of A being reachable. This approach therefore concludes that A is reachable. Unfortunately, the same conclusion is reached even when

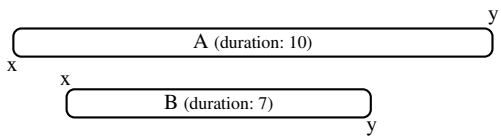


Figure 1: Two interdependent actions: A with a start effect x and an end condition y, and B with a start condition x and an end effect y.

B does not fit inside of A, because this "action-splitting" approach allows A to "stretch" beyond its actual duration.

In this paper, we present an approach to reachability analysis that addresses the above limitation and show how it can be beneficial for both generative and hierarchical temporal planners.

2 Planning Model & Relaxation

Temporal Model. We consider temporal planning problems similar to those of PDDL 2.1. For ease of presentation, we consider a discrete time model and restrict ourselves to actions with fixed duration and positive conditions.[4] An action has a set of conditions C_a and a set of effects E_a, all at arbitrary instants in the action envelope. A condition $c \in C_a$ has the form $\langle [t_c]\ f_c \rangle$, where f_c is a fluent and t_c is a positive delay from the start of the action to the moment f_c is required to be true. An effect $e \in E_a$ has the form $\langle [t_e]\ f_e \rangle$ (resp. $\langle [t_e]\ \neg f_e \rangle$ for negative effects), where t_e is the positive delay from the start of the action to the moment the fluent f_e becomes true (resp. false).

Relaxed Model. To estimate when each fact can be achieved, our reachability analysis utilizes *elementary actions*, which are artificial actions created from the original temporal actions defined in the domain description. Elementary actions contain: *(i)* only a single 'add' effect and *(ii)* the minimal set of conditions required to achieve that effect. More specifically for each positive effect $e = \langle [t_e]\ f_e \rangle$ of an action a, there will be an elementary action a_e with:

- a single effect $\langle [1]\ f_e \rangle$
- for each condition $\langle [t_c]\ f_c \rangle$ of a, a condition $\langle [t_c - t_e + 1]\ f_c \rangle$

Our relaxed model is composed of those delete-free elementary actions, each giving one possible way of achieving a given fluent. For any given elementary action, we say that a condition $c = \langle [t_c]\ f_c \rangle$ is a *before-condition* (resp. an *after-condition*) if $t_c \leq 0$ (resp. $t_c > 0$). An after-condition represents a condition that is required when or after the effect of the elementary action becomes active (e.g. y is an after-condition of the action A of Figure 1). In a reachability graph, such conditions would be represented by a negative edge. Those after-conditions are necessary for the presence of interdependencies such as the one shown in Figure 1 [3].

[1] LAAS-CNRS, Université de Toulouse, email: arthur.bit-monnot@laas.fr
[2] NASA Ames Research Center, email: {david.smith, minh.do}@nasa.gov
[3] A long version of this paper was presented at the HSDIP workshop [1].

[4] Extensions to more general models are discussed in the long version of this paper [1].

3 Reachability Analysis with After-conditions

In POPF, the splitting mechanism used for reachability analysis results in ignoring all after-conditions of durative actions. In order to avoid this additional relaxation, our method for reachability analysis is based on repeatedly alternating two steps: *(i)* we optimistically propagate achievements times while ignoring after-conditions; then *(ii)* we enforce all after-conditions. More specifically:

1. As a preliminary, we select a set of symbols that are assumed reachable at time 0. All fluents of the initial state are obviously part of this set. They are optimistically complemented with all elementary actions that have no before-conditions.
2. We then iteratively extend the set of assumed reachable nodes with: *(i)* all fluents that have an assumed reachable achiever and *(ii)* all actions whose every before-condition is assumed reachable. Each reachable symbol is associated with an earliest appearance time satisfying: *(i)* the minimal delays between an action and its before-conditions and *(ii)* the minimal delay between a fluent and its first achiever. This is done by a Dijkstra-like procedure that processes the nodes by increasing their earliest appearance times.
3. Our optimistic assumptions are then revised *recursively* by incorporating the ignored after-conditions. Specifically: *(i)* any elementary action with an after-condition on an unreachable fluent is removed from the model, *(ii)* if a removed action a is the only achiever of a fluent f, f is removed together with any action depending on it, *(iii)* the minimal delays between an action and its after-conditions are enforced by increasing the action's earliest appearance time as much as necessary.
4. If any action was updated in the previous step, we go back to step (2) and restart the propagation of earliest appearances from the updated nodes. Otherwise, analysis finishes with a set of reachable actions and fluents, each associated with an earliest appearance time.

Because earliest appearances could be endlessly increased towards infinity, we complement the last step with a detection of unreachable nodes. A group of nodes $\mathcal{N}$ is unreachable if for any node $n \in \mathcal{N}$, there is a delay of at least d_{max} between the earliest appearance of any node $n' \notin \mathcal{N}$ and n, d_{max} being the highest delay between any condition and its action or any action and its effect. The intuition is that nodes of this group are delaying each other due to unachievable interdependencies [1].

4 Extension to Hiearchical Planning

While automated reachability analysis is widespread in generative planners, hierarchical planners still rely on manual annotation of methods for this purpose. Here we propose a translation of hierarchical actions that exposes hierarchical features as additional conditions and effects for the purpose of reachability analysis.

We associate each hierarchical action a to a task symbol τ_a and a set of subtasks S_a. The intuition is that a achieves the task τ_a and requires all its subtasks to be achieved by other actions. For a plan π to be a solution it is required that:

- all initial tasks and all subtasks have been *achieved*. A task τ spanning a duration $[st_\tau, et_\tau]$ is said to be achieved if there is an action a_τ in the plan that: *(i)* achieves the task τ; *(ii)* starts at st_τ; and *(iii)* ends at et_τ.
- all actions in π achieve some task. This simulates HTN planners, in which all actions are inserted to achieve a pending task.

To allow reasoning on those additional requirements, we transform a hierarchical action a, with task τ_a, into a 'flat' action a_{flat} with:

- all conditions and effects of a,
- one start condition $\langle [0]\, required(\tau_a) \rangle$,
- one start effect $\langle [0]\, started(\tau_a) \rangle$ and one end effect $\langle [d_a]\, ended(\tau_a) \rangle$, where d_a is the duration of a,
- for each subtask $\langle [d_1, d_2]\, \tau \rangle$ of a:
  - two conditions $\langle [d_1]\, started(\tau) \rangle$ and $\langle [d_2]\, ended(\tau) \rangle$,
  - one effect $\langle [d_1]\, required(\tau) \rangle$.

Actions resulting from this compilation step encompass both causal and hierarchical features of the domain and can be split into elementary actions for reachability analysis techniques described earlier. This transformation usually exposes many interdependencies as each action both enables and requires the presence of its subactions.

5 Experiments & Conclusion

Our technique has been implemented in FAPE [4], a constraint-based temporal planner for the ANML language supporting both hierarchical and generative planning. Reachability analysis is used to *(i)* prune the search space by detecting dead-ends; and *(ii)* disregard resolvers involving unreachable actions.

Our method is tested with different configurations, R_∞ being the original method and R_5 and R_1 are variants in which the number of iterations are limited to 5 and 1 respectively. R^+ is the configuration where all after-conditions are ignored, thus producing the same result as the reachability analysis of POPF. $\varnothing$ denotes the configuration in which no reachability analysis is performed. Evaluation was done on various temporal domain with and without hierarchies.

	R_∞	R_5	R_1	R^+	$\varnothing$
(IPC-8) satellite (20)	14	14	14	14	**15**
(IPC-5) rovers (40)	**25**	**25**	**25**	**25**	**25**
(IPC-2) logistics (28)	**8**	**8**	**8**	**8**	**8**
(IPC-8) satellite-hier (20)	**17**	**17**	**17**	**17**	16
(IPC-5) rovers-hier (40)	**22**	**22**	**22**	**22**	**22**
(IPC-8) tms-hier (20)	**7**	**7**	**7**	**7**	**7**
(IPC-2) logistics-hier (28)	**28**	**28**	**28**	6	9
(LAAS) handover-hier (20)	**16**	**16**	**16**	7	7
(IPC-8) hiking-hier (20)	**20**	17	16	15	17
(LAAS) docks-hier (18)	**17**	13	12	7	7
Total (254)	**174**	167	165	128	133

Table 1: Number of solved tasks for various domains with a 30 minutes timeout. The best result is shown in bold. The number of problem instances is given in parenthesis.

As shown in Table 1, our method results in significant performance gain on hierarchical problems. This is because those problems feature many examples of interdependent actions for which our method is especially usefull. On temporally simple problems (here non-hierarchical ones), our method is equivalent to the reachability analysis of POPF and does not result in any runtime penalty. Note that the use of reachability analysis is here limited to dead-end detection. While this proves extremely useful on a wide variety of problems, one could also contemplate using it as a base for heuristic extraction.

REFERENCES

[1] Arthur Bit-Monnot, David E. Smith, and Minh Do, 'Delete-free Reachability Analysis for Temporal and Hierarchical Planning', in *Heuristics and Search for Domain-independent Planning*, pp. 93–102, (2016).
[2] Amanda Coles, Andrew Coles, Maria Fox, and Derek Long, 'Forward-Chaining Partial-Order Planning', in *ICAPS*, pp. 42–49, (2010).
[3] Martin C. Cooper, Frederic Maris, and Pierre Régnier, 'Managing Temporal Cycles in Planning Problems Requiring Concurrency', *Computational Intelligence*, **29**, 111–128, (2013).
[4] Filip Dvorak, Roman Bartak, Arthur Bit-Monnot, Felix Ingrand, and Malik Ghallab, 'Planning and Acting with Temporal and Hierarchical Decomposition Models', in *ICTAI*, pp. 115–121, (2014).

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

doi:10.3233/978-1-61499-672-9-1700

Intention Selection with Deadlines

Yuan Yao[1] and **Brian Logan**[1] and **John Thangarajah**[2]

1 INTRODUCTION

In BDI agent programming, an *intention* is the combined plan steps an agent commits to in order to achieve a goal. One of the key features of the BDI approach is the ability of an agent to pursue multiple goals concurrently, by interleaving the steps of multiple intentions. Choosing the next step to progress (execute) from these concurrent intentions is critical, as the wrong choice can result in failure to achieve one or more goals. Conversely, appropriate scheduling of the steps in intentions can maximise the number of goals achieved by the agent. Deciding which intention to progress next becomes more challenging in settings where goals must be achieved before a deadline. An interleaving of steps in the agent's intentions that avoids conflicts may still result in failure to achieve a goal by its deadline.

There has been relatively little work on intention selection with deadlines. One recent exception is AgentSpeak(RT). AgentSpeak(RT) [3, 2] is a real-time agent programming language based on AgentSpeak(L) in which top-level goals may have deadlines and priorities. Given the estimated execution time of plans, AgentSpeak(RT) schedules intentions so as to achieve a priority-maximal set of intentions by their deadlines with a specified level of confidence. However AgentSpeak(RT) avoids conflicts by scheduling potentially conflicting intentions in FIFO fashion, which can make it more difficult for an agent to achieve its goals by their deadlines.

In this paper, we present S_R, a novel approach to intention selection with deadlines. S_R extends the stochastic scheduling approach of Yao et al. [5, 6, 4] in two ways. First, goals may have both a preferred achievement time (the time by which the goal should ideally be achieved) and a deadline (the time by which the goal must be achieved). Second, S_R supports the concurrent execution of durative actions in different intentions. S_R schedules intentions so as to maximise the number of goals achieved and minimise tardiness (i.e, the difference between the time a goal is achieved and its preferred achievement time). We evaluate the performance of S_R and compare it to that of AgentSpeak(RT) in a simple blocks world domain. Our results suggest S_R outperforms AgentSpeak(RT) in this scenario.

2 INTENTIONS WITH DEADLINES

We extend the notion of goal found in BDI-based languages by introducing two temporal constructs: a preferred achievement time (soft deadline) p, and a deadline (hard deadline) d, where $p \leq d \leq \infty$. The *preferred achievement time* specifies the time by which a goal should ideally be achieved; however achieving the goal after the preferred achievement time still has value. The *deadline* specifies the time by which a goal must be achieved; achieving the goal after the deadline has no value. Goals which are not achieved by their (hard) deadline are dropped. If a top-level goal with preferred achievement time p is achieved at time $t \leq d$, we define the *tardiness* in achieving the goal as $max(0, t - p)$.

Each action a an agent may perform also has an *estimated execution time*, $e(a, \Phi)$, which is a function of the agent's beliefs Φ. The estimated execution time may be specified by the developer, e.g., the expected execution time of a 'move' action may be based on the distance to be moved, or learnt from experience, e.g., the agent may learn that traversing a hallway during a busy period takes longer than in the middle of the night. In general, the actual time it takes an action to execute, t, will differ from $e(a, \Phi)$. We define the estimation error in the execution time of an action a by $\frac{|e(a,\Phi)-t|}{e(a,\Phi)}$.

We assume that the agent executes its intentions in parallel. Progressing intentions in parallel allows the agent to achieve the maximum number of goals by their deadlines. (Executing intentions in parallel is often desirable for other reasons, e.g., to ensure more predictable response times for users or other agents.) Since external actions will, in general, require more time than a single deliberation cycle to execute, this means that multiple actions will typically be executing concurrently.

3 INTENTION SELECTION

In this section, we present our approach to intention selection, S_R. S_R attempts to find an interleaving of steps in the agent's intentions which (a) maximises the number of goals achieved (by their hard deadline), and (b), where the number of goals achieved is the same, minimises the total tardiness (i.e, the difference between the time goals are achieved and their preferred achievement time).

We use *goal-plan trees* to represent the relations between goals, plans and actions, and to reason about the interactions between intentions. The root of a goal-plan tree is a top-level goal (goal-node), and its children are the plans that can be used to achieve the goal (plan-nodes). Plans may in turn contain subgoals (goal nodes), giving rise to a tree structure representing all possible ways an agent can achieve the top-level goal. As in [6, 4] goal-plan trees may contain actions in plans (action nodes). We assume that each top-level goal is associated with a preferred achievement time and deadline, and each action is associated with the expected execution time of the action.

The agent's current intentions are represented by a set of pairs, $I = \{(T_1, c_1), \ldots, (T_n, c_n)\}$, where each T_i is a goal-plan tree corresponding to a top-level goal of the agent, and c_i is the *current step* (primitive action or a subgoal) of T_i. The *next step* of a goal-plan tree is the step after the current step. For example, if the current step is a subgoal, advancing the current step involves choosing a plan for the subgoal, and setting the next step to be the first action of the plan. If the current step is a primitive action, the next step is the basic action or subgoal following that action in the same plan. An intention is *progressable* if its current step does not point to an action that is

[1] University of Nottingham, email: {yvy | bsl}@cs.nott.ac.uk
[2] RMIT University, email: john.thangarajah@rmit.edu.au

currently executing, and the next-step (step after the current step) of the intention is either an action whose precondition holds and its pre- and postcondition does not conflict with the pre- or postcondition of any currently executing action, or a sub-goal for which there is at least one applicable plan in the current state.

If there is at least one progressable intention at the current deliberation cycle, the S_R algorithm is invoked to determine which intention to progress (which current-step pointer to advance). S_R is based on Single-Player Monte-Carlo Tree Search (SP-MCTS) [1], a best-first search in which pseudorandom simulations are used to guide expansion of the search tree. Each node in the search tree represents a possible interleaving of steps from the goal-plan trees in I, and records the state of the agent's environment resulting from the execution of this interleaving, the estimated time at which that state is achieved, the current-step of each intention and the completion time for the goals achieved on the path to this node. Edges represent the selection of a plan for a subgoal or the execution of primitive action in a plan.

Starting from the current step in each intention and the current environment, S_R iteratively builds a search tree. Each iteration consists of four phases. In the *selection* phase, a leaf node, n_e, representing a non-terminal state $s(n_e)$ is selected for expansion using a modified version of Upper Confidence bounds applied to Trees (UCT) [1]. In the *expansion* phase, n_e is expanded by adding child nodes representing the environment state (and the new current step of each intention) resulting from executing each next step of a goal-plan tree that is executable in state $s(n_e)$. Each child node therefore corresponds to a different choice of which intention to progress at this cycle. In the *simulation* phase, a developer-specified number of simulations are performed from a randomly selected child node, n_s, to determine the maximum number of goals that can be achieved from n_s, v_g, and the total tardiness of the interleaving, v_t. Starting in the environment represented by n_s, a next step of a goal-plan tree that is executable in $s(n_s)$ is randomly selected, and the environment and the current step pointer updated. This process is repeated until a terminal state is reached in which no next steps can be executed or all top-level goals are achieved. The value of the simulation is taken to be the values of v_g and v_t in the terminal state. Finally, in the *back-propagation* phase, v_g and v_t are back-propagated from n_s to all nodes on the path to the root node n_0 to inform selection at the next iteration.

After a developer-specified number of iterations, the step that leads to the best child of the root node is returned for execution at this deliberation cycle. S_R selects the best step using $<_{g,t}$, i.e., first on the number of top-level goals achieved, and, where two or more steps result in the achievement of the same number of goals, it prefers steps that also minimise total tardiness.

4 EVALUATING S_R

To evaluate the ability of our approach to schedule intentions with deadlines, we compared the performance of S_R with AgentSpeak(RT) in a simple Blocks World scenario in which an agent must pick blocks and stack them to build towers. Each top-level goal involves picking up two blocks from different locations and building a tower at a specified location. Each top-level goal has a preferred achievement time and deadline, specifying the times by which the task should and must be completed. In the experiments reported below, we used an environment with 30 cells. Initially, 20 randomly selected cells contain a block (numbered from 1 to 20), and the remaining 10 cells (the locations of the towers) are empty. The actions of picking up a block and stacking a block require 100 time units. The time required by a move action is determined by the distance the agent has to move (moving from one cell to another requires 100 time units). For simplicity, we assume the blocks and tower positions are located on a straight line, and the number of blocks the agent can carry is unlimited. The Blocks World scenario is very simple, but representative of a large class of real-world problems in which an agent must collect and use resources in order to achieve goals.

In our experiment, we used 50 sets of 10 goals. Each goal involves collecting two specified blocks and stacking them in a specified empty cell (the block ids and tower positions are different for each goal). Three goals are given initially, and the remaining top-level goals are posted every 1650 time units. As the average time for achieving a top-level goals is 3300 time units, posting a new goal every 1650 time units means that, typically, the agent must achieve more than one goal in parallel. We varied the deadline d for each top-level goal from 7500 to 30000, and the preferred achievement time p was 2500 time units less than the deadline.

Table 1: Goals achieved and tardiness with decreasing deadline

		S_R		AS(RT)	
d	p	#Goals	Tardiness	#Goals	Tardiness
30000	27500	10	0	10	0
15000	12500	10	0	6.96	2386
10000	7500	10	2340	5.38	3195
7500	5000	9.34	3874	5.04	3220

The results are shown in Table 1. As can be seen, as the deadline decreases, the performance of AgentSpeak(RT) declines rapidly. In contrast, S_R was able to achieve at least 9.34 goals on average, though with tighter deadlines, tardiness increases significantly.

5 FUTURE WORK

S_R takes deadline and preferred achievement time of goals into account when choosing which intention to progress. However, while the choice of action at each deliberation cycle is based on the current state of the agent's environment at that cycle, the simulation of the possible outcomes of actions assumes a static environment. In future work we plan to investigate the incorporation of simple environment models to allow the prediction of likely environment changes during simulation, and evaluate S_R in dynamic environments.

REFERENCES

[1] Maarten P. D. Schadd, Mark H. M. Winands, Mandy J. W. Tak, and Jos W. H. M. Uiterwijk, 'Single-player Monte-Carlo tree search for SameGame', *Knowledge-Based Systems.*, **34**, 3–11, (2012).

[2] Konstantin Vikhorev, Natasha Alechina, Rafael Bordini, and Brian Logan, 'An operational semantics for AgentSpeak(RT) (preliminary report)', in *Proceedings of the Ninth International Workshop on Declarative Agent Languages and Technologies (DALT 2011)*, pp. 79–94, (May 2011).

[3] Konstantin Vikhorev, Natasha Alechina, and Brian Logan, 'Agent programming with priorities and deadlines', in *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pp. 397–404, (May 2011).

[4] Yuan Yao and Brian Logan, 'Action-level intention selection for BDI agents', in *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016)*, pp. 1227–1235, (May 2016).

[5] Yuan Yao, Brian Logan, and John Thangarajah, 'SP-MCTS-based intention scheduling for BDI agents', in *Proceedings of the 21st European Conference on Artificial Intelligence*, (August 2014).

[6] Yuan Yao, Brian Logan, and John Thangarajah, 'Robust execution of BDI agent programs by exploiting synergies between intentions', in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, pp. 2558–2564, (February 2016).

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

doi:10.3233/978-1-61499-672-9-1702

Cost-Optimal Algorithms for Planning with Procedural Control Knowledge

Vikas Shivashankar[1] and **Ron Alford**[2] and **Mark Roberts**[3] and **David W. Aha**[4]

1 Motivation and Background

Formalisms for automated planning (to represent and solve planning problems) broadly fall into either *domain-independent planning* or *domain-configurable planning*. Domain-independent planning formalisms, such as *classical planning* requires that the users only provide models of the base actions executable in the domain. In contrast, domain-configurable planning formalisms (e.g., *Hierarchical Task Network (HTN) planning* [1]) allow users to supplement action models with additional domain-specific knowledge structures that increases the expressivity and scalability of planning systems.

An impressive body of work exploring search heuristics has been developed for classical planning that has helped speed up generation of high-quality solutions. More specifically, search heuristics such as the relaxed planning graph heuristic [2], landmark generation algorithms [3, 5], and landmark-based heuristics [5, 4] dramatically improved optimal and anytime planning algorithms by guiding search towards (near-) optimal solutions to planning problems.

Yet relatively little effort has been devoted to develop analogous techniques to guide search towards high-quality solutions in domain-configurable planning systems. In lieu of such search heuristics, domain-configurable planners often require additional domain-specific knowledge to provide the necessary search guidance. This requirement not only imposes a significant burden on the user, but also sometimes leads to brittle or error-prone domain models.

In this paper, we address this gap by developing the ***H**ierarchically-**Op**timal **G**oal **D**ecomposition **P**lanner (*HOpGDP*)*, a hierarchical planning algorithm that uses admissible heuristic estimates to generate *hierarchically-optimal* plans (i.e., plans that are valid and optimal with respect to the given hierarchical knowledge). HOpGDP leverages recent work on a new hierarchical planning formalism called *Hierarchical Goal Network* (HGN) Planning [8, 6], which combines the hierarchical structure of HTN planning with the goal-based nature of classical planning.

In particular, our contributions are as follows:

- **Admissible Heuristic**: We present h_{HL}(**H**GN **L**andmark heuristic), a planning heuristic that extends landmark-based admissible classical planning heuristics to derive admissible cost estimates for HGN planning problems. To the best of our knowledge, h_{HL} is the first non-trivial admissible hierarchical planning heuristic.
- **Optimal Planning Algorithm**: We introduce HOpGDP, an A* search algorithm that uses h_{HL} to generate *hierarchically-optimal* plans.
- **Experimental Evaluation**: We describe an empirical study on three benchmark planning domains in which HOpGDP outperforms optimal classical planners due to its ability to exploit hierarchical knowledge. We also found that h_{HL} provides useful search guidance; despite substantial computational overhead, it compares favorably in terms of runtime and nodes explored to $\mathsf{HOpGDP}_{\mathsf{blind}}$, using the trivial heuristic $h = 0$.

For the full paper, the readers are referred to the online e-Print [7].

2 Preliminaries

An *HGN planning problem* is a triple $P = (D, s_0, gn_0)$, where D is an HGN domain, s_0 is the initial state, and $gn_0 = (T, \prec)$ is the initial goal network. An *HGN domain* is a pair $D = (D_c, M)$ where D_c is a classical planning domain and M is a set of HGN methods. D_c describes the models of the base actions, while the HGN methods M specifies hierarchical control knowledge the planner needs to respect when generating plans. Finally, a *goal network* is a partially ordered multiset of goals; this is analogous to the central data structure in HTN planning, the *task network* [1].

HGN Methods. An *HGN method* encodes knowledge on how to decompose goals. Each method m consists of a goal goal(m) that m decomposes, the conditions precond(m) under which it is applicable, and the goal network network(m)) that m decomposes into.

Solutions to HGN Planning Problems. The set of *solutions* for $P = (D, s_0, gn_0)$ is inductively defined as follows: (1) if gn_0 is empty, the empty plan is a solution. If not, assuming $g \in gn_0$ is a goal without predecessors, (2) if g is true in s_0, we can remove g from gn_0, or (3) if there is an action/method applicable in s_0 and relevant to g, we can apply it; actions progress the state while methods progress the goal network.

Let us denote $\mathcal{S}(P)$ as the set of solutions to an HGN planning problem P as allowed by the previous definition. Then we can define what it means for a solution π to be *hierarchically optimal* with respect to P as follows:

Definition 1 (Hierarchically Optimal Solutions)**.** *A solution $\pi^{h,*}$ is hierarchically optimal with respect to P if $\pi^{h,*} = argmin_{\pi \in \mathcal{S}(P)} \mathsf{cost}(\pi)$.*

3 The HOpGDP Algorithm and the h_{HL} Heuristic

HOpGDP takes as input an HGN planning problem $P = (D, s_0, gn_0)$. It does a standard A* search using the admissible HGN

[1] Knexus Research Corporation, National Harbor, MD, vikas.shivashankar@knexusresearch.com
[2] MITRE, McLean, VA, ralford@mitre.org
[3] NRC Postdoctoral Fellow, Naval Research Laboratory, Washington DC, mark.roberts.ctr@nrl.navy.mil
[4] Naval Research Laboratory, Washington DC, david.aha@nrl.navy.mil

Full paper: https://arxiv.org/abs/1607.01729

heuristic h_{HL} (described later in this section) to compute a hierarchically optimal solution to the problem; it either returns a plan if it finds one, or failure if the problem is unsolvable. In particular, starting from the search node (s_0, gn_0), HOpGDP (1) generates successors according to the solution definition in Section 2, and (2) evaluates them using using h_{HL}; it repeats this cycle until it either (a) finds a search node with an empty goal network, at which point it can terminate and return the corresponding plan, or (b) it exhausts the entire search space, in which case it returns failure. HOpGDP thus explores an identical search space as previous HGN planners like GDP [8], but unlike them, it explores the space in a best-first manner, allowing it to explicitly optimize for total plan cost.

The h_{HL} Heuristic. As mentioned previously, HOpGDP uses h_{HL} to compute the h-values (and thus, the f-values) of search nodes. The main insight behind the construction of h_{HL} is as follows: *given a problem $P = (D, s, gn)$, every goal in gn must be achieved, and in the order specified in gn. In other words, the elements of gn can be thought of as landmarks enforced by the hierarchical knowledge, with the partial order serving as landmark orderings.* So, one way to develop admissible HGN heuristics is to use goals in gn as starting points for generating an expanded set of landmarks, and then invoke off-the-shelf landmark-based classical planning heuristics on these landmarks to compute admissible estimates.

Concretely, we construct h_{HL} as follows (details in [7]):

1. We define a relaxation of HGN planning that ignores the provided methods and allows unrestricted action chaining as in classical planning, which expands the set of allowed solutions,
2. We extend landmark generation algorithms for classical planning problems to compute sound landmark graphs for the relaxed HGN planning problems, which in turn are sound with respect to the original HGN planning problems as well, and finally
3. We use admissible classical planning heuristics like h_L [4] on these landmark graphs to compute admissible cost estimates for HGN planning problems.

Based on the admissibility of h_L we can prove that h_{HL} generates admissible cost estimates for HGN planning problems:

Theorem 2 (Admissibility of h_{HL})**.** *Given an HGN planning domain D, a search node (s, gn, π) and its cost-optimal solution $\pi^{*,HGN}_{s,gn}$, $h_{HL}(s, gn, \pi) \leq \pi^{*,HGN}_{s,gn}$.*

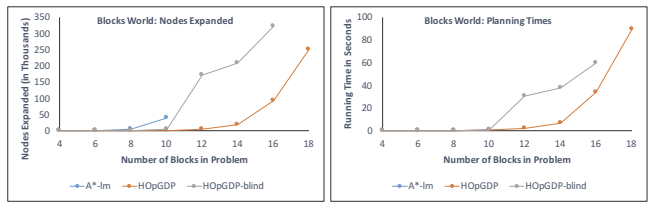
4 Experimental Study

We implemented HOpGDP within the Fast-Downward codebase, and extended LAMA's landmark generation code to develop h_{HL}, our HGN planning heuristic. We tested two hypotheses in our study:

H1. HOpGDP*'s ability to exploit hierarchical planning knowledge enables it to outperform state-of-the-art optimal classical planners.* To test this, we compared the performances of HOpGDP with A*-h_L [4], the optimal classical planner whose heuristic we extended to develop h_{HL}.

H2. *The heuristic used by* HOpGDP*, h_{HL}, provides useful search guidance.* To test this, we compared the performances of HOpGDP with $\text{HOpGDP}_{\text{blind}}$, which is identical to HOpGDP except that it uses the trivial heuristic estimate of $h = 0$.

We evaluated HOpGDP, $\text{HOpGDP}_{\text{blind}}$, and A*-$h_L$ on three well-known planning benchmarks, Logistics, Blocks World and Depots. Due to space constraints, we show results only for Blocks-World; the reader can find the full experimental study in [7].



(a) (b)

Figure 1: Node expansions and running times in Blocks-World. Each data point is the average over 25 randomly generated problems. Data points where all the problems are not solved were discarded.

Figure 1, which shows the results for Blocks-World, shows that A*-h_L could only solve problems up to size 10, while $\text{HOpGDP}_{\text{blind}}$ and HOpGDP could solve problems up to sizes 16 and 18 respectively within the 30 minute timelimit, thus supporting Hypothesis **H1**. In terms of node expansions, Figure 1a indicates that the guidance provided by h_{HL} helped substantially; HOpGDP on average expanded 76% fewer nodes than $\text{HOpGDP}_{\text{blind}}$. This savings far outweighed the heuristic computation overhead (on average about 48% of the total running time), resulting in smaller overall planning times for HOpGDP as shown in Figure 1b, supporting Hypothesis **H2**.

To conclude, our experimental results demonstrate that HOpGDP outperforms optimal classical planners (due to its ability to exploit domain-specific planning knowledge) as well as optimal blind search HGN planners (due to the search guidance provided by h_{HL}).

In the future, we plan to explore extensions of HOpGDP to support *anytime-optimal* planning as well as temporal planning.

ACKNOWLEDGEMENTS

This work is sponsored in part by OSD ASD (R&E). The information in this paper does not necessarily reflect the position or policy of the sponsors, and no official endorsement should be inferred. Ron Alford performed part of this work under an ASEE postdoctoral fellowship at NRL. We also would like to thank the anonymous reviewers at ECAI 2016 and HSDIP 2016 for their insightful comments.

REFERENCES

[1] Kutluhan Erol, James Hendler, and Dana S. Nau, 'UMCP: A sound and complete procedure for hierarchical task-network planning', in *Proc. Int. Conf. on AI Planning Systems*, pp. 249–254, (June 1994).

[2] J. Hoffmann and Bernhard Nebel, 'The FF planning system', *Journal of Artificial Intelligence Research*, **14**, 253–302, (2001).

[3] Jörg Hoffmann, Julie Porteous, and Laura Sebastia, 'Ordered landmarks in planning', *J. Artif. Intell. Res. (JAIR)*, **22**, 215–278, (2004).

[4] Erez Karpas and Carmel Domshlak, 'Cost-optimal planning with landmarks', in *International Joint Conference on Artificial Intelligence*, pp. 1728–1733, (2009).

[5] Silvia Richter and Matthias Westphal, 'The LAMA planner: Guiding cost-based anytime planning with landmarks', *J. Artif. Intell. Res. (JAIR)*, **39**, 127–177, (2010).

[6] Vikas Shivashankar, Ron Alford, Ugur Kuter, and Dana Nau, 'The GoDeL planning system: a more perfect union of domain-independent and hierarchical planning', in *International Joint Conference on Artificial Intelligence*, pp. 2380–2386. AAAI Press, (2013).

[7] Vikas Shivashankar, Ron Alford, Mark Roberts, and David W. Aha. Cost-optimal algorithms for planning with procedural control knowledge. `https://arxiv.org/abs/1607.01729`, 2016.

[8] Vikas Shivashankar, Ugur Kuter, Dana Nau, and Ron Alford, 'A hierarchical goal-based formalism and algorithm for single-agent planning', in *Proc. of the 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 2, pp. 981–988, (June 2012).

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

doi:10.3233/978-1-61499-672-9-1704

Adaptive Condorcet-Based Stopping Rules Can Be Efficient

Omer Reingold[1] and **Nina Narodytska**[2]

Abstract. A crowdsourcing project is usually comprised of many unit tasks known as Human Intelligence Tasks (HITs). As answers to each HIT varies between workers, each HIT is often contracted to more than one worker to obtain a reliable and consistent enough answer. When implementing a project, an important design decision is how to formulate HITs and how to aggregate workers' answers. These decisions have strong impact on the quality of results and cost of elicitation process. One way to design an efficient elicitation procedure is to use *adaptive stopping rules*, which allows terminating the elicitation process as soon as a high quality result is guaranteed.

Adaptively deciding how many times to issue a HIT is mostly well understood for the case of binary-answer HITs, thanks to the work of Abraham *et al.* [3, 2, 1]. In this line of work the authors focused on plurality-based stopping rules and provided their theoretical analysis. As a decision rule (when many alternatives are offered), it is well known that plurality may be inferior to other rules, such as the Condorcet method. We argue that for large number of possible answers, plurality-based stopping rules may also be terribly inefficient. In other words, one may need to elicit answers from many workers (at least linear in the number of answers) in order to get any reasonable approximation of the plurality answer. Somewhat surprisingly, we show that Condorset-based stopping rules may be much more efficient (with the number of workers to find the approximate Condorset winner depending only logarithmically on the number of answers). Moreover, in an important case of restricted domains, namely single-peaked domains, we show that the stopping time to find an approximate Condorcet-based winner does not depend on the number of answers at all. Overall, our results suggest that both crowdsourcing platform developers and HIT designers, should consider Condorcet-based adaptive stopping rules as a useful tool in their toolboxes.

Introduction Crowdsourcing is a simple and efficient way to gather information from a large population of people. A typical crowdsourcing platform is a marketplace for two types of users: requesters and workers. The central notion at the marketplace is a human intelligence task (HIT). HIT is a unit task that can be allocated to a worker. The requester formulates their question as a HIT and assigns a monetary reward for *answering* this question. Each time a worker answers a HIT, they receive the reward and the requester obtains their answer.

Some of the most popular crowdsourcing tasks are data classification and data ranking [9, 14], e.g., labeling an image as child-appropriate or adult-only, tagging tweets with relevant sentiments [7], ranking search results based on relevance, etc. Such tasks are used, in particular, to provide input to analytics algorithms, including machine learning, natural language processing and information retrieval algorithms. For example, image labeling is an important preprocessing step for training deep neural networks, which typically requires a training set of 1-2 million labeled images. Obtaining such a set is a labor demanding task. Thanks to crowdsourcing platforms, like Amazon Mechanical Turk or CrowdFlower, large volumes of data can be processed efficiently and with reasonable budget.

Being a gateway to a large population of workers, the crowdsourcing framework is unable to evaluate the qualification of workers or to monitor the quality of their answers. Therefore, ensuring high-quality answers is one of the first problems that any requester faces in dealing with such frameworks. One natural way to improve the quality of results is to obtain answers from many different workers. Implementing such redundancy in practice can be costly as the requester does not know in advance how many workers to assign to the same task. In this context, Abraham *et al.* [3, 2, 1] studied *adaptive* stopping rules that at each stage, given previous workers' answers, decide whether to stop or to ask for an additional answer. Stopping as early as possible saves costs and is therefore highly motivated. They focus on a single HIT stopping rule, where each HIT is a binary question and they perform a thorough theoretical investigation of the proposed stopping rule.

In this work we consider non-binary or multi-answer HITs. Non-binary tasks naturally occur in data classification and data ranking problems. For example, a tweet tagging task requires the worker to choose the most relevant emotional labels from a list of labels. Another example is to choose the most appropriate label for a picture. Next we overview our contributions.

Approximate-Condorcet winner. The plurality-based stopping rule proposed by Abraham *et al.* [3] has several disadvantages that can be avoided when additional information about the structure of the preferences is available. For example, if the probability of the plurality answer is small then the rule is inefficient, even if the plurality answer is twice as likely as any other answer. To overcome these issues, we propose to natural extension of the plurality-based stopping rule to a Condorcet-based stopping rule. We use the notion of approximate-Condorcet winner which is defined as an alternative that wins or effectively ties with any other alternative. We say that an alternative x effectively ties with another alternative x' if the probability that a random worker prefers x to x' is at least $\frac{1}{2} - \mu$, where μ is a parameter. Intuitively, if the probability that one alternative is preferred to another is within a μ circle of $\frac{1}{2}$, we decide that these alternatives are equally preferred by the workers. We introduce this notion with dual motivation. First, it allows us to determine a socially acceptable answer even in cases where an exact Condorcet winner does not exist. Second, it decreases the expected stopping time significantly. To be able to find an approximate Condorcet winner and perform

[1] Samsung Research America, email: o.reingold@samsung.com
[2] Samsung Research America, email: nina.n@samsung.com

the search efficiently, we make two technical extensions of existing stopping rules.

Extensions of stopping rules. Given a k-answer HIT, the Condorcet winner determination procedure finds $O(k^2)$ outcomes of pairwise comparisons between alternatives. Each of these comparisons can be seen as an independent competition between two corresponding alternatives. Hence, we propose an extension of plurality-based stopping rule [3] that can deal with parallel processes. We show that the expected stopping time depends only logarithmically on the number of parallel processes. The second extension helps us to deal with cases when several alternatives are very close to be Condorcet winners in workers' preferences. These alternatives might effectively tie in several pairwise competitions. Applying the existing stopping rule [3] to establish the winner in these competitions is costly. In this case, we augment our stopping rule with the ability to output the 'too close to call' answer. Such enchantment achieves a significant improvement in the expected stopping time and allows us to find an approximate Condorcet winner efficiently.

Single-peaked preferences. In the general case, when HIT's answers are unrelated, we cannot reduce the elicitation cost per worker. However, for multi-answer HITs where possible answers are related we can reduce the elicitation cost by requiring workers to provide only their top choice, as in the case of plurality-based stopping rules. In particular, we show how to reduce the stopping time needed to identify approximate Condorcet winners in single-peaked domains to be *completely independent* of the number of answers and only depend on the approximation parameter.

Designing individual HITs within a larger project involves many considerations, e.g. the attention span of a typical worker, budget constraints, workers' qualification. We suggest that the cost of reaching a high-quality answer is another important consideration, and we offer our quantitative analysis as a design tool for task requesters to take into account. In particular, our results reveal that expected stopping time to find (an approximate) Condorcet winner is not very expensive and, in case of single-peaked domains, no more expensive than finding the plurality winner (for binary HITs). In other words, our results show that, especially in case of single-peaked domains, the requester can find a Condorcet winner without the significant budget overhead.

Related work The related work can be divided into two main categories. The first category is on adaptive stopping rules by Abraham *et al.* [3, 2, 1]. The main goal of this line of work is to reduce the total cost of crowdsourcing procedure by using adaptive stopping rules. In particular, the proposed algorithm decides whether to stop or to ask one more worker at each round. The crowdsourcing platform stops if it can provide a high-quality answer with high probability. The main contribution of our work in to extend these plurality-based rules to Condorcet-based rules efficiently.

The second related work category consists of finding sampling bounds and aggregation procedures in computational social choice [6, 10, 4, 12, 11]. The main difference from our work from the work is that we focus on adaptive stopping rules for the elicitation procedure. Hence, the main advantage of our work that we can stop dynamically, often before reaching the requesters static bound on the number of HIT assignments. For example, adaptive stopping rules are particularly useful if the crowd significantly prefers x over other alternatives and can efficiently produce the correct answer with a high probability. Moreover, in most of the work, a different access model to users' preferences is considered, based on comparison queries (the elicitor can ask an agent which of two alternatives they prefer), while we assume that voters can provide their top choices. Hence, we will briefly overview related work in this line of research. Kruger *et al.* [10] considered several aggregation procedures, including the plurality rule and the bias-correcting method. The authors provided axiomatic analysis of these aggregation methods and conducted an experimental study on Amazon Mechanical Turk. Procaccia *et al.* [13] present a new aggregation method that assumes that workers' answers are noisy. Lee *et al.* [12], introduced the notion of ϵ-approximate winners which is closely related to our approximation notion. They show how to elicit these approximations in time $O(klog(k))$, where k is the number of alternatives. These methods can return an approximate winner for ϵ-Borda and ϵ-Condorcet with high probability. Our notion of approximate Condorcet winner is different from ϵ-Condorcet [12] as we require that an alternative to win or tie in all pairwise competitions rather than a fraction of them. Later these results were extended to a larger class of voting rules [11]. Conitzer [5] focuses on elicitation of voters preferences in single-peaked domains. He showed preferences can be elicited using only a linear number of comparison queries if the underlying order with respect to which preferences are single-peaked is known. Goel and Lee [8] extended results of Conitzer [5] to the case when the single-peaked axis is unknown assuming that candidates and participants coincide.

REFERENCES

[1] Ittai Abraham, Omar Alonso, Vasilis Kandylas, Rajesh Patel, Steven Shelford, and Aleksandrs Slivkins, 'How Many Workers to Ask? Adaptive Exploration for Collecting High Quality Labels', Technical report, http://arxiv.org/abs/1411.0149, (2014).

[2] Ittai Abraham, Omar Alonso, Vasilis Kandylas, Rajesh Patel, Steven Shelford, and Alex Slivkins, 'Using Worker Quality Scores to Improve Stopping Rules', in *Second AAAI Conference on Human Computation and Crowdsourcing*, (2014).

[3] Ittai Abraham, Omar Alonso, Vasilis Kandylas, and Aleksandrs Slivkins, 'Adaptive Crowdsourcing Algorithms for the Bandit Survey Problem', in *Conference on Learning Theory*, pp. 882–910, (2013).

[4] Ioannis Caragiannis, Ariel D Procaccia, and Nisarg Shah, 'When do noisy votes reveal the truth?', in *Proceedings of the fourteenth ACM conference on Electronic commerce*, pp. 143–160. ACM, (2013).

[5] Vincent Conitzer, 'Eliciting single-peaked preferences using comparison queries', *JAIR*, **35**(1), 161–191, (2009).

[6] Ulle Endriss and Raquel Fernández, 'Collective annotation of linguistic resources: Basic principles and a formal model', in *BNAIC 2013: Proceedings of the 25th Benelux Conference on Artificial Intelligence, Delft, The Netherlands, November 7-8, 2013*, (2013).

[7] Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze, 'Annotating named entities in Twitter data with crowdsourcing', 80–88, (June 2010).

[8] Ashish Goel and David Lee, 'Triadic Consensus', in *Internet and Network Economics*, 434–447, Springer, (2012).

[9] Jonathan Kraus, 'Lessons Learned from Large Scale Crowdsourced Data Collection for ILSVRC', Technical report, (2015).

[10] Justin Kruger, Ulle Endriss, Raquel Fernández, and Ciyang Qing, 'Axiomatic analysis of aggregation methods for collective annotation', in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 1185–1192, (2014).

[11] David Timothy Lee, 'Efficient, Private, and eps-Strategyproof Elicitation of Tournament Voting Rules', in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, (2015).

[12] David Timothy Lee, Ashish Goel, Tanja Aitamurto, and Helene Landemore, 'Crowdsourcing for participatory democracies: Efficient elicitation of social choice functions', in *Second AAAI Conference on Human Computation and Crowdsourcing*, (2014).

[13] Ariel D Procaccia, Nisarg Shah, and Yair Zick, 'Voting rules as error-correcting codes', *Artificial Intelligence*, **231**, 1–16, (2016).

[14] Luis von Ahn and Laura Dabbish, 'Labeling images with a computer game', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems(CHI'04)*, pp. 319–326. ACM Press, (2004).

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

doi:10.3233/978-1-61499-672-9-1706

Landmark-Based Plan Recognition

Ramon Fraga Pereira and **Felipe Meneguzzi** [1]

1 Introduction

As more computer systems require reasoning about what agents (both human and artificial) other than themselves are doing, the ability to accurately and efficiently recognize goals and plans from agent behavior becomes increasingly important. Plan recognition is the task of recognizing goals and plans based on often incomplete observations that include actions executed by agents and properties of agent behavior in an environment [10]. Accurate plan recognition is important to monitor and anticipate agent behavior, such as in crime detection and prevention, monitoring activities, and elderly-care. Most plan recognition approaches [3, 1] employ plan libraries (*i.e*, a library with all plans for achieving a set of goals) to represent agent behavior, resulting in approaches to recognize plans that are analogous to language parsing. Recent work use planning domain definitions (domain theories) to represent potential agent behavior, bringing plan recognition closer to planning algorithms [8, 7, 6, 2].

In this paper, we develop a plan recognition approach that relies on planning landmarks [4] to filter candidate goals and plans from observations. Landmarks are properties (or actions) that every plan must satisfy (or execute) at some point in every plan execution to achieve a goal. In this way, we use this filtering algorithm in two settings. First, we build a landmark-based plan recognition heuristic that analyzes the amount of achieved landmarks to estimate the percentage of completion of each filtered candidate goal. Second, we show that the filter we develop can also be applied to other planning-based plan recognition approaches, such as the approach from Ramírez and Geffner [8]. We evaluate our approach empirically against the current state-of-the-art [8] using their own datasets [8, 7], and show that our approach has multiple advantages over existing approaches: it is more accurate than the state-of-the-art; it is substantially faster on its own; and it can also be used to speed up existing approaches. A complete discussion of our plan recognition approach and experiments is provided in the full paper[2].

2 Filtering Candidate Goals from Landmarks in Observations

Key to our approach to plan recognition is the ability to filter candidate goals based on the evidence of fact landmarks and partitioned facts in preconditions and effects of observed actions in a plan execution. Our filtering process analyzes fact landmarks inferred from observed actions, and selects goals from a set of candidate goals with the highest number of observed landmarks having been achieved. We take as input a plan recognition problem T_{PR}, which is composed of a planning domain definition Ξ, an initial state $\mathcal{I}$, a set of candidate goals $\mathcal{G}$, a set of observed actions O, and a filtering threshold θ. The threshold gives us flexibility when dealing with incomplete observations and sub-optimal plans, which, when $\theta = 0$, may cause some potential goals to be filtered out before we get additional observations. Our algorithm iterates over the set of candidate goals $\mathcal{G}$, and, for each goal G in $\mathcal{G}$, it extracts and classifies fact landmarks and partitions for G from the initial state $\mathcal{I}$. We then check whether the observed actions O contain fact landmarks or partitioned facts of G in either their preconditions or effects. As we deal with partial observations in a plan execution some executed actions may be missing from the observation, thus whenever we identify a fact landmark, we also infer that its predecessors have been achieved. Given the number of achieved fact landmarks of G, we estimate the percentage of fact landmarks that the observed actions O have achieved according to the ratio between the amount of achieved fact landmarks and the total amount of landmarks. Finally, we return the goals from $\mathcal{G}$ with the highest percentage of achieved landmarks within threshold θ.

3 Heuristic Plan Recognition using Landmarks

We now develop a landmark-based heuristic method that estimates the goal completion of every goal in the set of filtered goals. This estimate represents the percentage of sub-goals (atomic facts that are part of a conjunction of facts) in a goal that have been accomplished based on the evidence of achieved fact landmarks in observations. Our heuristic method estimates the percentage of completion towards a goal by using the set of achieved fact landmarks provided by the filtering process. We aggregate the percentage of completion of each sub-goal into an overall percentage of completion for all facts in a candidate goal. This heuristic, denoted as h_{prl}, is computed by Equation 1, where $\mathcal{AL}_g$ is the number of achieved landmarks from observations of every sub-goal g of the candidate goal G, and $\mathcal{L}_g$ represents the number of necessary landmarks to achieve every sub-goal g of G. Thus, heuristic $h_{prl}(G)$ estimates the completion of a goal G by calculating the ratio between the sum of the percentage of completion for every sub-goal $g \in G$, *i.e*, $\sum_{g \in G} \frac{|\mathcal{AL}_g|}{|\mathcal{L}_g|}$, and the number of sub-goals in G.

$$h_{prl}(G) = \left(\frac{\sum_{g \in G} \frac{|\mathcal{AL}_g|}{|\mathcal{L}_g|}}{\mid G \mid} \right) \quad (1)$$

4 Landmark-based Plan Recognition

Our plan recognition approach is detailed in Algorithm 1, which takes as input a plan recognition problem T_{PR}, and works in two stages. First, this algorithm filters candidate goals using the filtering process, which returns the candidate goals with the highest percentage of achieved landmarks within a given threshold θ. Second, from

[1] Pontifical Catholic University of Rio Grande do Sul (PUCRS), Brazil. Contact: ramon.pereira@acad.pucrs.br and felipe.meneguzzi@pucrs.br
[2] https://arxiv.org/pdf/1604.01277v2.pdf

					Landmark-based Plan Recognition		R&G		Filter + R&G	
Domain	$\|\mathcal{G}\|$	$\|\mathcal{L}\|$	**%Obs**	$\|O\|$	**Time** θ **(0 / 10 / 20 / 30)**	**Accuracy** θ **(0 / 10 / 20 / 30)**	**Time**	**Accuracy**	**Time**	**Accuracy**
Blocks-World (855)	20	15.6	10	1.1	0.99 / 0.100 / 0.105 / 0.111	36.1% / 38.8% / 70.0% / 89.4%	1.656	83.8%	0.452	52.7%
			30	2.9	0.107 / 0.109 / 0.118 / 0.122	54.4% / 61.1% / 86.1% / 97.2%	1.735	90.0%	0.458	77.7%
			50	4.2	0.113 / 0.113 / 0.120 / 0.127	63.8% / 83.8% / 98.3% / 100.0%	1.836	97.2%	0.462	94.4%
			70	6.5	0.138 / 0.139 / 0.141 / 0.148	81.6% / 94.4% / 100.0% / 100.0%	2.056	98.8%	0.483	96.1%
			100	8.5	0.163 / 0.166 / 0.172 / 0.185	100.0% / 100.0% / 100.0% / 100.0%	2.378	100.0%	0.494	100.0%
Campus (75)	2	8.5	10	1	0.038 / 0.039 / 0.042 / 0.044	93.3% / 100.0% / 100.0% / 100.0%	0.083	100.0%	0.090	100.0%
			30	2	0.048 / 0.050 / 0.055 / 0.057	100.0% / 100.0% / 100.0% / 100.0%	0.091	100.0%	0.089	100.0%
			50	3	0.063 / 0.062 / 0.066 / 0.068	93.3% / 100.0% / 100.0% / 100.0%	0.105	100.0%	0.092	100.0%
			70	4.4	0.060 / 0.060 / 0.063 / 0.065	100.0% / 100.0% / 100.0% / 100.0%	0.112	100.0%	0.095	100.0%
			100	5.5	0.068 / 0.069 / 0.073 / 0.072	100.0% / 100.0% / 100.0% / 100.0%	0.126	100.0%	0.097	100.0%
Easy-IPC-Grid (465)	7.5	11.3	10	1.8	0.585 / 0.588 / 0.609 / 0.623	82.2% / 85.5% / 97.7% / 100.0%	1.206	97.7%	0.770	97.7%
			30	4.3	0.597 / 0.600 / 0.614 / 0.644	86.6% / 93.3% / 97.7% / 100.0%	1.291	98.8%	0.790	98.8%
			50	6.9	0.608 / 0.609 / 0.627 / 0.656	94.4% / 97.7% / 97.7% / 100.0%	1.306	98.8%	0.860	100.0%
			70	9.8	0.629 / 0.628 / 0.661 / 0.715	95.5% / 98.8% / 98.8% / 100.0%	1.715	100.0%	0.932	100.0%
			100	13.3	0.630 / 0.632 / 0.685 / 0.759	100.0% / 100.0% / 100.0% / 100.0%	2.263	100.0%	1.091	100.0%
Intrusion-Detection (465)	15	16	10	1.9	0.197 / 0.200 / 0.211 / 0.233	76.4% / 96.6% / 100.0% / 100.0%	1.130	98.8%	0.506	98.8%
			30	4.5	0.214 / 0.219 / 0.227 / 0.241	94.4% / 100.0% / 100.0% / 100.0%	1.142	100.0%	0.521	100.0%
			50	6.7	0.218 / 0.221 / 0.246 / 0.269	100.0% / 100.0% / 100.0% / 100.0%	1.203	100.0%	0.531	100.0%
			70	9.5	0.219 / 0.223 / 0.258 / 0.274	100.0% / 100.0% / 100.0% / 100.0%	1.482	100.0%	0.568	100.0%
			100	13.1	0.277 / 0.281 / 0.303 / 0.325	100.0% / 100.0% / 100.0% / 100.0%	1.567	100.0%	0.566	100.0%
Kitchen*o* (75)	3	5	10	1.3	0.003 / 0.003 / 0.002 / 0.004	93.3% / 100.0% / 100.0% / 100.0%	0.099	100.0%	0.093	100.0%
			30	3.5	0.003 / 0.004 / 0.005 / 0.005	93.3% / 100.0% / 100.0% / 100.0%	0.111	100.0%	0.107	100.0%
			50	4	0.004 / 0.004 / 0.006 / 0.006	93.3% / 100.0% / 100.0% / 100.0%	0.112	100.0%	0.111	100.0%
			70	5	0.006 / 0.007 / 0.007 / 0.008	93.3% / 93.3% / 100.0% / 100.0%	0.111	100.0%	0.110	100.0%
			100	7.4	0.007 / 0.008 / 0.008 / 0.009	100.0% / 100.0% / 100.0% / 100.0%	0.118	100.0%	0.112	100.0%
Logistics (465)	10	18.7	10	2	0.441 / 0.449 / 0.455 / 0.458	73.3% / 96.6% / 100.0% / 100.0%	1.125	100.0%	0.615	98.8%
			30	5.9	0.447 / 0.452 / 0.461 / 0.466	88.7% / 100.0% / 100.0% / 100.0%	1.195	100.0%	0.663	100.0%
			50	9.5	0.457 / 0.469 / 0.474 / 0.488	96.6% / 100.0% / 100.0% / 100.0%	1.248	98.8%	0.712	98.8%
			70	13.4	0.474 / 0.481 / 0.490 / 0.497	100.0% / 100.0% / 100.0% / 100.0%	1.507	100.0%	0.786	100.0%
			100	18.7	0.498 / 0.505 / 0.513 / 0.522	100.0% / 100.0% / 100.0% / 100.0%	1.984	100.0%	0.918	100.0%

Table 1: Comparison and experimental results of our landmark-based approach against Ramirez and Geffner [8] approach. R&G denotes their plan recognition approach and Filter + R&G denotes the same approach but using our filtering algorithm.

Algorithm 1 Recognize goals and plans using the filtering process and the landmark-based heuristic.

Input: $\Xi = \langle \Sigma, \mathcal{A} \rangle$ *planning domain*, $\mathcal{I}$ *initial state*, $\mathcal{G}$ *set of candidate goals*, O *observations*, and θ *threshold.*

Output: *Recognized goal(s).*

1: **function** RECOGNIZE($\Xi, \mathcal{I}, \mathcal{G}, O, \theta$)
2: $\quad \Lambda_{\mathcal{G}} := \langle \rangle$ $\quad \triangleright$ *Map goals to % of landmarks achieved.*
3: $\quad \Lambda_{\mathcal{G}} :=$ FILTERCANDIDATEGOALS($\Xi, \mathcal{I}, \mathcal{G}, O, \theta$)
4: $\quad$ **return** $\arg \max_{G \in \Lambda_{\mathcal{G}}} h_{prl}(G)$

the filtered candidates, this algorithm then uses h_{prl} to return the recognized goals by estimating the percentage of completion using the set of achieved fact landmarks provided by the filtering process.

Table 1 shows the result of our experiments, which uses six domains from datasets provided by Ramírez and Geffner [8, 7], comprising hundreds of plan recognition problems, *i.e*, a domain description, an initial state, a set of candidate goals $\mathcal{G}$, a hidden goal G in $\mathcal{G}$, and an observation sequence O (10%, 30%, 50%, 70%, or 100% of observability). We use two metrics, the accuracy of recognizing the correct hidden goal G in $\mathcal{G}$ and the speed to recognize a goal, and compare our approach to two other approaches: the approach of Ramírez and Geffner [8] on its own, and this approach combined with our filter. More specifically, we use their faster and most accurate approach. For our approach, we show the accuracy under different filtering thresholds (0%, 10%, 20% and 30%). If threshold $\theta = 0$, our approach gives no flexibility for filtering candidate goals, returning only the goals with the highest percentage of achieved landmarks. Each row of this table shows the observability (% Obs) and averages of the number of candidate goals $|\mathcal{G}|$, the number of observed actions $|O|$, recognition time (seconds), and accuracy. We can see from the table that our approach is both faster and more accurate than Ramírez and Geffner [8], and, when we combine their algorithm with our filter, the resulting approach gets a substantial speedup.

5 Conclusion

We have developed an approach for plan recognition that relies on planning landmarks and a new heuristic based on these landmarks. Landmarks provide key information about what cannot be avoided to achieve a goal, and we show that landmarks can be used efficiently for very accurate plan recognition. We have shown empirically that our approach yields not only superior accuracy results but also substantially faster recognition times for all domains used in evaluating against the state of the art [8] at varying observation levels.

There are multiple avenues for future work, such as: evaluating heuristics and symmetries in classical planning [9]; other landmark extraction techniques [5]; adding a probability interpretation to the observed landmarks and comparing to a recent work [2]; and account for information gain over multiple competing plan hypotheses.

Acknowledgments: This research was carried out in cooperation with HP Brazil using incentives of the Brazilian Informatics Law (# 8.2.48 of 1991).

REFERENCES

[1] Dorit Avrahami-Zilberbrand and Gal A. Kaminka, 'Fast and Complete Symbolic Plan Recognition', in *IJCAI 2005, Edinburgh, Scotland*, pp. 653–658, (2005).
[2] Yolanda E.-Martín, María D. R.-Moreno, and David E. Smith, 'A Fast Goal Recognition Technique Based on Interaction Estimates', in *IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pp. 761–768, (2015).
[3] Christopher W. Geib and Robert P. Goldman, 'Partial Observability and Probabilistic Plan/Goal Recognition', in *Workshop on Modeling Others from Observations (MOO-2005)*, (2005).
[4] Jörg Hoffmann, Julie Porteous, and Laura Sebastia, 'Ordered Landmarks in Planning', *Journal of Artificial Intelligence Research (JAIR)*, **22**(1), 215–278, (November 2004).
[5] Erez Karpas, David Wang, Brian C. Williams, and Patrik Haslum, 'Temporal landmarks: What must happen, and when', in *ICAPS 2015, Jerusalem, Israel, June 7-11, 2015.*, pp. 138–146, (2015).
[6] David Pattison and Derek Long, 'Domain Independent Goal Recognition.', in *STAIRS*, pp. 238–250, (2010).
[7] Miquel Ramírez and Hector Geffner, 'Probabilistic Plan Recognition Using Off-the-Shelf Classical Planners', in *AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, (2010).
[8] Miquel Ramírez and Hector Geffner, 'Plan Recognition as Planning.', in *IJCAI 2009.*, pp. 1778–1783, (2009).
[9] Alexander Shleyfman, Michael Katz, Malte Helmert, Silvan Sievers, and Martin Wehrle, 'Heuristics and Symmetries in Classical Planning', in *AAAI 2015, 2015, Austin, Texas, USA.*, pp. 3371–3377, (2015).
[10] Gita Sukthankar, Robert P. Goldman, Christopher Geib, David V. Pynadath, and Hung Hai Bui, *Plan, Activity, and Intent Recognition: Theory and Practice*, Elsevier, 2014.

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

doi:10.3233/978-1-61499-672-9-1708

Multi-Level Semantics with Vertical Integrity Constraints

Alison R. Panisson[1] and **Rafael H. Bordini**[1] and **Antônio Carlos da Rocha Costa**[2]

Abstract. Operational semantics is a fundamental approach to the formalisation of programming languages and almost a standard when it comes to agent-oriented programming languages. It helps ensure the correctness of interpreters, facilitates their implementation, and supports proofs of important properties. Multi-agent oriented systems are a particular kind of distributed systems and through the semantics of agent languages, operational semantics ended up playing an important role towards ensuring their desired behaviour, even though the operational semantics becomes more involved than originally intended. This work presents a new style for the operational semantics of systems with multiple levels of abstractions (such as multi-agent systems), by providing multi-level transitions (i.e., multiple hierarchical transition systems) with vertical (i.e., inter-level) integrity constraints to ensure consistency of interrelated transitions.

1 Introduction

In multi-agent systems (MAS) there are multiple different levels of specification, each one corresponding to a different conceptual level in the system, and playing important roles in a more general framework for programming MAS (e.g., JaCaMo Framework [1], PopOrg (Populational-Organisational) Model [2], Electronic Institutions [3], etc.). In order to ensure the desired behaviour of MAS, it has been common in the Agents literature to give formal semantics to such systems as transition systems [8, 5, 4, 6] (using operational semantics [7]), expressing the possible states of the system and the necessary conditions for the system to move from one such state to another.

Given the complexity of multi-agent system abstractions, we propose a multi-level operational semantics with vertical (i.e., inter-level) integrity constraints in order to specify those systems. Such vertical integrity constraints aim to ensure that the transition systems giving separate semantics to the individual levels of abstraction are combined in a way that preserves the required interrelations of those levels. Furthermore, they allow some of the required semantic rules to be automatically generated from compact representations of such integrity constraints. In this work, we focus on the functioning of multi-agent organisations, modelling them as multiple, independent but interrelated, transition systems. In particular, we express the specification in terms of *count-as* relations between levels, i.e., relations that express certain combinations of actions — i.e., *processes* — executed at a given abstraction level *count as* actions being executed at an upper level.

2 Vertical Integrity Constraints

In order to exemplify our work, here we focus on *processes* (a ordering of actions/events) which occurs at a level of abstraction and *count-as* actions/events on superior levels. In particular, here we consider three levels in the multi-agent system, the *agents*, the *organisations*, and the *social sub-system*, so we introduce two vertical integrity constraints:

$$\#VIC^{org}_{ag} \Uparrow [\mathcal{AS}_{org}, \mathcal{A}_{org}, \mathcal{AS}_{ag}, Act_{ag}] = S \qquad (1)$$

being $\mathcal{AS}_{org}$ the history of the organisational actions already executed, $\mathcal{A}_{org}$ the set of organisational actions yet to be executed (i.e., that the organisation aims to see executed), $\mathcal{AS}_{ag}$ the history of actions executed by the agents in that organisation, Act_{ag} the set of agent actions being executed at that moment, and S the set of organisational actions that can be considered as executed given that history of agents actions ($\mathcal{AS}_{ag}$ and Act_{ag}) and the count-as relations.

$$\#VIC^{ss}_{org} \Uparrow [\mathcal{AS}_{ss}, \mathcal{A}_{ss}, \mathcal{AS}_{org}, Act_{org}] = S \qquad (2)$$

being $\mathcal{AS}_{ss}$ the history of the social sub-system actions, $\mathcal{A}_{ss}$ the set of social sub-system actions that were not executed, $\mathcal{AS}_{org}$ the history of actions executed by the organisations in that social sub-system, Act_{org} the set of organisational action being executed at that moment, and S the set of social sub-system actions that can be considered as executed given that history of organisational actions ($\mathcal{AS}_{org}$ and Act_{org}) and the count-as relations.

3 Abstract Semantic Rule

The definition of multi-level semantics with vertical integrity constraints between the levels allows just one abstract semantic rule interpreting all *count-as* relations in the multi-agent system, where the corresponding actions can be instantiated to particular cases. The abstract semantic rule is showed below[3]:

$$\frac{\begin{array}{c} execute(\mathtt{act}) \qquad \mathtt{act} \in \mathcal{A} \\ \#VIC^{org}_{ag} \Uparrow [\mathcal{AS}_{org}, \mathcal{A}_{org}, \mathcal{AS}_{ag}, \{\mathtt{act}\}] = \mathcal{S}_{org} \\ \#VIC^{ss}_{org} \Uparrow [\mathcal{AS}_{ss}, \mathcal{A}_{ss}, \mathcal{AS}_{org}, \mathcal{S}_{org}] = \mathcal{S}_{ss} \end{array}}{\begin{array}{lll} (a) & \langle ss_l, \mathcal{O}, \mathcal{A}_{ss}\rangle \ldots \mathcal{AS}_{ss} & \rightarrow_{ss} \quad \langle ss_l, \mathcal{O}, \mathcal{A}'_{ss}\rangle \ldots \mathcal{AS}'_{ss} \\ (b) & \langle org_n, Ag, \mathcal{A}_{org}\rangle \ldots \mathcal{AS}_{org} & \rightarrow_{org} \quad \langle org_n, Ag, \mathcal{A}'_{org}\rangle \ldots \mathcal{AS}'_{org} \\ (c) & \langle ag_m, \mathcal{A}\rangle \ldots \mathcal{AS}_{ag} & \rightarrow_{ag} \quad \langle ag_m, \mathcal{A}'\rangle \ldots \mathcal{AS}'_{ag} \end{array}}$$

where:

$$\begin{array}{lll} (a) & \mathcal{AS}'_{ss} = \mathcal{AS}_{ss} \cup \mathcal{S}_{ss} \\ & \mathcal{A}'_{ss} = c(\mathcal{A}_{ss}, \mathcal{AS}_{ss}, \mathcal{S}_{ss}) \\ (b) & \mathcal{AS}'_{org} = \mathcal{AS}_{org} \cup \mathcal{S}_{org} \\ & \mathcal{A}'_{org} = c(\mathcal{A}_{org}, \mathcal{AS}_{org}, \mathcal{S}_{org}) \\ (c) & \mathcal{AS}'_{ag} = \mathcal{AS}_{ag} \cup \{\mathtt{act}\} \\ & \mathcal{A}' = c(\mathcal{A}, \mathcal{AS}_{ag}, \{\mathtt{act}\}) \end{array}$$

(ABSTRACTSEMANTICRULE)

where the tuple $\langle ss_{id}, \mathcal{O}, \mathcal{A}_{ss}\rangle$ represents the social sub-system, with ss_{id} the social sub-system identifier, $\mathcal{O}$ a set of organisation identifiers, each one representing the organisation populating the social

[1] Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil — Emails: alison.panisson@acad.pucrs.br, rafael.bordini@pucrs.br
[2] Graduate Program in Computers in Education – UFRGS, Porto Alegre, Brazil and Graduate Program in Computer Science – FURG, Rio Grande, Brazil — Email: ac.rocha.costa@gmail.com

[3] We use the notation "$\langle ag_m, \mathcal{A}\rangle \ldots \mathcal{AS}_{ag}$" as a simplified representation for "$\{\langle ag_1, A_{ag_1}\rangle, \ldots, \langle ag_n, A_{ag_n}\rangle\}, \mathcal{AS}_{ag}$".

sub-system, and $\mathcal{A}_{ss}$ the set of actions of the social sub-system. The tuple $\langle org_{id}, Ag, \mathcal{A}_{org} \rangle$ represents an organisation, with org_{id} the organisation identifier, Ag a set of agent identifiers, which are populating that organisation, and $\mathcal{A}_{org}$ the set of organisational actions. The tuple $\langle ag_{id}, \mathcal{A} \rangle$ represents an agent in the multi-agent system, with ag_{id} the agent identifier, and $\mathcal{A}$ the set of actions that the agent is able to execute. At each level of the system (i.e., *agents*, *organisations* and *social sub-systems*), we maintain a history (i.e., trace) of actions already executed at that level, $\mathcal{AS}_{ag}$, $\mathcal{AS}_{org}$, and $\mathcal{AS}_{ss}$, respectively. When an action is executed by an agent, the *VIC*s check if the particular action satisfies some *process* which count-as an action in the superior level, and update the set of organisational and social sub-system in accordance. Further, we use a continuation function $c(A, AS, a)$, in order to specify when the action being executed (i.e., a) is removed or not from the set of actions A, and if other actions are included in the set of actions given the ones that were executed. This function covers both achievement and maintenance tasks in MAS.

4 Example

Imagine the process of exchanging an employee by two organisations $\texttt{org}_1$ and $\texttt{org}_2$. On the top of our multi-agent system, we have the social sub-system `ssys` regulating such exchange through a process for a norm-conforming exchange `norm-conf_exchange`, which is achieved by the sequence of the organisational actions $\{\texttt{dismiss}_{\texttt{org}}, \texttt{appoint}_{\texttt{org}}\}$. The $\texttt{appoint}_{\texttt{org}}$ action is achieved by the sequential process {`interview`, `admission_request`, `admission_processing`, `appoint`}, and $\texttt{dismiss}_{\texttt{org}}$ is achieved by the sequential process {`give_notice`, `dismissal_request`, `dismissal_processing`, `dismiss`}, where all those are actions executed by agent populating those organisation, and that count-as organisational action when the process is followed.

In this section, we present three instances from the abstract semantic rule, in accordance with the scenario described. The first, `ExInterviewAction`, shows an agent called `interviewer` executing the first action of the $\texttt{appoint}_{\texttt{org}}$ process. In this semantic rule, as the *VIC*s are not satisfied (none process is satisfied) only the agent level (c) is updated in accordance.

$$\begin{array}{c} execute(\texttt{interview}) \quad \texttt{interview} \in \mathcal{A} \\ \#VIC^{org}_{ag} \Uparrow [\mathcal{AS}_{org}, \mathcal{A}_{org}, \mathcal{AS}_{ag}, \{\texttt{interview}\}] = \{\,\} \\ \#VIC^{ss}_{org} \Uparrow [\mathcal{AS}_{ss}, \mathcal{A}_{ss}, \mathcal{AS}_{org}, \{\,\}] = \{\,\} \end{array}$$

$$\begin{array}{lll} (a) & \langle \texttt{ssys}, \mathcal{O}, \mathcal{A}_{ss} \rangle \ldots \mathcal{AS}_{ss} & \rightarrow_{ss} \quad \langle \texttt{ssys}, \mathcal{O}, \mathcal{A}_{ss} \rangle \ldots \mathcal{AS}_{ss} \\ (b) & \langle \texttt{org}_2, Ag, \mathcal{A}_{org} \rangle \ldots \mathcal{AS}_{org} & \rightarrow_{org} \quad \langle \texttt{org}_2, Ag, \mathcal{A}_{org} \rangle \ldots \mathcal{AS}_{org} \\ (c) & \langle \texttt{interviewer}, \mathcal{A} \rangle \ldots \mathcal{AS}_{ag} & \rightarrow_{ag} \quad \langle \texttt{interviewer}, \mathcal{A}' \rangle \ldots \mathcal{AS}'_{ag} \end{array}$$

where:

$$\begin{array}{lrcl} (c) & \mathcal{AS}'_{ag} & = & \mathcal{AS}_{ag} \cup \{\texttt{interview}\} \\ & \mathcal{A}' & = & c(\mathcal{A}, \mathcal{AS}_{ag}, \{\texttt{interview}\}) \end{array}$$

(EXINTERVIEWACTION)

$$\begin{array}{c} execute(\texttt{dismiss}) \quad \texttt{dismiss} \in \mathcal{A} \\ \#VIC^{org}_{ag} \Uparrow [\mathcal{AS}_{org}, \mathcal{A}_{org}, \mathcal{AS}_{ag}, \{\texttt{dismiss}\}] = \{\texttt{dismiss}_{\texttt{org}_1}\} \\ \#VIC^{ss}_{org} \Uparrow [\mathcal{AS}_{ss}, \mathcal{A}_{ss}, \mathcal{AS}_{org}, \{\texttt{dismiss}_{\texttt{org}_1}\}] = \{\,\} \end{array}$$

$$\begin{array}{lll} (a) & \langle \texttt{ssys}, \mathcal{O}, \mathcal{A}_{ss} \rangle \ldots \mathcal{AS}_{ss} & \rightarrow_{ss} \quad \langle \texttt{ssys}, \mathcal{O}, \mathcal{A}_{ss} \rangle \ldots \mathcal{AS}_{ss} \\ (b) & \langle \texttt{org}_1, Ag, \mathcal{A}_{org} \rangle \ldots \mathcal{AS}_{org} & \rightarrow_{org} \quad \langle \texttt{org}_1, Ag, \mathcal{A}'_{org} \rangle \ldots \mathcal{AS}'_{org} \\ (c) & \langle \texttt{manager}, \mathcal{A} \rangle \ldots \mathcal{AS}_{ag} & \rightarrow_{ag} \quad \langle \texttt{manager}, \mathcal{A}' \rangle \ldots \mathcal{AS}'_{ag} \end{array}$$

where:

$$\begin{array}{lrcl} (b) & \mathcal{AS}'_{org} & = & \mathcal{AS}_{org} \cup \{\texttt{dismiss}_{\texttt{org}_1}\} \\ & \mathcal{A}'_{org} & = & c(\mathcal{A}_{org}, \mathcal{AS}_{org}, \{\texttt{dismiss}_{\texttt{org}_1}\}) \\ (c) & \mathcal{AS}'_{ag} & = & \mathcal{AS}_{ag} \cup \{\texttt{dismiss}\} \\ & \mathcal{A}' & = & c(\mathcal{A}, \mathcal{AS}_{ag}, \{\texttt{dismiss}\}) \end{array}$$

(EXDISMISSACTION)

In contrast, when an agent (exemplified by the agent named `manager`) executes the last action for the process $\texttt{dismiss}_{\texttt{org}}$, i.e., the action `dismiss`, such organisational action is achieved. This is exemplified by the semantic rule `ExDismissAction`, where we assume that all other action for such $\texttt{dismiss}_{\texttt{org}}$ process have been executed, and therefore the $\#VIC^{org}_{ag} \Uparrow$ is satisfied, i.e., the agents' actions satisfies a process for $\texttt{dismiss}_{\texttt{org}}$. In this case both, the agent and organisational levels change in accordance.

The last semantic rule `ExAppointAction` shows when the last action of the `norm-conf_exchange` process is achieved, i.e., $\texttt{appoint}_{\texttt{org}}$ is achieved by the organisation. Further, the $\texttt{appoint}_{\texttt{org}}$ organisational action is achieved when the agent named `manager` executes the last action of such process, i.e., the action `appoint`. Observe that both *VIC*s are satisfied and all levels in the systems are updated.

$$\begin{array}{c} execute(\texttt{appoint}) \quad \texttt{appoint} \in \mathcal{A} \\ \#VIC^{org}_{ag} \Uparrow [\mathcal{AS}_{org}, \mathcal{A}_{org}, \mathcal{AS}_{ag}, \{\texttt{appoint}\}] = \{\texttt{appoint}_{\texttt{org}_2}\} \\ \#VIC^{ss}_{org} \Uparrow [\mathcal{AS}_{ss}, \mathcal{A}_{ss}, \mathcal{AS}_{org}, \{\texttt{appoint}_{\texttt{org}_2}\}] = \{\texttt{norm-conf\_exchange}\} \end{array}$$

$$\begin{array}{lll} (a) & \langle \texttt{ssys}, \mathcal{O}, \mathcal{A}_{ss} \rangle \ldots \mathcal{AS}_{ss} & \rightarrow_{ss} \quad \langle \texttt{ssys}, \mathcal{O}, \mathcal{A}'_{ss} \rangle \ldots \mathcal{AS}'_{ss} \\ (b) & \langle \texttt{org}_2, Ag, \mathcal{A}_{org} \rangle \ldots \mathcal{AS}_{org} & \rightarrow_{org} \quad \langle \texttt{org}_2, Ag, \mathcal{A}'_{org} \rangle \ldots \mathcal{AS}'_{org} \\ (c) & \langle \texttt{manager}, \mathcal{A} \rangle \ldots \mathcal{AS}_{ag} & \rightarrow_{ag} \quad \langle \texttt{manager}, \mathcal{A}' \rangle \ldots \mathcal{AS}'_{ag} \end{array}$$

where:

$$\begin{array}{lrcl} (a) & \mathcal{AS}'_{ss} & = & \mathcal{AS}_{ss} \cup \{\texttt{norm-conf\_exchange}\} \\ & \mathcal{A}'_{ss} & = & c(\mathcal{A}_{ss}, \mathcal{AS}_{ss}, \{\texttt{norm-conf\_exchange}\}) \\ (b) & \mathcal{AS}'_{org} & = & \mathcal{AS}_{org} \cup \{\texttt{appoint}_{\texttt{org}_2}\} \\ & \mathcal{A}'_{org} & = & c(\mathcal{A}_{org}, \mathcal{AS}_{org}, \{\texttt{appoint}_{\texttt{org}_2}\}) \\ (c) & \mathcal{AS}'_{ag} & = & \mathcal{AS}_{ag} \cup \{\texttt{appoint}\} \\ & \mathcal{A}' & = & c(\mathcal{A}, \mathcal{AS}_{ag}, \{\texttt{appoint}\}) \end{array}$$

(EXAPPOINTACTION)

5 Conclusion

In this work, we introduced an approach to the formalisation of MAS based on operational semantics; we call it multi-level semantics with vertical integrity constraints. The approach allows the representation of the interactions between components of different system-levels. Given the complexity and ubiquity of such multiple levels in MAS, the approach seems to allow for a clearer understanding of such complex semantics. Furthermore, we demonstrate, using count-as relations applied to *processes*, how the proposed style for multi-level operational semantics can be strengthened through the definition of vertical integrity constraints. Such multi-level semantics with vertical integrity constraints allows the independent specification of the various levels typical of the MAS, so that each level is formalised through its own transition system, and the vertical integrity constraints between these transition systems help guarantee that the overall system operates coherently in all possible executions.

REFERENCES

[1] Olivier Boissier, Rafael H Bordini, Jomi F Hübner, Alessandro Ricci, and Andrea Santi, 'Multi-agent oriented programming with jacamo', *Science of Computer Programming*, **78**(6), 747–761, (2013).

[2] Antônio Carlos Rocha Costa and Graçaliz Pereira Dimuro, 'A minimal dynamical mas organization model', *Handbook of Research on Multia-Agent Systems: Semantics and Dynamics of Organizational Models*, 419–445, (2009).

[3] Pablo Noriega, *Agent mediated auctions: the fishmarket metaphor*, Institut d'Investigació en Intelligència Artificial, 1999.

[4] Alison R Panisson, Rafael H Bordini, and Antônio Carlos Rocha Costa, 'Towards multi-level semantics for multi-agent systems', in *3th Workshop-School on Theoretical Computer Science*, (2015).

[5] Alison R. Panisson, Felipe Meneguzzi, Moser Fagundes, Renata Vieira, and Rafael H. Bordini, 'Formal semantics of speech acts for argumentative dialogues', in *AAMAS*, pp. 1437–1438, (2014).

[6] Alison R. Panisson, Felipe Meneguzzi, Renata Vieira, and Rafael H. Bordini, 'Towards practical argumentation in multi-agent systems', in *BRACIS*, (2015).

[7] Gordon D Plotkin, 'A structural approach to operational semantics', (1981).

[8] Renata Vieira, Álvaro Moreira, Michael Wooldridge, and Rafael H. Bordini, 'On the formal semantics of speech-act based communication in an agent-oriented programming language', *J. Artif. Int. Res.*, **29**(1), 221–267, (June 2007).

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

doi:10.3233/978-1-61499-672-9-1710

Supervised Graph-Based Term Weighting Scheme for Effective Text Classification

Niloofer Shanavas and **Hui Wang** and **Zhiwei Lin** and **Glenn Hawe**[1]

Abstract. Due to the increase in electronic documents, automatic text classification has gained a lot of importance as manual classification of documents is time-consuming. Machine learning is the main approach for automatic text classification, where texts are represented, terms are weighted on the basis of the chosen representation and a classification model is built. Vector space model is the dominant text representation largely due to its simplicity. Graphs are becoming an alternative text representation that have the ability to capture important information in text such as term order, term co-occurrence and term relationships that are not considered by the vector space model. Substantially better text classification performance has been demonstrated for term weighting schemes which use a graph representation. In this paper, we introduce a graph-based term weighting scheme, tw-srw, which is an effective supervised term weighting method that considers the co-occurrence information in text for increasing text classification accuracy. Experimental results show that it outperforms the state-of-the-art unsupervised term weighting schemes.

1 INTRODUCTION

A challenging task in text classification is the effective representation of text. The features that represent the document affect the performance of text classification. The documents for classification are usually represented in the vector space model. It assumes that the terms are independent and represents a document as an unordered set of terms and their frequencies. Although it is simple and fast, this representation does not consider structural information (order of words, relationship between words) or the semantics of text. An alternative to vector space model for representing documents is graph-based representation. A document represented as a graph instead of a vector can retain its inherent structure, thereby increasing the classification performance.

Graph-based term weighting schemes improve classification accuracy compared to traditional frequency-based term weighting methods [1, 2, 3]. The existing graph-based term weighting schemes are unsupervised, so the class-separating information is not considered for text classification where class labels are given. Supervised term weighting methods utilize the information on the membership of the training documents in the predefined classes to give higher weights to terms that are distributed differently in the classes [4]. In this paper, a supervised graph-based term weighting scheme is presented, which utilizes the rich information in text and the relationship of the terms to the predefined classes. It is a class-based function that considers the co-occurrence information in text. Experimental results show that the proposed term weighting scheme can indeed improve classification performance.

The rest of the paper is organized as follows. Section 2 explains the proposed supervised graph-based term weighting scheme. Section 3 presents the experimental results of the proposed term weighting scheme for text classification. Finally, Section 4 concludes the paper and discusses possible future work.

2 PROPOSED TERM WEIGHTING SCHEME FOR TEXT CLASSIFICATION

We have focused our study on undirected co-occurrence graphs, an effective structure-based representation of text, to capture the co-occurrence of words. The text documents are pre-processed by removing stop words and stemming before converting it to co-occurrence graphs. The nodes in the co-occurrence graph represent the unique terms in text and the edges link co-occurring terms within a sliding window of fixed length. Each document is represented by a co-occurrence graph. The terms are weighted based on the node's importance in the graph. Since degree centrality measures perform the best on co-occurrence graph [1], we used it to assign weights to terms. The degree centrality scores of the nodes are used to convert the co-occurrence graph to vector-based representation.

The research done on supervised term weighting schemes for text classification emphasizes the fact that term weighting should depend on the classification task. Text classification performance can be improved by giving more weights to terms that support in the classification of documents into the right class. This is done by utilizing the known information on the training documents in the predefined classes. In [5], a supervised term weighting scheme $tf * rf$ is proposed where tf is the term frequency and rf is the relevance frequency. In [6], the supervised term weighting scheme proposed in [5] is modified to compute a single relevance frequency value for each term for a classification task with n classes by taking the maximum of the n relevance frequency values of each term. The rf factor defined in [5] gives greater weights to terms that are more concentrated in the positive class than in the negative class. But, this weighting measure cannot handle imbalanced data. This happens when one or more classes contain more training documents than the rest of the classes. In such cases, the rf factor gives low weights to terms in the classes with few training samples (minor classes). To overcome this problem, a probability-based approach for term weighting is defined in [7] to assign better weights to terms in minor classes. However, this term weighting scheme results in overweighting of commonly occurring terms.

In order to avoid the problems that reduces the effectiveness of supervised term weighting schemes such as overweighting of commonly occurring terms, higher weights for terms in classes with large

[1] School of Computing and Mathematics, Ulster University, United Kingdom, email: shanavas-n@email.ulster.ac.uk, {h.wang, z.lin, gi.hawe}@ulster.ac.uk

number of training documents, our approach considers all the below three elements for computation of the term's relevance in the classes.

1. The term's concentration in each class as compared to its concentration in other classes.
2. The number of documents in each class that do not contain the term.
3. The average density of the term in the classes.

The new supervised term weight factor, which we name as supervised relevance weight (srw), takes into account the above three elements for its computation as discussed below.

Table 1. Notations used in the supervised term weighting scheme

Notation	Description
a	The number of documents in class C_i that contain the term t.
b	The number of documents in class C_i that do not contain the term t.
c	The number of documents not in class C_i that contain the term t.

The notations a, b and c used in the supervised term weighting scheme are explained in Table 1. In Equation 1, $(a/max(1,c))$ determines the concentration of the term t in class C_i as compared to its concentration in other classes and $(a/max(1,b))$ helps to reduce the higher weights assigned to terms in classes having more training documents by considering the number of documents in class C_i that do not contain the term t.

$$class\_rel\_prob(t, C_i) = \log_2\left(2+\frac{a}{max(1,c)}\right) * \log_2\left(2+\frac{a}{max(1,b)}\right) \quad (1)$$

$class\_rel\_prob(t, C_i)$ defined in Equation 1 is computed for each class C_i. The maximum of $class\_rel\_prob(t, C_i)$ values of the term t, $max(class\_rel\_prob(t, C_i))$, is calculated to obtain a single value for the term t. In order to reduce the overweighting of commonly occurring terms, the average density of the term t in the classes is determined by dividing the sum of densities of the term t in the classes by the number of classes. The computation of average density of the term t is shown in Equation 2 where N_i is the total number of documents in class C_i and C is the total number of classes.

$$avg\_density(t) = \frac{\sum_{i=1}^{C}\left(\frac{a}{N_i}\right)}{C} \quad (2)$$

The supervised relevance weight (srw) of each term t determines the relevance of the term in the classes. It gives higher weights to terms that help in distinguishing the documents in different classes. It is calculated as shown below in Equation 3.

$$srw = max(class\_rel\_prob(t, C_i)) * log_{10}\left(\frac{1}{avg\_density(t)}\right) \quad (3)$$

The new improved term weighting measure (tw-srw) for a term t represented by a node in the co-occurrence graph is defined as the product of tw and srw i.e. $tw * srw$ where tw is the term weight determined by the centrality score for the term t in the graph.

3 EXPERIMENTS AND RESULTS

We have experimented with four pre-processed datasets[1], WebKB, R8, R52, 20 Newsgroups, to evaluate the proposed supervised term weighting scheme for text classification. The documents in the training and testing set are converted to co-occurrence graphs that capture terms that co-occur within a sliding window of size 2. The performance of the supervised graph-based term weighting measure (tw-srw), traditional term weighting measure i.e. term frequency - inverse document frequency (tf-idf) and graph-based term weighting measures such as tw and tw-idf are evaluated for text classification with SVM. tw is determined by the degree centrality score and tw-idf is computed as the product of tw and inverse document frequency (idf). We used the linear support vector machine implementation in scikit-learn called the LinearSVC as the classifier and set the penalty parameter C of the error term to its default value of 1 and the loss function to hinge. The proposed weighting scheme consistently outperforms the unsupervised term weighting schemes for the four datasets tested as shown in Table 2, Table 3 and Table 4.

Table 2. Precision scores for different term weighting schemes

Dataset	tf-idf	tw	tw-idf	tw-srw
WebKB	0.8459	0.8961	0.8757	**0.9111**
R8	0.9622	0.9689	0.9758	**0.9802**
R52	0.9200	0.9031	0.9389	**0.9549**
20 Newsgroups	0.7813	0.7845	0.8353	**0.8462**

Table 3. Recall scores for different term weighting schemes

Dataset	tf-idf	tw	tw-idf	tw-srw
WebKB	0.8467	0.8933	0.8746	**0.9112**
R8	0.9621	0.9685	0.9758	**0.9799**
R52	0.9190	0.9186	0.9435	**0.9552**
20 Newsgroups	0.7763	0.7814	0.8335	**0.8441**

Table 4. F1 scores for different term weighting schemes

Dataset	tf-idf	tw	tw-idf	tw-srw
WebKB	0.8462	0.8902	0.8693	**0.9108**
R8	0.9618	0.9681	0.9756	**0.9799**
R52	0.9146	0.9037	0.9370	**0.9520**
20 Newsgroups	0.7760	0.7740	0.8301	**0.8430**

4 CONCLUSION

Effective representation that considers the structure information in text and an appropriate term weighting measure that takes into account the relationship between terms and the term's relevance to the classification task increases the performance of text classification. An interesting future work is to explore graph-based term weighting measures that consider more complex dependencies in text for classification task.

REFERENCES

[1] K. Valle and P. Ozturk, 'Graph-based Representations for Text Classification', India-Norway Workshop on Web Concepts and Technologies, Trondheim, Norway, 2011.

[2] F. D. Malliaros and K. Skianis, 'Graph-Based Term Weighting for Text Categorization', Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015, pp. 1473-1479, 2015.

[3] S. Hassan, R. Mihalcea, and C. Banea, 'Random-Walk Term Weighting for Improved Text Classification', Proceedings of the IEEE International Conference on Semantic Computing, ICSC 2007, pp. 242-249, 2007.

[4] F. Debole and F. Sebastiani, 'Supervised Term Weighting for Automated Text Categorization', Text Mining and its Applications: Results of the NEMIS Launch Conference, Springer Berlin Heidelberg, pp. 81-97, 2004.

[5] M. Lan, C. L. Tan, J. Su, and Y. Lu, 'Supervised and Traditional Term Weighting Methods for Automatic Text Categorization', IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 721-735, 2009.

[6] N. P. Xuan and H. L. Quang, 'A New Improved Term Weighting Scheme for Text Categorization', Knowledge and Systems Engineering: Proceedings of the Fifth International Conference KSE 2013, Springer International Publishing, pp. 261-270, 2014.

[7] Y. Liu, H. T. Loh, and A. Sun, 'Imbalanced text classification: A term weighting approach', Expert Systems with Applications, pp. 690-701, 2009.

[1] http://ana.cachopo.org/datasets-for-single-label-text-categorization

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

doi:10.3233/978-1-61499-672-9-1712

Temporal Planning with Constants in Context

Josef Bajada[1] and **Maria Fox** and **Derek Long**

1 INTRODUCTION

Required concurrency [3] can cause actions to interfere with running continuous effects. This interference can modify the rate of change, including the polarity, of a continuous effect. In this work, we propose a mechanism to support discrete interference of rates of change caused by instantaneous actions, the start and end endpoints of other durative actions, and numeric timed initial fluents [9]. Current temporal planners have very limited support for such numeric dynamics. COLIN [2] reduces a temporal numeric planning problem to a linear program (LP), but operates on an implicit assumption that the rate of change of a durative action's continuous effect is constant throughout its execution. In this work we propose some enhancements to the algorithms used in COLIN [2], in order to support discrete interference of continuous effects, and a new planner, DICE, was developed to implement them.

2 CONSTANTS IN CONTEXT

A *context* refers to the interval between two adjacent discrete *happenings* in a temporal plan, where a happening refers to the time-stamp of a discrete state transition [5]. We only assume that the rate of change is constant throughout a temporal context. This enables a single durative action to have piecewise linear continuous effects.

Definition 2.1. A temporal *context*, $C = [t_i, t_{i+1}]$, in a plan's happening sequence, T, is an interval enclosed by two adjacent discrete happenings, t_i and t_{i+1}, where $0 \leq t_i < t_{i+1}$, and there is no intermediate *happening*, t, that is $\{t | t_i < t < t_{i+1}\} \cap T = \emptyset$.

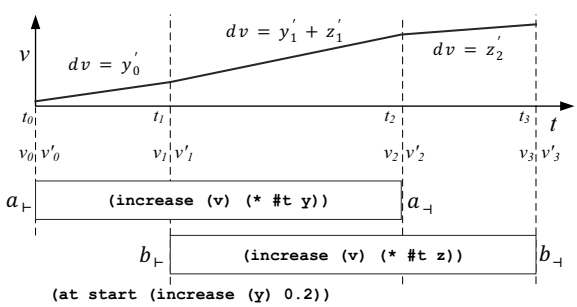


Figure 1. Linear Continuous Effects with Constants in Context.

An LP is used to verify the feasibility and validity of a temporal plan. Its variables consist of the happenings that correspond to the discrete steps of the plan, together with the values of the non-time-dependent [1] numeric fluents before and after each happening. The constraints of the LP consist of temporal constraints between the steps of the plan, mainly the ordering constraints and duration conditions of durative actions, together with numeric pre-conditions and effects of each action in the plan. The rate of change of each continuous effect on a variable, v, is computed dynamically within each temporal context, from the list of durative actions running concurrently, as illustrated in Figure 1.

The rate of change of v in temporal context i is computed from the continuous effects running throughout context i, denoted $ceffs_i$, on v. This is shown in Equation 1, where $ceffs_i$ is represented as a multiset, since the same continuous effect could take place $n > 0$ times concurrently. Each continuous effect, $expr$, on v, is evaluated in the context of the discrete state, s_i, that initiated context i.

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = \sum_{\langle v, expr\rangle \rightarrow n \in ceffs_i} n \cdot expr(s_i) \tag{1}$$

Since continuous effects depend on action durations, whether a sequence of actions achieves a numeric goal or not could depend on the chosen schedule for those actions. We refer to such numeric goals as *schedule dependent*. When during forward search the planner evaluates a state against such goals, G, a dummy action, a_G, with precondition $pre(a_G) = G$, and effects $eff(a_G) = \langle \emptyset, \emptyset, \emptyset \rangle$, is appended to the current plan. If the LP for the resultant plan is solvable the schedule dependent goals are achievable with the plan.

3 A NUMERIC ENHANCED TRPG

The Temporal Relaxed Planning Graph (TRPG) [2] adapts the Metric-FF delete relaxation heuristic [6], and associates a time-stamp to each action layer, which represents the earliest time at which the actions in that layer can be applied [2]. However, the TRPG does not take into account actions whose effects indirectly enable numeric goals to be achieved. A *numeric enhanced* TRPG (TRPGne) is proposed, which takes into account richer numeric causality. It propagates effects on variables that are used in effect expressions of other variables, and identifies *implicit intermediate goals* by inferring new numeric preconditions on actions that achieve numeric goals. These preconditions are then used during relaxed plan extraction.

4 EHC WITH ASCENT BACKTRACKING

Enforced Hill-Climbing (EHC), a popular heuristic search algorithm used in classical [8], numeric [6], and temporal planners [2], suffers from an inherent weakness that hill-climbing decisions could lead to a dead end [7]. This issue becomes more evident in planning problems with temporal and numeric bounded constraints, where the heuristic can be too optimistic and leads to a constraint violation.

[1] King's College London, United Kingdom. email: josef.bajada@kcl.ac.uk

Table 1. Experimental results for three domains, performed on an Intel® Core™ i7-3770 CPU @ 3.40Ghz, allocated a maximum of 3GB RAM.

	Plantery Rover				Intelligent Pump Control				Demand-Side Electricity Management			
	DICE		UPMurphi (1.0)		DICE		UPMurphi (20.0)		DICE		UPMurphi (10.0)	
#	Time (s)	States	Time (s)	States	Time (s)	States	Time (s)	States	Time (s)	States	Time (s)	States
1	8.208	622	22.84	805,145	0.348	40	2.04	51,137	0.899	76	78.14	3,740,370
2	8.243	429	153.04	5,264,013	0.605	55	135.68	3,244,517	1.099	97	1,437.38	61,313,989
3	15.695	442	•	•	0.683	70	•	•	1.782	125	•	•
4	17.589	417	•	•	1.023	86	•	•	1.548	138	•	•
5	43.653	431	•	•	1.404	103	•	•	1.503	204	•	•
6	45.455	474	•	•	1.197	81	•	•	3.582	247	•	•
7	72.977	518	•	•	1.744	86	•	•	6.932	310	•	•
8	152.903	666	•	•	2.522	97	•	•	11.623	292	•	•
9	117.159	647	•	•	3.7	138	•	•	14.805	321	•	•
10	160.035	755	•	•	5.897	132	•	•	12.971	298	•	•

When EHC fails, most planners resort to an exhaustive search such as Weighted A*. However, this means that a wrong hill-climb late in the search could cause EHC to fail even if it is close to a solution.

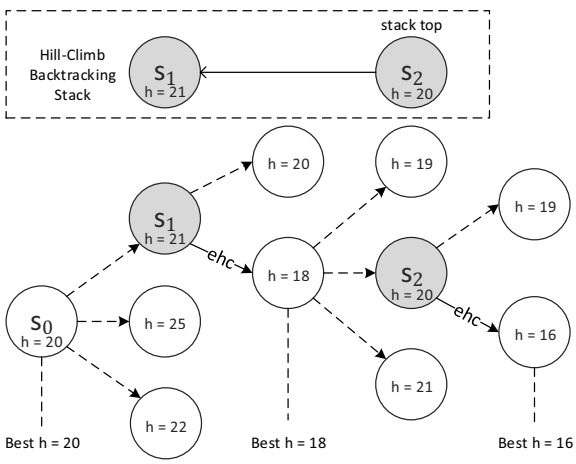


Figure 2. EHC with Ascent Backtracking.

We propose enhancing EHC with an *ascent backtracking* mechanism. A Hill-Climb Backtracking Stack of states from which an enforced hill-climb was performed is stored in memory, as illustrated in Figure 2. This introduces negligible memory overheads. When EHC fails, the state at the top of the stack is popped, to reverse the latest hill-climb. At this point all the *helpful actions* are reconsidered, without performing any enforced hill-climbing if one of them has a better heuristic value than the one encountered so far. If this also fails, the next state is popped off the stack and the process is repeated, until a solution is found or the stack is empty.

5 EVALUATION

The algorithms described above were implemented in a new planner, referred to as DICE (Discrete Interference of Continuous Effects). It was developed in Scala, which runs on the Java™ Virtual Machine. The performance of DICE was compared to that of the PDDL-based hybrid planner, UPMurphi 3.1 [4], which uses a time discretization approach to support complex non-linear functions. Tests were performed on 3 domains that feature *constants in context*. The problem instances for each domain increase in complexity incrementally. The time-step used in UPMurphi is indicated in parentheses in Table 1, which shows the results from both planners. Missing timings indicate that the planner ran out of memory.

6 CONCLUSION

In this work we have proposed a mechanism through which planning with rich numeric characteristics under required concurrency can be performed. The algorithms used by COLIN [2] were enhanced in order to support durative actions whose continuous effects change their gradient at specific time points due to interference from other discrete actions. These algorithms were implemented in a new planner called DICE. We also proposed a new heuristic, the TRPGne, which performs a better analysis of numeric causality. EHC was also complemented with ascent backtracking, which recovers from dead-ends by reversing the latest hill-climb decisions, and adds negligible memory overheads. DICE was evaluated on domains featuring these rich numeric characteristics and compared with UPMurphi [4], the only known PDDL-based planner that supports these characteristics.

ACKNOWLEDGEMENTS

This research was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) as part of the project entitled *The Autonomic Power System* (Grant Ref: EP/I031650/1).

REFERENCES

[1] J. Bajada, M. Fox, and D. Long, 'Temporal Planning with Semantic Attachment of Non-Linear Monotonic Continuous Behaviours', in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI-15)*, pp. 1523–1529, (2015).

[2] A. Coles, A. Coles, M. Fox, and D. Long, 'COLIN: Planning with Continuous Linear Numeric Change', *Journal of Artificial Intelligence Research*, **44**, 1–96, (2012).

[3] W. Cushing, S. Kambhampati, Mausam, and D. S. Weld, 'When is Temporal Planning Really Temporal?', in *20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, (2007).

[4] G. Della Penna, D. Magazzeni, F. Mercorio, and B. Intrigila, 'UPMurphi: A Tool for Universal Planning on PDDL+ Problems.', in *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS 2009)*, pp. 106–113. AAAI, (2009).

[5] M. Fox and D. Long, 'PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains', *Journal of Artificial Intelligence Research*, **20**, 61–124, (2003).

[6] J. Hoffmann, 'The Metric-FF Planning System: Translating "Ignoring Delete Lists" to Numeric State Variables', *Journal of Artificial Intelligence Research*, **20**, 291–341, (2003).

[7] J. Hoffmann, 'Where "Ignoring Delete Lists" Works: Local Search Topology in Planning Benchmarks', *Journal of Artificial Intelligence Research*, 685–758, (2005).

[8] J. Hoffmann and B. Nebel, 'The FF Planning System: Fast Plan Generation Through Heuristic Search', *Journal of Artificial Intelligence Research*, **14**, 253–302, (2001).

[9] C. Piacentini, V. Alimisis, M. Fox, and D. Long, 'An Extension of Metric Temporal Planning with Application to AC Voltage Control', *Artificial Intelligence*, **229**, 210–245, (2015).

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

doi:10.3233/978-1-61499-672-9-1714

Secure Multi-Agent Planning Algorithms

Michal Štolba[1] and **Jan Tožička**[1] and **Antonín Komenda**[1]

Abstract. Multi-agent planning (MAP) is often motivated by the preservation of private information. Such motivation is not only natural for multi-agent systems, but is one of the main reasons, why MAP problems cannot be solved centrally.

In this paper, we analyze privacy leakage of the most common MAP paradigms. Then, we propose a new class SECMAP of secure MAP algorithms and show how the existing techniques can be modified to fall in the proposed class.

1 Introduction

Cooperative multi-agent planning models the problems in which multiple agents need to find a plan fulfilling a common goal. The reason the agents cannot simply feed their problem descriptions into a centralized planner typically lies in that although the agents cooperate, they want to share only the information necessary for their cooperation, but not the information about their inner processes.

A number of planners solving Multi-Agent Planning (MAP) has been proposed in recent years, such as MAFS [4], FMAP [6], PSM [7] and GPPP [3]. Although all of the mentioned planners claim to be privacy-preserving, thorough formal treatment of such claims is rather scarce. The privacy of MAFS is discussed in [4] and expanded upon in [1], proposing Secure-MAFS, a version of MAFS with stronger privacy guarantees.

We propose a new class of MAP algorithms, SECMAP and show that it preserves more privacy than the existing algorithms and how the existing algorithms can be modified to be SECMAP.

1.1 Multi-Agent Planning

In this contribution we use the MA-STRIPS [2] formalism to describe MAP. Formally, for a set of agents $\mathcal{A}$, a MAP problem $\mathcal{M} = \{\Pi_i\}_{i=1}^{|\mathcal{A}|}$ is a set of agents' local STRIPS problems. An agent problem of agent $\alpha_i \in \mathcal{A}$ is defined as $\Pi_i = \langle \mathcal{F}_i, \mathcal{O}_i, s_I, s_\star \rangle$, where $\mathcal{F}_i \subseteq \mathcal{F}$ is a set of facts partitioned into the set $\mathcal{F}^{\mathsf{pub}}$ and $\mathcal{F}_i^{\mathsf{priv}}$ of public (common to all agents) and private (of agent α_i) facts. The state $s_I \subseteq \mathcal{F}$ is the initial state and $s_\star \subseteq \mathcal{F}$ represents the goal condition. The set $\mathcal{O}_i$ of actions comprises of three pairwise disjoint sets: a set $\mathcal{O}_i^{\mathsf{priv}}$ of private actions of α_i, a set $\mathcal{O}_i^{\mathsf{pub}}$ of public actions of α_i and a set $\mathcal{O}^{\mathsf{proj}}$ of public projections of other agents' actions. *Public projections, e.g.* $\pi^{\triangleright}$, of actions / (partial) plans / problem which are shared with other agents are restrictions to public facts and actions. *Local solution* is a solution of Π_i and *global solution* is a solution of the whole $\mathcal{M}$.

A public plan is α_i-extensible, if by adding $a_k \in \mathcal{O}^{\mathsf{priv}}$ to the plan we can obtain a local solution to Π_i. According to [7], a public plan α_i-extensible by all $\alpha_i \in \mathcal{A}$ is a global solution to $\mathcal{M}$.

[1] {stolba,tozicka,komenda}@agents.fel.cvut.cz, Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic

2 Analysis of MAP Algorithms

To analyze the worst-cases of different planning paradigms, we alter the multi-agent planning problem $\mathcal{M}$. Let $\mathcal{M}^*$ be the problem of finding all solutions of $\mathcal{M}$. This modified problem corresponds to the worst-case execution of state-space search algorithms (explore complete search space), partial-order planning algorithms (explore all possible partial plans) and coordination-space search algorithms (explore all possible combinations of local plans).

Let $T(\Pi_i)$ denote a structure containing all solutions of Π_i and $T^*(\mathcal{M})$ denote a structure containing all solutions of $\mathcal{M}^*$. Obviously, $T^*(\mathcal{M})$ represents the minimal knowledge that is revealed by the solution of $\mathcal{M}^*$ and thus also by the worst case scenario in $\mathcal{M}$. It is not tractable to achieve $T^*(\mathcal{M})$ leakage as it is at least as difficult to solve the $\mathcal{M}^*$ problem.

2.1 Privacy Leakage of MAP Algorithms

There are two dominating MAP paradigms: FS is a forward-chaining (or analogously backward-chaining) state-space search. In the multi-agent version, each state expanded by a public action is sent to all relevant agents. Examples of such planners are MAFS [4], SECURE-MAFS [1], or forward-chaining Partial Order Planning (POP). In POP (e.g. FMAP [6]), the public projections of plans are shared in order to coordinate the exploration. CS is a coordination-space search, a paradigm specific for multi-agent planning, where agents attempt to agree on a coordination scheme (public projections of local solutions) which is then extended by private actions of all agents. Examples of such are the PSM [7] and GPPP [3] planners.

We analyze the leakage of private information of the described planning paradigms and particular planners in the worst-case scenario, that is when solving $\mathcal{M}^*$. We will focus on three types of private knowledge leakages. *Superfluous plans* are (partial) plans revealed by the algorithm without being the actual solution (or its prefix). *Superfluous distinct states* are publicly equivalent states s, s' revealed that $s \neq s'$ and either s or s' is not part of the solution. The most common situation where the superfluous distinct state information leaks is the use of unique state labels. And finally *superfluous action applicability* is an information of applicability of an action a on two distinct publicly equivalent states s, s' s.t. $a^{\triangleright}$ is applicable in both $s^{\triangleright}, s'^{\triangleright}$ known to the adversary that s, s' are distinct states and either s or s' is not part of the solution, in other words, the states s, s' are superfluous distinct states.

Forward/Backward State-Space Search The most significant source of leakage in state-space search algorithms is the use of unique IDs representing the private parts of the states, thus distinguishing publicly equivalent states even when it is not necessary.

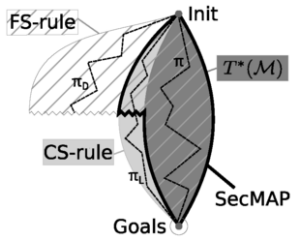


Figure 1. Portions of the state space leaked by application of the CS-RULE, FS-RULE and their combination, where π_D is a sequence of actions leading to a dead-end, π_L is a local plan for Π which cannot be extended to form a global plan for $\mathcal{M}$ and π is a global plan for $\mathcal{M}$.

A multi-agent forward search algorithm jointly explores the state-spaces of all agents, therefore only globally reachable states are explored. A source of superfluous distinct states is that dead-end states are also explored, communicated with other actions and subsequently action applicability is revealed. There are no superfluous plans in FS.

The SECURE-MAFS [1] algorithm reduces privacy leakage by not communicating a state with equal public and other agents' private parts twice. While this approach reduces the number of revealed states and actions, it does not prevent the exploration of dead-end states.

Coordination-Space Search Only states and actions which appear in some local plan are explored in the coordination-space search, that is states which are locally reachable and are not local dead-ends. On the contrary, parts of the state-space which are not globally reachable may be explored as well. In CS, only the necessary publicly equivalent states need to be distinguished, which also results in less superfluous action applicability revealed.

2.2 Designing a Secure Multi-Agent Planner

Based on the above analysis, we can attempt to improve the existing algorithms to reduce leaked private information when solving $\mathcal{M}^*$. Let us state three rules preventing privacy leakage based on the techniques used in the existing algorithms:

CS-RULE: Before communicating a state s, make sure it is part of a local solution to the agent's problem Π_i.

FS-RULE: Before communicating a state s, make sure it is reachable in $\mathcal{M}$.

$\triangleright$-RULE: Do not communicate a state with equivalent public and other agent's private parts more than once.

Figure 1 illustrates portions of the state space leaked by application of the CS-RULE, FS-RULE and their intersection. The $\triangleright$-RULE is not shown in the figure as it does not directly influence the search space, but rather make the leaked information less dense. Obviously, the best algorithm would expand and communicate only states of $T^*(\mathcal{M})$, thus resulting in zero leakage, but that would require to check whether a state is part of a global solution (i.e. solving $\mathcal{M}^*$).

We propose a class of algorithms called SECMAP, containing algorithms which follow all three proposed rules when communicating about states, actions and plans.

2.3 SECMAP Algorithms

The rules defining SECMAP are constructive and thus they can help us modify each of algorithms to fall in the SECMAP class.

MAFS and SECURE-MAFS already satisfy the FS-RULE as all reached states during the search are globally reachable. To satisfy the CS-RULE, the agents need to verify that the extracted state s is part of some local solution, before sending it to other agents. Since MAFS assures that s is (globally) reachable, it is enough to check that also the goal is reachable from this state using $\mathcal{O}_i$ actions. Such check requires to solve new local planning task and if it is unsolvable the state s can to be ignored. The $\triangleright$-RULE can be satisfied by the same way as in SECURE-MAFS, that is never sending a state s which differs only in the private part of the sending agent.

PSM builds local plans in parallel by all agents. These plans are exchanged among all agents, therefore all agents have (in the end) public projections of all agents' local solutions. A non-empty intersection of these solutions represent global solutions (a public plan α_i-extensible by all $\alpha_i \in \mathcal{A}$ is a global solution [7]).

PSM naturally fulfills the CS-RULE. To satisfy the FS-RULE, the secure variant SECMAP-PSM must not send the whole local solutions π at once, but each agent has to check prefixes of the generated plans whether they are all globally reachable. Again, the $\triangleright$-RULE can be satisfied by never sending a state s which differs only in the agent's own private part from some already sent state s'.

Theorem 1. *The* SECMAP-MAFS *and* SECMAP-PSM *algorithms do not leak more information than* SECMAP.

Proof. The proof can be found in [5]. □

3 Conclusions and Future Work

We have identified which cases of information leakage are presented in the most common multi-agent planning paradigms and a new class SECMAP of privacy preserving algorithms has been proposed. This class is guaranteed to leak less information than any currently known algorithm for a certain classes of problems. We proposed how to change the existing planners to belong to SECMAP for the price of increased computational complexity as all SECMAP algorithms require another (albeit local) planning process. It allows to decrease the need for communicating information which can be used to deduce private parts of the problem. Proposing a practically efficient SECMAP algorithm is left for future work.

Acknowledgments

This research was supported by the Czech Science Foundation (grant no. 15-20433Y) and by the Grant Agency of the CTU in Prague (grant no. SGS14/202/OHK3/3T/13).

References

[1] Ronen I. Brafman, 'A privacy preserving algorithm for multi-agent planning and search', in *Procs. of the IJCAI'15*, pp. 1530–1536, (2015).
[2] Ronen I. Brafman and Carmel Domshlak, 'From one to many: Planning for loosely coupled multi-agent systems', in *Procs. of the ICAPS'08*, pp. 28–35, (2008).
[3] Shlomi Maliah, Guy Shani, and Roni Stern, 'Collaborative privacy preserving multi-agent planning', *Procs. of the AAMAS'16*, 1–38, (2016).
[4] Raz Nissim and Ronen I. Brafman, 'Distributed heuristic forward search for multi-agent planning', *JAIR*, **51**, 293–332, (2014).
[5] Michal Štolba, Jan Tožička, and Antonín Komenda, 'Secure multi-agent planning', in *Proc. of the Intl. Workshop on PrAISe*, (2016).
[6] Alejandro Torreño, Eva Onaindia, and Oscar Sapena, 'FMAP: distributed cooperative multi-agent planning', *AI*, **41**(2), 606–626, (2014).
[7] Jan Tožička, Jan Jakubův, Antonín Komenda, and Michal Pěchouček, 'Privacy-concerned multiagent planning', *KAIS*, 1–38, (2015).

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

doi:10.3233/978-1-61499-672-9-1716

Analysis of Swarm Communication Models

Musad Haque[1] **Christopher Ren**[1] **Electa Baker**[1] **Douglas Kirkpatrick**[2] and **Julie A. Adams**[1]

Abstract. The biological swarm literature presents communication models that attempt to capture the nature of interactions among the swarm's individuals. The evaluated hypothesis that the choice of a biologically inspired communication model can affect an artificial swarm's performance for a given task was supported.

1 INTRODUCTION

Bio-inspired artificial swarms inherit desirable properties from their counterparts in nature, such as decentralized control laws, scalability, and robustness, yet despite the beneficial properties, a poorly designed communication network for an artificial swarm can lead to undesirable consequences, such as the swarm fragmenting into multiple components [4]. Proposed communication models for group behavior in animals include the metric [2], the topological [1], and the visual models [6]. The metric model is directly based on spatial proximity: two individuals interact if they are within a certain distance of one another [2]. Ballerini *et. al.*'s [1] topological model requires each individual to interact with a finite number of nearest group members. The visual model permits an individual to interact with other agents in its visual field [6]. The development of communication networks is described as "one of the main challenges" in swarm robotics [4]. The importance of the reported research is that the task performance of a swarm can be amplified through the deliberate selection of a communications model.

2 RELATED WORK

Prior research compared the communication models to identify which model best explains the propagation of information within biological species. Stranburg-Peshkin *et al.* [6] reported that for golden shiners, *Notemigonus crysoleucas*, the visual model best predicts information transfer within the school. The Metric and topological models were compared for flocks of European starlings, *Sturnus vulgaris* [1], and the topological model most accurately described the starlings' information network (starlings coordinate, on average, with the nearest 6 to 7 birds [1]). The experiment also compared the cohesion of simulated swarms using the topological and metric models, and the topological model generated more cohesive swarms.

3 COORDINATION ALGORITHMS

If there is a communication link from agent i to agent j, where $i \neq j$, agent j is a neighbor of agent i. The neighbor set of agent i, denoted by $\mathcal{N}_i(t)$ is the collection of all the neighbors of agent i at time t.

The coordination of the swarm agents is designed through a multi-level coordination algorithm. At the higher level of abstraction, an agent's neighbors are determined by the communication model. Thus, for agent i, the communication model constructs the set $\mathcal{N}_i(t)$ at each time t. All agents within a distance d_{met} from agent i are i's neighbors in the metric model. $\mathcal{N}_i(t)$ is the set containing the n_{top} nearest agents from agent i in the topological model. A sensing range, a blindspot, and occlusion are used to describe the visual model [6]. Agent j is a neighbor of agent i, if three conditions are met: 1) The distance between the agents is less than d_{vis}, 2) Agent j is not in agent i's blindspot, and 3) The line-of-sight between the agents is not occluded by another agent or object in the environment.

At the lower abstraction level, agents only interact with their neighbors and the nature of this interaction is governed by three rules: repulsion, orientation, and attraction, based on Reynolds's rules for boids (see [5]). Each agent has a zone of repulsion, orientation, and attraction, parameterized by the distances r_{rep}, r_{ori}, and r_{att}, respectively, where $r_{rep} < r_{ori} < r_{att}$. The heading of each agent i is updated as follows: 1) Veer away from all agents in $\mathcal{N}_i(t)$ within a distance r_{rep}, 2) Align velocity with all agents in $\mathcal{N}_i(t)$ that are between a distance of r_{rep} and r_{ori}, and 3) Remain close to all agents $j \in \mathcal{N}_i(t)$ that are between a distance of r_{ori} and r_{att} [5].

4 THE SEARCH FOR A GOAL EXPERIMENT

The objective of the artificial swarm during the *search for a goal* is to locate a single goal location. The world is bounded by a wall that exerts a repulsive force. An agent can sense the goal if it is within r_{att} of the goal area's location. Once an agent locates the goal, it can communicate the location to its neighbors. Agents aware of the goal's location update their headings by equally weighing the desire to go to the goal and the desire to follow the interaction rules.

The *percent reached* (R) determines the number of agents that reached the goal area, expressed as a percentage of the swarm's size, N, at the end of the task. The hypothesis for this task is H_{sg}: $R_V > R_T > R_M$, where the subscripts indicate the metric (M), the topological (T) and the visual (V) models. The hypothesis is based on the *potentially* long-range sensing capabilities associated with the visual model. Agents favorably oriented and not occluded have a higher chance of communicating with an agent that has located the goal.

The experimental parameters are presented in Table 1. A trial is defined as a single simulation run for a given selection of parameters, $(N, N_{obs}, r_{rep}, r_{ori}, r_{att})$. The simulation for each trial runs for $1,000$ iterations. Twenty-five trials for each parameter selection were completed. The total number of trials for the search for a goal task was $10,800$: $1,800$ trials for each of the metric and visual models, and $7,200$ trials for the topological model.

The Anderson-Darling test for normality indicated that the percent reached (A=431.01, p<0.001) was distributed normally. An ANOVA by n_{top} did not find a significant difference for the topological model's performance. Without loss of generality, the topological

[1] Vanderbilt University, USA, email: musad.a.haque@vanderbilt.edu
[2] Michigan State University, USA

Table 1. Experimental Design.

Parameter	Values
World	500 pixels × 500 pixels
Body length of an agent (BL)	2 pixels
Number of agents, (N)	$\{50, 100, 200\}$
Number of obstacles, (N_{obs})	$\{5\%N, 10\%N, 20\%N\}$
Radius of repulsion (r_{rep})	$\{5 \times BL, 10 \times BL\}$
Radius of orientation (r_{ori})	$\{1.50 \times r_{rep}, 2 \times r_{rep}\}$
Radius of attraction (r_{att})	$\{1.50 \times r_{ori}, 2 \times r_{ori}\}$
Number of obstacles (N_{obs})	$\{0\%N, 10\%N, 20\%N\}$
Metric range (d_{met})	r_{att}
Topological number (n_{top})	$\{5, 6, 7, 8\}$
Visual range (d_{vis})	Half the diagonal of world
Blindspot (ϕ)	$\pm\pi/3$ radians

trials with $n_{top} = 7$ are used in the reported results.

The topological and visual models had virtually identical mean percent reached (R), as reported in Table 2. The ANOVA found that model type had a significant impact on the percent reached ($F(2,5398)=83.91$, $p<0.001$). A Fisher's LSD test investigated the pair-wise differences. There was no significant difference between the visual and topological models, but the metric model had a significantly lower percent reached compared to the other models.

H_{sg} was partially supported. The topological and visual models outperformed the metric model in reaching the goal area, yet there was no clear difference between the visual and topological models.

5 THE AVOID AN ADVERSARY EXPERIMENT

The swarm is required to avoid a predator-like agent during the *avoid an adversary* task, which is modeled through a repulsive force exerted by the adversary on the swarm agents [1]. The swarm is initially aligned facing the predator, which is the same size as the swarm agents and can occlude the visual communication between agents. The predator and swarm travel toward each other, and when the swarm agents are within r_{att} of the adversary, the predator's repulsive forces affect the swarm agents' heading.

A connected component is defined as the largest collection of agents in which any two agents are either connected directly by a communication link or indirectly via neighbors [3]. The *number of connected components*, CCO, is calculated at the end of a trial, and is 1 at the start of a trial. The hypothesis for this trial is that H_{aa}: $CCO_V < CCO_T < CCO_M$, where the subscripts indicate the communication models. The hypothesis is based on the metric model's limited sensing range. Moreover, fewer stragglers may arise with the visual and topological models, thus decreasing their CCO.

The experimental parameters were identical to the previous experiment, other than N_{obs}, which was set to 0 for this experiment. The total number of trials for the avoid adversary task was $3,600$: 600 trials for the metric and the visual models, and $2,400$ trials for the topological model. Each trial's simulation was set to 200 iterations.

The number of connected components ($A=179.90$, $p<0.001$) was distributed normally according to the Anderson-Darling test. Similar to the prior experiment, n_{top} was set to 7, as the ANOVA found no significant interactions by the topological number. The visual model had the lowest CCO, while metric had the highest, (see Table 3). An ANOVA determined that model type had a significant impact on CCO ($F(2,5398)=1776.23$, $p<0.001$). This metric was significantly different between each of the communication models, as indicated by a Fisher's LSD test. H_{aa} was fully supported.

Table 2. The search for a goal task descriptive statistics by model, with the best means in bold.

Model	Percent Reached		
	Mean	Median	Std. Dev.
Metric	27.68	0.00	41.60
Topological	**39.08**	34.00	31.75
Visual	**41.10**	22.00	42.56

Table 3. The avoid an adversary descriptive statistics by model.

Model	Number of Connected Components		
	Mean	Median	Std. Dev.
Metric	4.46	4.00	2.78
Topological	1.75	2.00	0.79
Visual	**1.35**	1.00	0.58

6 CONCLUSION

The presented research focuses on a general hypothesis that the selection of communications model impacts the swarm's task performance. The general findings demonstrated that there was a significant impact of model type on task performance. The relevance of this outcome is that the intelligence of a remotely deployed swarm is amplified through the deliberate selection of a communication model. A more comprehensive analysis of the performance of the communication models requires considering the effects of pair-wise combinations of communication models with additional independent variables, such as a the number of agents, the number of obstacles, and the radii of interactions. Additional analysis of typical artificial swarm tasks is also necessary to fully support the general hypothesis; however, the presented results provide preliminary evidence that artificial swarm design needs to consider the communication model and task pairing in order to optimize the overall swarm performance.

ACKNOWLEDGEMENTS

This material is based upon research supported by, or in part by, the U.S. Office of Naval Research under award #N000141210987.

REFERENCES

[1] Michele Ballerini, Nicola Cabibbo, Raphael Candelier, Andrea Cavagna, Evaristo Cisbani, Irene Giardina, Vivien Lecomte, Alberto Orlandi, Giorgio Parisi, Andrea Procaccini, Massimiliano Viale, and Vladmiri Zdravkovic, 'Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study', *PNAS*, **105**(4), 1232–1237, (2008).

[2] Iain D. Couzin, Jens Krause, Richard James, Graeme D. Ruxton, and Nigel R. Franks, 'Collective memory and spatial sorting in animal groups', *Journal of Theoretical Biology*, **218**(1), 1–11, (2002).

[3] David Easley and Jon Kleinberg, *Networks, Crowds, and Markets : Reasoning About a Highly Connected World*, Cambridge University Press, New York, NY, 2010.

[4] Andreas Kolling, Phillip Walker, Nilanjan Chakraborty, Katia Sycara, and Michael Lewis, 'Human interaction with robot swarms: A survey', *IEEE Trans. Human-Machine Systems*, **46**(1), 9–26, (2016).

[5] Craig Reynolds, 'Flocks, herds and schools: A distributed behavioral model', *Computer Graphics*, **21**, 25–34, (1987).

[6] Ariana Strandburg-Peshkin, Colin R. Twomey, Nikolai W.F. Bode, Albert B. Kao, Yael Katz, Christos C. Ioannou, Sara B. Rosenthal, Colin J. Torney, Hai Shan Wu, Simon A. Levin, and Iain D. Couzin, 'Visual sensory networks and effective information transfer in animal groups', *Current Biology*, **23**(17), R709–R711, (2013).

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

doi:10.3233/978-1-61499-672-9-1718

Fault Manifestability Verification for Discrete Event Systems

Lina Ye [1] and **Philippe Dague** [2] and **Delphine Longuet** [3] and **Laura Brandán Briones** [4] and **Agnes Madalinski**[5]

Abstract. Fault diagnosis is a crucial and challenging task in the automatic control of complex systems, whose efficiency depends on the diagnosability property of a system. Diagnosability describes the system ability to determine whether a given fault has effectively occurred based on the observations. However, this is a very strong property that requires generally high number of sensors to be satisfied. Consequently, it is not rare that developing a diagnosable system is too expensive. To solve this problem, in this paper, we first define a new system property called manifestability that represents the weakest requirement on faults and observations for having a chance to identify on line fault occurrences and can be verified at design stage. Then, we propose an algorithm with PSPACE complexity to automatically verify it.

1 Introduction

The diagnosability problem has received considerable attention in the literature. Diagnosability describes the system ability to determine with certainty whether a given fault has effectively occurred based on the observations. In a given system, the existence of two infinite behaviors, with the same observations but exactly one containing the considered fault, violates diagnosability. The existing works search for such ambiguous behaviors both in centralized [5, 3] and distributed [4, 7] ways. The most classical method is to construct a structure called twin plant that captures all pairs of observable equivalent behaviors to directly check the existence of such ambiguous pairs. However, in reality, diagnosability is a very strong property that requires generally high number of sensors. Consequently, it is often too expensive to develop a diagnosable system.

To achieve a trade-off between the cost, i.e., the reasonable number of sensors, and the possibility to observe a fault manifestation, we define in this paper a new property called manifestability. This is a property describing the capability of a system to manifest (i.e., be distinguishable from any non faulty behavior) a fault occurrence in at least one context, i.e., one future behavior. This should be analyzed at design stage on the system model. Manifestability is the weakest property to require from the system to have a chance to identify the fault occurrence.

Our contributions in this paper are described as follows: Firstly, we define formally the new manifestability property. Then, we provide a sufficient and necessary condition for manifestability. Finally, we propose an algorithm based on equivalence checking of Finite State Machines (FSMs) with PSPACE complexity.

[1] LRI, CentraleSupélec, France, email: lina.ye@lri.fr
[2] LRI, Univ. Paris-Sud, CNRS, France, email: philippe.dague@lri.fr
[3] LRI, Univ. Paris-Sud, CNRS, France, email: Delphine.Longuet@lri.fr
[4] Universidad Nacional de Córdoba, Argentina
[5] Otto-von-Guericke-University Magdeburg, Germany

2 Manifestability for DESs

We model a Discrete Event System (DES) as a FSM, denoted by $G = (Q, \Sigma, \delta, q^0)$, where Q is the finite set of states, Σ is the finite set of events, $\delta \subseteq Q \times \Sigma \times Q$ is the set of transitions (the same notation will be kept for its natural extension to words of Σ^*), and q^0 is the initial state. The set of events Σ is divided into three disjoint parts: Σ_o is the set of observable events, Σ_u the set of unobservable normal events and Σ_f the set of unobservable fault events. Similar to diagnosability, the manifestability algorithm that we will propose has exponential complexity in the number of fault types. For the sake of reducing this complexity to linear, as in [4, 6], we consider only one fault type at a time but multiple occurrences of faults are allowed.

Given a system model G, its prefix-closed language $L(G)$, which describes both normal and faulty behaviors of the system, is the set of words produced by G: $L(G) = \{s \in \Sigma^* | \exists q \in Q, (q^0, s, q) \in \delta\}$. Those words containing (resp. not containing) F will be denoted by $L_F(G)$ (resp. $L_N(G)$). In the following, we call a word from $L(G)$ a trajectory in the system G and a sequence $q_0\sigma_0 q_1\sigma_1 ...$ a path in G, where $\sigma_0\sigma_1 ...$ is a trajectory in G and we have $\forall i, (q_i, \sigma_i, q_{i+1}) \in \delta$. Given $s \in L(G)$, we denote the post-language of $L(G)$ after s by $L(G)/s$, formally defined as: $L(G)/s = \{t \in \Sigma^* | s.t \in L(G)\}$. The projection of the trajectory s to observable events of G is denoted by $P(s)$. This projection can be extended to a language $L(G)$, i.e., $P(L(G)) = \{P(s) | s \in L(G)\}$. Traditionally, we assume that the system language is always live (any trajectory has a continuation, i.e., is a strict prefix of another trajectory) without unobservable cycle. We will need some infinite objects. So, inspired from the notation of Büchi automata [1], we denote by Σ^ω the set of infinite words on Σ and by $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$ the set of words on Σ, finite or infinite. We define in an obvious way $L^\omega(G)$ (infinite words whose all finite prefixes belong to $L(G)$) and $L^\infty(G)$ and thus infinite trajectories and infinite paths. Particularly, we use $L^\omega_F(G) = L^\omega(G) \cap \Sigma^* F \Sigma^\omega$ for the set of infinite faulty trajectories, and $L^\omega_N(G) = L^\omega(G) \cap (\Sigma \setminus \{F\})^\omega$ for the set of infinite normal trajectories, where $\setminus$ denotes set subtraction. We will use the classical synchronization of two FSMs for which only synchronized events should occur simultaneously, denoted by $G_1 \|_{\Sigma_s} G_2$, where Σ_s is the set of synchronized events.

Definition 1 *(Delay Closure). Given a FSM $G = (Q, \Sigma, \delta, q^0)$, its delay closure with respect to Σ_d, where $\Sigma_d \subseteq \Sigma$, is $\mathsf{C}_{\Sigma_d}(G) = (Q_d, \Sigma_d, \delta_d, q^0)$, where $Q_d = \{q^0\} \cup \{q \in Q \mid \exists s \in \Sigma^*, \exists \sigma \in \Sigma_d, (q^0, s\sigma, q) \in \delta\}$ and $(q, \sigma, q') \in \delta_d$ if $\exists s \in (\Sigma \setminus \Sigma_d)^*, (q, s\sigma, q') \in \delta$.*

Delay closure is to keep all information about events in Σ_d, deleting those not in Σ_d, while keeping the same structure. It will be used to simplify the system model without affecting the result.

Informally speaking, a fault F is diagnosable in a system G if and only if (iff) it can be determined without ambiguity when enough events are observed from G after its occurrence.

Definition 2 ***(Critical Pair)****. A pair of trajectories s, s' is called a critical pair with respect to F, denoted by $s \nsim s'$, iff $s \in L_N^\omega(G), s' \in L_F^\omega(G)$ and $P(s) = P(s')$.*

The existence of a critical pair w.r.t. F violates diagnosability and thus diagnosability verification consists in checking the nonexistence of such a pair. To design a diagnosable system, each faulty trajectory should be distinguished from all normal trajectories, which is most often expensive in terms of sensors required. To reduce such a cost and still make possible to show the fault after enough runs of the system, we now define another much weaker property, manifestability, as follows, where s^F denotes a trajectory ending with F.

Definition 3 ***(Manifestability)****. A fault F is manifestable in G iff*

$$\exists s^F \in L(G), \exists t \in L(G)/s^F,$$
$$\forall p \in L(G),\ P(p) = P(s^F t) \Rightarrow F \in p.$$

F is manifestable iff there exists at least one occurrence s^F in G, there exists at least one extension t of s^F in G, such that every trajectory p that is observable equivalent to $s^F t$ should contain F.

Theorem 1 *A fault F is manifestable in G iff the following condition $\Im$ is satisfied: $\exists s \in L_F^\omega(G), \nexists s' \in L_N^\omega(G)$, such that $s \nsim s'$.*

3 Manifestability Verification

Given a system, manifestability verification consists in checking whether the condition $\Im$ in Theorem 1 is satisfied. Next we show, given a system model, how to construct different structures to obtain $L_F^\omega(G)$, $L_N^\omega(G)$, and critical pairs set. The condition $\Im$ can be checked by using equivalence techniques based on these structures.

Given a system model, the first step that we propose is to construct a structure showing fault information for each state, i.e., whether the fault has effectively occurred up to this state from the initial state.

Definition 4 *(Diagnoser). Given a system G, its diagnoser with respect to a considered fault F is the FSM $D_G = (Q_D, \Sigma_D, \delta_D, q_D^0)$, where: 1) $Q_D \subseteq Q \times \{N, F\}$ is the set of states; 2) $\Sigma_D = \Sigma$ is the set of events; 3) $\delta_D \subseteq Q_D \times \Sigma_D \times Q_D$ is the set of transitions; 4) $q_D^0 = (q^0, N)$ is the initial state.*

The transitions of δ_D are those $((q,\ell), e, (q',\ell'))$, with (q,ℓ) reachable from the initial state q_D^0, such that there is a transition $(q, e, q') \in \delta$, and $\ell' = F$ if $\ell = F \vee e = F$, otherwise $\ell' = N$.

Based on a diagnoser, we define the following two structures to obtain the subparts with only faulty or only normal trajectories.

Definition 5 *(Fault (Refined) Diagnoser). Given a diagnoser D_G, its fault diagnoser is the FSM $D_G^F = (Q_{D^F}, \Sigma_{D^F}, \delta_{D^F}, q_{D^F}^0)$, where: 1) $q_{D^F}^0 = q_D^0$; 2) $Q_{D^F} = \{q_D \in Q_D \mid \exists q'_D = (q, F) \in Q_D, \exists s, s' \in \Sigma_D^*, (q_D^0, s, q_D) \in \delta_D^*, (q_D, s', q'_D) \in \delta_D^*\}$; 3) $\delta_{D^F} = \{(q_D^1, \sigma, q_D^2) \in \delta_D \mid q_D^1, q_D^2 \in Q_{D^F}\}$; 4) $\Sigma_{D^F} = \{\sigma \in \Sigma_D \mid \exists (q_D^1, \sigma, q_D^2) \in \delta_{D^F}\}$. Now the fault refined diagnoser, denoted by D_G^{FR}, is obtained by $D_G^{FR} = \mathsf{C}_{\Sigma_o}(D_G^F)$.*

Definition 6 *(Normal (Refined) Diagnoser). Given a diagnoser D_G, its normal diagnoser is the FSM $D_G^N = (Q_{D^N}, \Sigma_{D^N}, \delta_{D^N}, q_{D^N}^0)$, where: 1) $q_{D^N}^0 = q_D^0$; 2) $Q_{D^N} = \{(q, N) \in Q_D\}$; 3) $\delta_{D^N} = \{(q_D^1, \sigma, q_D^2) \in \delta_D \mid q_D^1, q_D^2 \in Q_{D^N}\}$; 4) $\Sigma_{D^N} = \{\sigma \in \Sigma_D \mid \exists (q_D^1, \sigma, q_D^2) \in \delta_{D^N}\}$. Now the normal refined diagnoser, denoted by D_G^{NR}, is obtained by $D_G^{NR} = \mathsf{C}_{\Sigma_o}(D_G^N)$.*

Now we propose another structure that can capture the set of critical pairs, which can then be used for equivalence checking to examine the manifestability condition $\Im$.

Definition 7 *(Pair Verifier). Given a system G, its pair verifier V_G is obtained by synchronizing the corresponding fault refined diagnoser D_G^{FR} and normal refined diagnoser D_G^{NR} based on the set of observable events, i.e., $V_G = D_G^{FR} \parallel_{\Sigma_o} D_G^{NR}$.*

To construct a pair verifier, we impose that the synchronized events are the whole set of observable events. Only in this way, we can guarantee that the language of the pair verifier is the intersection of the languages of the fault refined diagnoser and that of the normal refined diagnoser. In the pair verifier, each state is composed of two diagnoser states, indicating whether the fault has effectively occurred in the first of the two corresponding trajectories.

Once constructed the pair verifier V_G, we now give our major result as follows, whose proof with associated lemmas are omitted here due to lack of space.

Theorem 2 *Given a system G and its pair verifier V_G, a fault F is manifestable iff $L^\omega(V_G) \neq P(L_F^\omega(G))$.*

To check manifestability, the complexity of different diagnosers constructions is linear. For the pair verifier, we have to synchronize the fault refined diagnoser and the normal refined diagnoser, which is obviously polynomial with the number of system states. To finally check the manifestability, the equivalence checking should be performed, which is already demonstrated to be PSPACE in the literature [2]. Thus, the total complexity of this algorithm is PSPACE.

4 Conclusion and future work

In this paper we addressed the formal verification of manifestability for DESs, inspired from diagnosability checking largely studied in the literature. To bring an alternative to diagnosability analysis, whose satisfaction is very demanding in terms of sensors placement, we defined a new weaker property, manifestability, with a sufficient and necessary condition. Then, we constructed different structures from the system model to be able to check this manifestability condition by using equivalence techniques. One future work is to extend our approach to fault manifestability in at most a fixed number of transitions after the fault occurrence, such as I-diagnosability in [5].

REFERENCES

[1] J.R. Büchi, 'On a decision method in restricted second order arithmetic', *Z. Math. Logik Grundlag. Math*, **6**, 66–92, (1960).

[2] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction To Automata Theory, Languages, and Computation*, Pearson Education, 1979.

[3] S. Jiang, Z. Huang, V. Chandra, and R. Kumar, 'A Polynomial Time Algorithm for Testing Diagnosability of Discrete Event Systems', *Transactions on Automatic Control*, **46**(8), 1318–1321, (2001).

[4] Y. Pencolé, 'Diagnosability Analysis of Distributed Discrete Event Systems', in *Proceedings of the 16th European Conference on Articifial Intelligent (ECAI04)*, pp. 43–47. Nieuwe Hemweg: IOS Press., (2004).

[5] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, 'Diagnosability of Discrete Event System', *Transactions on Automatic Control*, **40**(9), 1555–1575, (1995).

[6] A. Schumann and J. Huang, 'A Scalable Jointree Algorithm for Diagnosability', in *Proceedings of the 23rd American National Conference on Artificial Intelligence (AAAI-08)*, pp. 535–540. Menlo Park, Calif.: AAAI Press., (2008).

[7] L. Ye and P. Dague, 'Diagnosability Analysis of Discrete Event Systems with Autonomous Components', in *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI-10)*, pp. 105–110. Nieuwe Hemweg: IOS Press., (2010).

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

doi:10.3233/978-1-61499-672-9-1720

Using Petri Net Plans for Modeling UAV-UGV Cooperative Landing

Andrea Bertolaso and **Masoume M. Raeissi** and **Alessandro Farinelli** and **Riccardo Muradore** [1]

Abstract. Use of cooperative multi vehicle team including aerial and ground vehicles has been growing rapidly over the last years, ranging from search and rescue to logistics. In this paper, we consider a cooperative landing task problem, where an unmanned aerial vehicle (UAV) must land on an unmanned ground vehicle (UGV) while such ground vehicle is moving in the environment to execute its own mission. To solve this challenging problem we consider the Petri Net Plans (PNPs) framework, an advanced planning specification framework, to effectively use different controllers in different conditions and to monitor the evolution of the system during mission execution so that the best controller is always used even in face of unexpected situations. Empirical simulation results show that our system can properly monitor the joint mission carried out by the UAV/UGV team, hence confirming that the use of a formal planning language significantly helps in the design of such complex scenarios.

1 Introduction

There are a wide variety of applications that take advantage of cooperative multi vehicle team including aerial and ground vehicles. Search and rescue[3], target detection and tracking[6] and mines detection and disposal [1] are a few examples of such applications that benefit from collective behavior of different types of unmanned robots.

In this work we consider a cooperative control scenario, where the UAV/UGV team should operate in tight cooperation to perform a joint task. In particular, here we focus on a cooperative landing scenario, where an unmanned aerial vehicle (UAV) must land on an unmanned ground vehicle (UGV) while such ground vehicle is moving in the environment to execute its own mission. Our goal is for the UAV to perform a fast and safe landing maneuver, hence we propose a strategy where the UAV quickly approaches the UGV and then carefully plans a safe landing trajectory.

A crucial open issue for multi robot systems that perform tight cooperation is to recover from possible failures due to unexpected events. For example, consider a situation where the UAV is initiating the landing maneuver based on the future positions communicated by the UGV. If the UGV must suddenly change its current trajectory (e.g., due to a moving obstacle) the UAV should smoothly adapt its plan to recover from a possible failure.

In this paper we investigate the use of high level languages or team plans [5, 4, 7, 2] to describe and monitor the activities of vehicles during mission execution to achieve the collective behaviors and goals even in face of such unexpected events. Specifically, our focus is on Petri Net Plans (PNPs) framework [7] to specify the collaborative landing task. There are several benefits related to the use of the PNP framework: first it provides a rich graphical representation that helps the designers to create plans with minimal effort, second the generated plans can be monitored during the execution, third PNPs support well-defined structures for handling tight coordination and on-line synchronization in multi robot systems.

Finally We evaluate our approach in V-REP, a realistic simulation environment, using state of the art tools for robots control. Our experiments show that the proposed approach can effectively monitor the cooperative behavior of the two vehicles recovering from possible failures.

[1] Computer Science Department, University of Verona, Italy, email: andrea.bertolaso@studenti.univr.it, masoume.raeissi@univr.it, alessandro.farinelli@univr.it, riccardo.muradore@univr.it

2 UAV/UGV Cooperative Landing Scenario

The problem addressed in this paper is a particular kind of collaboration between heterogeneous autonomous vehicles: the landing of an UAV on an UGV. The collaboration task is composed of three phases:

1. both the UGV and UAV are moving according to their specific and non-cooperative tasks;
2. the UAV approaches the UGV (*flyFar* action using the PNP terminology);
3. the UAV lands on the UGV (*flyClose* action using the PNP terminology).

In Phase 2 the UAV is using its sensing system (e.g. camera) to locate the UGV and plans the faster trajectory to approach the UGV. In this phase the UGV in not aware of the intention of the UAV and so it is continuing its task as in Phase 1. In Phase 3, the UAV is close to the UGV and information are exchanged between them: the UGV is getting aware of the intention of the UAV and so it decreases its velocity and sends to the UAV its planned trajectory to easier the landing. This means that the UGV is still pursuing its objective (e.g. patrolling an area) but in a slower way.

We used JARP to create the Petri Net plan, however any of the available graphical tool that supports pnml (Petri Net Markup Language) file format could be used. The simplified version of the plan is shown in figure 1(b).

Actions name and all external conditions have been defined in the plan. Actions represent robot behaviors, for example in our case the *flyFar* action represents the UAV flying towards the UGV constantly following its position. In addition to the actions, we use interrupt operator, an important structure of PNP, to model action failures and activate recovery procedures based on the occurred condition. Conditions are external events and need to be checked at run time. The possibility to define conditions is a powerful feature that allows to enrich the plan behavior at run time.

The plan execution will be started by initializing the UGV and then the UAV. Both robots will execute the *moveFar* and *flyFar* actions, until UAV gets close to the UGV or decides to landing. The close and far distances are application-dependent. When the UAV is close to the UGV, the *flyFar* action is interrupted and UAV sends the *close* event to the UGV. The UGV *moveFar* action is interrupted as well.

When the vehicles are getting close, UAV's behavior should be changed based on UGV's future position. UAV must be informed of UGV's future position to coordinate their actions. Thus UAV will receive UGV's future position periodically (T seconds). Every time a new future position is sent to the UAV, this new position will interrupt the *flyClose* action so the UAV can recompute flight trajectory in order to follow the new location of UGV.

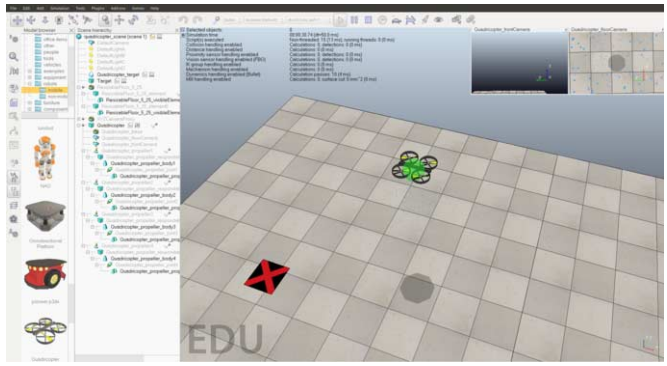
After the plan is designed, we have to hand coding actions and conditions by using ROS [2] actionlib interface and makes them available to the PNPros which connects the PNP executor with ROS. Then PNP executor processes the Petri Net (pnml file described above) and executes the plan within the ROS.

3 Simulation and Evaluation

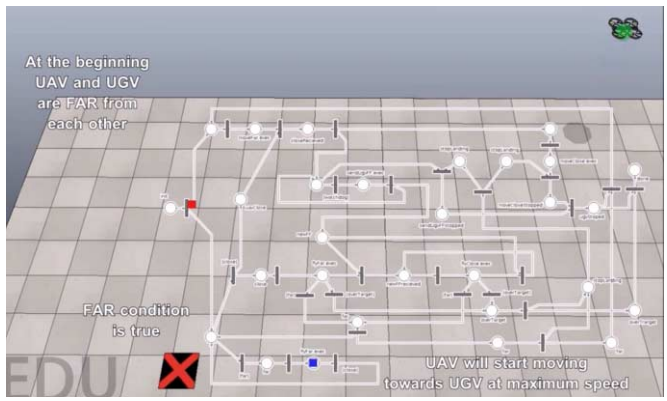
For running the experiments, we create in V-REP a simulation environment containing the UAV and the UGV. The initial position of both vehicles can be chosen arbitrarily in order to obtain different experimental setups. Figure 1(a) shows an initial positions of UAV and UGV in V-REP environment. Communication with V-REP is possible through ROS topics. When the simulation environment and the system that handles the plan are launched, the initial position of UAV and UGV is retrieved from V-REP via ROS topics.

The actual positions of the UAV and UGV are communicated to V-REP during the execution of the plan. The simulation environment will be updated according to the new changes. The whole system keeps on running until a final state in the Petri Net plan is reached. Figure 1(b) shows a snapshot of the simulation when the UAV is flying toward the UGV (*flyFar* action during Phase 2).

A video showing the complete execution of the plan can be seen at the link in the footnote[3]. The video illustrates that the coordination between the two vehicles is not a one-step synchronization action but it is a continuous behavior. The vehicles start far away from each other; then the UAV flies toward the UGV with maximum speed (Phase 2) until the close condition comes true (Phase 3). At this moment, UAV sends an external event to the UGV and the UGV starts sending its next position to the UAV. UGV decreases its speed to make the landing easier. The future position of the UGV is important for the UAV because unexpected events may happens (e.g. obstacles) that prevent the UAV to land on the UGV and so they may get far away again (from Phase 3 to Phase 2). The video also shows the evolution of the simplified version of the Petri Net plan during the simulation in order to better illustrate the behavior of the system. The mission is accomplished when the UAV lands on the UGV: this corresponds to the final state (place) of the plan.



(a) Initializing UAV and UGV in the environment



(b) The Petri Net plan demonstrates the situation when UAV flies toward UGV

Figure 1. V-REP environment setup for simulating the cooperative landing task.

ACKNOWLEDGEMENTS

This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 689341. This work reflects only the authors' view and the EASME is not responsible for any use that may be made of the information it contains.

[2] http://www.ros.org/
[3] https://goo.gl/hiZIKP

REFERENCES

[1] L. Cantelli, M. Mangiameli, C. D. Melita, and G. Muscato, 'Uav/ugv cooperation for surveying operations in humanitarian demining', in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–6, (Oct 2013).

[2] Alessandro Farinelli, Nicoló Marchi, Masoume M. Raeissi, Nathan Brooks, and Paul Scerri, 'A mechanism for smoothly handling human interrupts in team oriented plans', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pp. 377–385, (2015).

[3] M. K. Habib, Y. Baudoin, and F. Nagata, 'Robotics for rescue and risky intervention', in *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pp. 3305–3310, (Nov 2011).

[4] Gal A. Kaminka and Inna Frenkel, 'Flexible teamwork in behavior-based robots', in *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pp. 108–113, (2005).

[5] Milind Tambe, 'Towards flexible teamwork', *Journal of Artificial Intelligence Research*, 83–124, (1997).

[6] H. Yu, R. W. Beard, M. Argyle, and C. Chamberlain, 'Probabilistic path planning for cooperative target tracking using aerial and ground vehicles', in *Proceedings of the 2011 American Control Conference*, pp. 4673–4678, (June 2011).

[7] V.A. Ziparo, L. Iocchi, PedroU. Lima, D. Nardi, and P.F. Palamara, 'Petri net plans', *Autonomous Agents and Multi-Agent Systems*, **23**(3), 344–383, (2011).

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

doi:10.3233/978-1-61499-672-9-1722

Applications of Argumentation: The *SoDA* Methodology

Nikolaos I. Spanoudakis[1] and **Antonis C. Kakas**[2] and **Pavlos Moraitis**[3]

1 Introduction

The area of argumentation is now at a stage that it can benefit from the study of systematic methodologies for building applications of argumentation. Herein, we present such a systematic argumentation software methodology, called *SoDA (Software Development for Argumentation)*, that facilitates the principled modeling of real life problems, and the *Gorgias-B* tool that supports it. *Gorgias-B* builds on the system *Gorgias* that implements the theoretical framework proposed in [1]. *SoDA* and *Gorgias-B* built on the successful use of *Gorgias* during the last ten years by different users for developing real life applications (see http://gorgiasb.tuc.gr/Apps.html).

2 Background

In [1] the authors proposed a preference-based argumentation framework where theories may be composed at different levels. The framework allows to represent arguments as rules whose heads are either alternative options (e.g. actions, decisions) or priorities over rules. An argument attacks (or is a counter argument to) another when they derive a contrary conclusion. These are conflicting arguments. A conflicting argument is admissible if it counter-attacks all the arguments that attack it. Arguments of the first level (i.e. concerning options) have to take along priority (or higher level) arguments and make themselves at least as strong as their counter-arguments.

In applications, we have **options** and **beliefs**. Beliefs refer to properties of the application problem environment. They can be decomposed, although not necessarily, into *Defeasible and Non-Defeasible* beliefs and some of the defeasible beliefs can be designated as **abducible** beliefs. Finally, we can have a **complementary or conflict** relation between the different options of the application.

In a medical access application, possible (simplified) examples of rules at the different levels are (here we are assuming that the parameter variables in the rules take values in their respective types, P is for Patient, D is for Doctor, F is for File, H is for Hospital):

$r_1(D,P,F) : denyAccess(D,P,F) \leftarrow true$
$r_2(D,P,F) : allowAccess(D,P,F) \leftarrow medical(P,F) \wedge worksFor(D,H) \wedge isIn(P,H)$
$p_1(D,P,F) : h\_p(r_1(D,P,F), r_2(D,P,F)) \leftarrow true$
$p_2(D,P,F) : h\_p(r_2(D,P,F), r_1(D,P,F)) \leftarrow treating(D,P)$
$pp_1(D,P,F) : h\_p(p_2(D,P,F), p_1(D,P,F)) \leftarrow true$

The first two rules are **object-level** rules. The next pair of rules are **priority rules** at the *first level*. The last rule is a *second level* priority rule.

If the argumentation theory can provide support both for an option literal and a complement one, then they are both credulously supported. In the case that only one option can be supported then we have a sceptical conclusion.

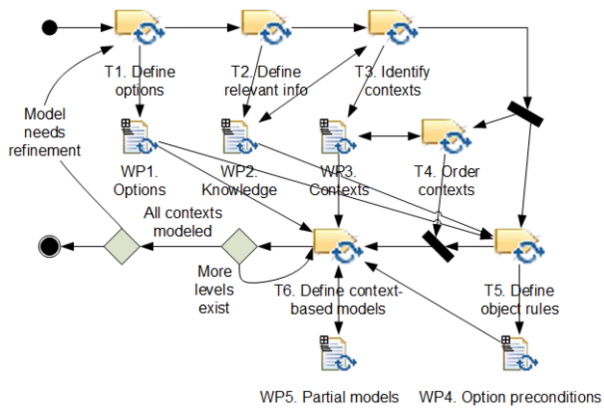


Figure 1. The *SoDA* process depicted using SPEM (Software Process Engineering Metamodel) 2.0 (http://www.omg.org/spec/SPEM/2.0/)

3 *SoDA*: Software Development for Argumentation

The *SoDA* methodology defines a high level process (presented graphically in Figure 1) requiring from the developer to consider questions about the requirements of the problem at various scenarios without the need to consider the underlying software code that will be generated. The Software Engineering Process is defined as a series of tasks (or activities) that produce Work Products (WPs). The different *SoDA* tasks (T) and their input and output WPs are:

T1: This task defines the different **options** of the application problem, including the **conflict relation** between them (written in WP1)
T2: The second task identifies the knowledge needed in order to describe the application environment (written in WP2)
T3: This task aims to separate the information in WP2 into two types: information that always exists for all instances of the problem and information that is **circumstantial**, which may be present in all instances of the problem. Circumstantial predicates are removed from WP2 and inserted in WP3. The next two tasks can be executed in parallel (T4 and T5)
T4: This task aims to sort the circumstantial information from the more general to the more specific application **contexts** in levels, starting from level one (more general contexts). Independent contexts (i.e. when the one is not a refinement of the other) can appear at the same level
T5: This task begins the process of capturing the application requirements. It aims to define for each option, O_i, the different problem environments, i.e. the sets of **preconditions**, C_i, in terms

[1] Technical University of Crete, Greece, email: nikos@science.tuc.gr
[2] University of Cyprus, Cyprus, email: antonis@cs.ucy.ac.cy
[3] LIPADE, Paris Descartes University, France, email: pavlos@mi.parisdescartes.fr

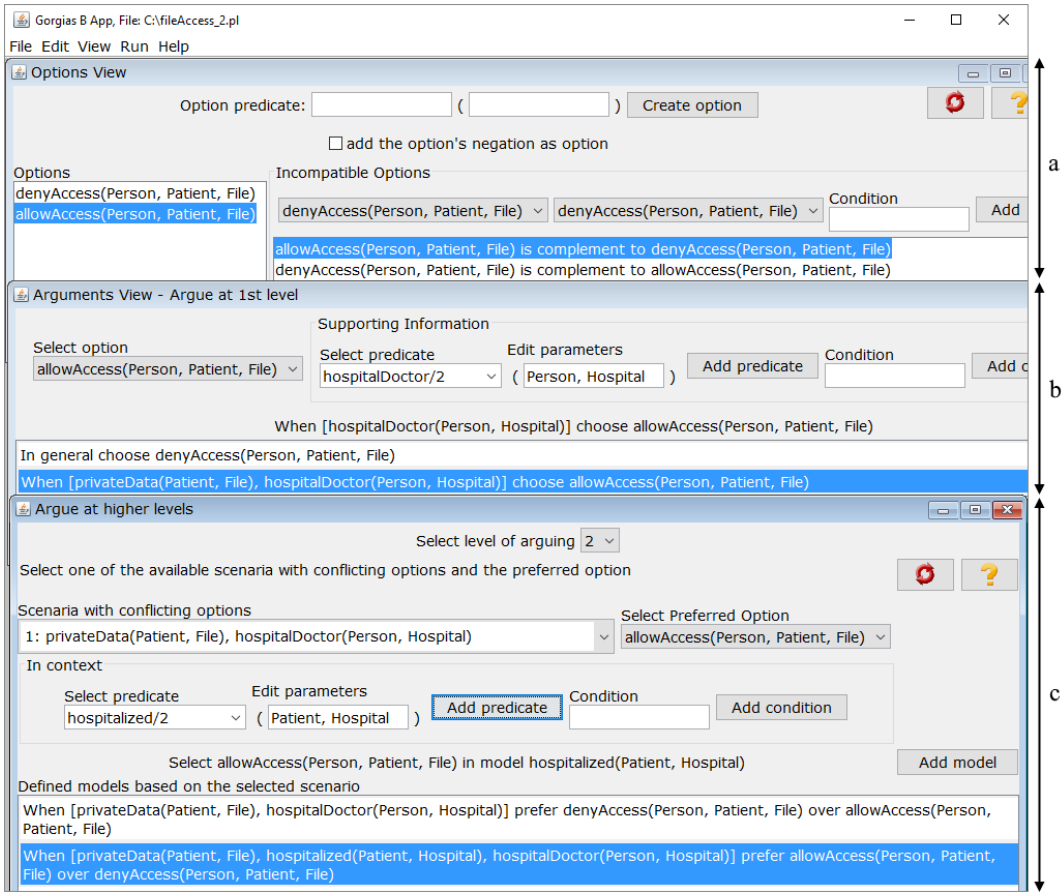


Figure 2. A screenshot of the *Gorgias-B* tool with the options view (a), the arguments view (b) and the argue view (c)

of non-circumstantial predicates appearing in WP2, where the **option is possible**. Its output, WP4, contains all such sets of preconditions

T6: This final task iteratively defines sequences of increasingly more specific **partial models or scenarios** of the world (stored in WP5) and considers how options might win over others. This starts with information from WP4 to precondition the world and iterates getting each time contextual information from the next level in WP3. At each **level** of iteration it defines which option is stronger over another under the more specific contextual information. In the final iteration, the winning options (if they exist) for each partial model are defined without extra information

4 Tool Support for *SoDA* Methodology

We have developed a new tool, *Gorgias-B* [4], to support the development of applications of argumentation under *Gorgias*, following the *SoDA* methodology, in a way that allows users with little or no knowledge of argumentation to model their application domains. Using *Gorgias-B* the user can automatically generate source code in the form of an application argumentation theory in the *Gorgias* framework. Moreover, this theory can be executed under *Gorgias-B* by specifying scenarios of interest and asking which options are credulous or sceptical solutions. The tool returns these together with the admissible arguments that support them. *Gorgias-B* also allows to specify some predicates as *abducible*, and the tool can find scenario conditions under which an option will be a solution.

In the *Options View* (Figure 2(a)), the user defines the different options. Then, the user can edit preconditions (WP4) for options in the *Argument View* (Figure 2(b)). Here the user is building the (object-level) arguments for the various options. The *Argue View* (Figure 2(c)) appears as soon as the user clicks the "Resolve conflicts" button. Here the user selects among scenarios with conflicting options more specialized cases (if they exist) where one of the other option can win. When this happens the contextual information of both previous level scenarios is combined to a new more specific scenario and the user can then repeat to select contexts in the new current level, which is always visible at the top of the dialog window.

[4] http://www.amcl.tuc.gr/gorgiasb

ACKNOWLEDGEMENTS

We acknowledge the collaboration with the Resilient Information Systems Security (RISS) group at Imperial College in the study of the application of SoDA to cyber security and data access and data sharing problems.

REFERENCES

[1] Antonis C. Kakas and Pavlos Moraitis, 'Argumentation based decision making for autonomous agents', in *Proc. Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, Melbourne, Australia*, pp. 883–890, (2003).

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

doi:10.3233/978-1-61499-672-9-1724

Decoupling a Resource Constraint Through Fictitious Play in Multi-Agent Sequential Decision Making

Frits de Nijs[1] and **Matthijs T. J. Spaan**[1] and **Mathijs M. de Weerdt**[1]

Abstract. When multiple independent agents use a limited shared resource, they need to coordinate and thereby their planning problems become coupled. We present a resource assignment strategy that decouples agents using marginal utility cost, allowing them to plan individually. We show that agents converge to an expected cost curve by keeping a history of plans, inspired by fictitious play. This performs slightly better than a state-of-the-art best-response approach and is significantly more scalable than a preallocation Mixed-Integer Linear Programming formulation, providing a good trade-off between performance and quality.

1 INTRODUCTION

When multiple agents must coordinate under a shared resource constraint, individually tractable problems become tightly coupled through the dependency on the resource consumption of all other agents. In problems where agents have the ability to compute and execute their own plans, these agents may be used to decouple the problem into efficiently solvable sub-problems [4].

Resource-constrained agents can be decoupled by preallocating resources a priori. Wu and Durfee [5] present a Mixed-Integer Linear Programming (MILP) formulation to optimally preallocate resources. Unfortunately, preallocating resources still has an exponential complexity, which prevents application to real-world scale problems. To overcome these restrictions, we proposed an on-line conflict resolution approach by planning a best-response policy to the likelihood of successfully executing constrained actions [3]. While this results in efficiently computable policies, the assignment of such a state-independent success probability may be overly pessimistic.

In this paper we propose to look at the marginal utility gained as a consequence of being assigned a resource. By comparing this utility to that of other agents, they can make an informed decision on the distribution of resources. We use this idea to decouple agents by computing a marginal utility cost for the resource. The key insight is that a cost allows agents to compute an expected resource assignment also based on their state. Convergence of the cost function is obtained by keeping a history of expected states, similar to fictitious play.

2 PROBLEM DESCRIPTION

We define Resource Constrained Multi-agent Markov Decision Processes (RC-MMDPs) as an extension of finite horizon MMDPs [2].

Each individual agent i is modeled as a Markov Decision Process specified by tuple $M_i = \langle S_i, A_i, P_i, R_i \rangle$. The current state of an agent is an element $s_{i,j}$ of set S_i containing a finite number of possible states. In any state the agent can choose one of the finite number of actions $a_{i,j}$ contained in set A_i. The transition function $P_i(s_{i,l} \mid s_{i,j}, a_{i,k})$ defines the probability that agent i ends up in state $s_{i,l}$ from state $s_{i,j}$ by choosing action $a_{i,k}$. Agents are rewarded for their choice through reward function $R_i(s_{i,j}, a_{i,k})$ which returns a real-valued utility.

The independent agent problems are coupled through a resource constraint, turning it into an RC-MMDP problem. RC-MMDP problems are specified by tuple $\langle \mathcal{M}, c, L, h \rangle$. Set $\mathcal{M}$ contains the n individual agent problems, $\mathcal{M} = \langle M_1, M_2, \ldots, M_n \rangle$. The binary cost function $c(a_{i,j})$ is set to 1 if action $a_{i,j}$ uses the resource. We require that all agents have an action with $c(a_\emptyset) = 0$ to ensure feasibility of the model. The non-negative resource consumption limit L_t specifies the maximum consumption at any time t in finite horizon h.

Because the agents are cooperative, the goal of the agents is to maximize the sum of individual agent utilities over the entire horizon. A policy $\pi(\mathbf{s}, t)$ specifies for joint state $\mathbf{s} = \langle s_1, s_2, \ldots, s_n \rangle$ at time t which (feasible) joint action $\mathbf{a} = \langle a_1, a_2, \ldots, a_n \rangle$ the agents should take. Action $\mathbf{a}$ is feasible at time t if $c(\mathbf{a}) \leq L_t$, $c(\mathbf{a}) = \sum_{i=1}^{n} c(a_i)$.

The goal of RC-MMDP planning is to compute an optimal policy π^*, which returns the feasible joint action with the highest expected value for every possible joint state and time. We define the expected value of state $\mathbf{s}$ by following policy π as $V_\pi[\mathbf{s}, t]$, with $V_\pi[\mathbf{s}, h] = 0$. Given this, we define the expected value of taking action $\mathbf{a}$ in state $\mathbf{s}$ as

$$Q_\pi[\mathbf{s}, \mathbf{a}, t] = \mathcal{R}(\mathbf{s}, \mathbf{a}) + \sum_{\mathbf{s}' \in \mathcal{S}} \left(\mathcal{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) \cdot V_\pi[\mathbf{s}', t+1] \right). \tag{1}$$

3 MARGINAL UTILITY COST PLANNING

To improve on the preallocation algorithm, we propose to have the agents agree on the marginal utility cost u of the resource. The agents include this cost in their action selection. When for two actions it holds that (for decoupled policy π_i of agent i)

$$\begin{aligned} Q_{\pi_i}[s, a_1, t] > Q_{\pi_i}[s, a_2, t], \text{ and} \\ Q_{\pi_i}[s, a_1, t] - c(a_1) \cdot u < Q_{\pi_i}[s, a_2, t] - c(a_2) \cdot u, \end{aligned} \tag{2}$$

the agent will choose action a_2, even though it prefers a_1 in the unconstrained case. In general we are looking for the marginal utility cost $u_{\mathbf{s},t}$ which makes the sum of resource consumption induced over the preferred actions for joint state $\mathbf{s}$ at time t fit in the resource limit:

$$\begin{aligned} \max_{u_{\mathbf{s},t}} \; & \sum_{i=1}^{n} Q_{\pi_i}[\mathbf{s}_i, a_{i,j}, t] \\ \text{s.t.} \; & \sum_{i=1}^{n} c\left(\arg\max_{a_{i,j} \in A_i} \left(Q_{\pi_i}[\mathbf{s}_i, a_{i,j}, t] - c(a_{i,j}) \cdot u_{\mathbf{s},t} \right) \right) \leq L_t \\ & u_{\mathbf{s},t} \geq 0. \end{aligned} \tag{3}$$

[1] Delft University of Technology, email: f.denijs@tudelft.nl

Because we know there exists an action $c(a_\emptyset) = 0$, we are guaranteed that a feasible cost exists. The cost $u_{\mathbf{s},t}$ can be computed by sorting the expected future marginal utility values of the agents, and assigning the preferred action to each agent until the constraint is reached. The marginal utility of the agent that consumes the last remaining resource is equal to the cost $u_{\mathbf{s},t}$ that prevents overconsumption in state $\mathbf{s}$.

Of course, changing the executed actions of some agents can make their state trajectories deviate substantially from their plans. Therefore, our key idea is that agents coordinate on the expected resource cost $E[u_t]$ at plan time. Because the expected cost depends on the expected joint states that the agents visit, which in turn depends on their policy, we first let agents plan for the unconstrained case where $E[u_t] = 0, \forall t$. The resulting policies are then evaluated to obtain an informed prior over the reachable states. Let a prior over the starting states $p_{i,1}$ be given for each agent. Since the number of reachable states is (typically) exponential in the number of agents, we propose to perform Monte Carlo sampling to obtain an approximation of the probability distribution $p_t(\mathbf{s})$. Given this prior, the expected resource cost subject to the joint policy $\pi = \langle \pi_1, \pi_2, \ldots, \pi_n \rangle$ is

$$E_\pi[u_t] = \sum_{\mathbf{s}} p_t(\mathbf{s}) \cdot u_{\mathbf{s},t}, \tag{4}$$

where $u_{\mathbf{s},t}$ is determined by solving Equation 3. The agents can then re-plan their policies taking into account this resource cost by applying the modified Bellman equation

$$V_{\pi_i}[\mathbf{s}_i, t] = \max_{a_{i,j} \in A_i} \left(Q_{\pi_i}[\mathbf{s}_i, a_{i,j}, t] - c(a_{i,j}) \cdot E_\pi[u_t] \right). \tag{5}$$

The joint policy is derived by planning all agents individually using Value Iteration with this modified Bellman equation. Since each iteration modifies the expected value at time t, the expected cost $E_\pi[u_t]$ also needs to be updated to reflect future values. Therefore, cost $E_\pi[u_t]$ is computed on the basis of the newest $V_\pi[\mathbf{s}, t+1]$, before the Bellman equation is applied for time t.

This process changes where resource constraints restrict agents' actions. Thus, these steps should be repeated until convergence of the expected cost function. It is easy to imagine that the cost function may oscillate between extremes if we only consider the previous prior. Therefore, to ensure convergence, we keep the history of all past samples, inspired by fictitious play [1]. Each prior can be seen as the adversary 'nature' performing her actions as a consequence of our choices. By remembering all past plays, eventually the full strategy of nature is obtained. Thus, let p^k be the probability distribution over states in iteration (or play) k, then we maintain the set $\mathbb{P} = \langle p^1, p^2, \ldots, p^k \rangle$, and compute the expected cost as

$$E_\pi[u_t] = \sum_{j=1}^{k} \sum_{\mathbf{s} \in p_t^j} \frac{p_t^j(\mathbf{s})}{k} \cdot u_{\mathbf{s},t}. \tag{6}$$

4 EMPIRICAL EVALUATION

To evaluate the performance of this algorithm we compare it against an optimal preallocation MILP [5] and our Best-response planner [3] on an energy-consumption planning problem. In this setting a population of electric heaters must be controlled to keep the aggregate consumption below a power constraint, while satisfying consumers' heat demands. The power constraint may arise due to fluctuating supply of renewable sources like wind or solar. In the experiments we measure the time to compute a policy for 4 agents, and its quality. The Fictitious Play and Best-response algorithms are set to perform at most 10 iterations, computing 1000 priors each iteration.

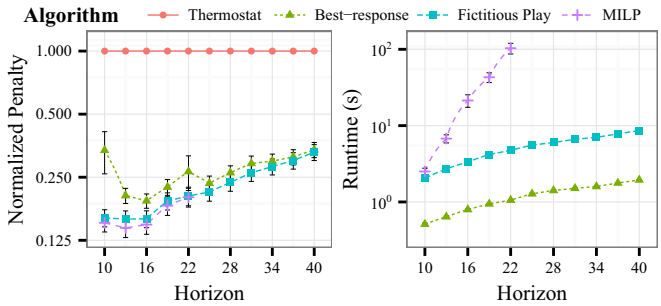


Figure 1. Algorithm performance for increasing horizon: policy quality normalized to the thermostat policy (left), and wall-clock computation time (right). Both plots on a log scale, lower values are better.

Figure 1 presents the mean and standard error of both runtime and policy quality. The policy quality metric penalizes the total amount of deviation of the current temperature from the set-point temperature. The quality is normalized to the myopic strategy of using thermostat controllers with an on-line prioritized load-shedding system to keep the resource demand below the limit.

Based on the MILP formulation, we expect that a linear increase in the length of the horizon results in an exponential growth of the runtime. We observe this exponential scaling in the right plot; several instances of $h = 22$ could not be solved within 30 minutes. The other algorithms have polynomial complexity, and are able to solve each instance within at most 10 seconds. Nevertheless, the policies found by Fictitious Play are almost as good as the MILP policies, and significantly better than Best-response for short horizon instances.

5 CONCLUSIONS AND FUTURE WORK

This paper introduces a decoupling algorithm for multi-agent planning problems under hard resource constraints based on fictitious play. The algorithm computes a time-dependent cost for resources which is used to decouple individual policies so that they can be computed in polynomial time. We compared against two state-of-the-art approaches, and found that the fictitious play algorithm produces policies which are not significantly worse than an optimal preallocation decoupling while requiring exponentially less runtime.

For future work we intend to adapt the fictitious play algorithm to handle stochastic resource levels and multiple resources.

ACKNOWLEDGEMENTS

Support of this research by network company Alliander is gratefully acknowledged.

REFERENCES

[1] U. Berger, 'Brown's original fictitious play', *Journal of Economic Theory*, **135**(1), 572–578, (2007).
[2] C. Boutilier, 'Planning, Learning and Coordination in Multiagent Decision Processes', in *TARK*, pp. 195–210, (1996).
[3] F. de Nijs, M. T. J. Spaan, and M. M. de Weerdt, 'Best-Response Planning of Thermostatically Controlled Loads under Power Constraints', in *AAAI*, pp. 615–621, (2015).
[4] F. A. Oliehoek, S. J. Witwicki, and L. P. Kaelbling, 'Influence-Based Abstraction for Multiagent Systems', in *AAAI*, pp. 1422–1428, (2012).
[5] J. Wu and E. H. Durfee, 'Resource-Driven Mission-Phasing Techniques for Constrained Agents in Stochastic Environments', *JAIR*, **38**, 415–473, (2010).

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

doi:10.3233/978-1-61499-672-9-1726

GPU-Accelerated Value Iteration for the Computation of Reachability Probabilities in MDPs

Zhimin Wu[1] and **Ernst Moritz Hahn**[2] and **Akın Günay**[1] and **Lijun Zhang**[2] and **Yang Liu**[1]

1 INTRODUCTION

Computation of reachability probabilities is an important subroutine for determining approximately optimal policies of MDPs [3]. Value iteration (VI) [2] is a well-known method to compute these values. However, sequential implementation of VI is computationally expensive both in terms of time and memory. Hence, we propose a highly parallel version of VI to solve general MDPs utilizing the GPU, which is widely used in the recent years to accelerate execution performance of various computational methods in many areas [1, 7, 6]. Our approach explores algebraic features (e.g., matrix structure) of MDPs, and uses action-based matrices to achieve massive parallelism for efficiency. We empirically evaluate our approach on several case studies. Our results show that we can achieve up to 10X$\sim$ speedup compared to sequential VI, and outperform topological value iteration (TVI) [4] in most of the cases. Particularly, for MDPs which do not contain strongly connected components (SCCs) with more than one state, or which contain a small number of large SCCs, our approach achieves up to 17X speedup compared to TVI.

Our main contributions are: (1) We take advantage of the algebraic structure of MDPs to define action-based matrices and corresponding data structures for efficient parallel computation of reachability probabilities on GPUs. (2) We develop an efficient parallel VI algorithm for computing reachability probabilities that utilizes features of modern GPUs, e.g., dynamic parallelism and memory hierarchy.

2 BACKGROUND AND RELATED WORK

An *MDP* is a tuple $M = (S, s_{init}, Act, P, R)$, where S is a finite set of *states*, $s_{init} \in S$ is the *initial state*, and Act is a finite set of *actions*. The (partial) *transition probability function* $P\colon S \times Act \mapsto Dist(S)$, where $Dist(S)$ is the set of discrete probability distributions over the set S, assigns probability distributions to combinations of states and actions. The *reward function* $R\colon S \times Act \mapsto Dist(R)$ assigns a numeric reward to each state/action pair. By $Act(s) = Dom(P(s,\cdot))$ we denote the actions that are *activated* in state s. We require $|Act(s)| \geq 1$ for every $s \in S$.

Given an MDP M, we are interested in computing the minimal (maximal) probability to reach a set of target states $T \subseteq S$ in an infinite horizon, formally: $P_{\min}(s,T) = \inf_{\alpha \in Adv} Prob^{\alpha}(s,T)$. Adv is the set of all *schedulers*, which choose the action to be performed in a state depending on the sequence of states and actions seen so far. $Prob^{\alpha}(s,T)$ is the probability of reaching T when starting from state s and following the scheduler α. The computation process is defined by the following equation:

$$x_s^n = \min_{\alpha \in Act(s)} \sum_{s' \in S} P(s' \mid s,a) \cdot x_{s'}^{(n-1)} \text{ for } s \notin T, n > 0 \quad (1)$$

Value iteration [2] is a general dynamic programming method to solve MDPs. It is an iterative computation process to update the value function of every state, which follows Equation 1 and terminates when it satisfies a convergence criterion.

There are some existing approaches which optimize VI using graphical features of MDPs. From these approaches, TVI [4] is the most relevant one to our approach. TVI utilizes the SCC structure of an MDP to construct an acyclic MDP for performing VI backups in the best order, and to only perform them when necessary. While TVI is based on the structure of SCCs in MDPs, our approach utilizes the algebraic features of MDPs that are related to the representation matrix of MDPs and the matrix-vector multiplication during the Bellman backup process. In addition, our approach is independent from the structure of SCCs, which may affect TVI's efficiency.

GPUs have several advantages over CPUs such as high memory bandwidth, computation capability, and massive parallelism. To the best of our knowledge, our approach is novel both in terms of fully utilizing the algebraic features of MDPs and parallel computing techniques in GPU to improve efficiency of VI for the computation of reachability probabilities.

3 GPU ACCELERATED VALUE ITERATION

We build a GPU accelerated parallel VI that can significantly improve the efficiency compared to the sequential VI by taking advantage of the algebraic structure (matrix) of MDPs and the matrix operation involved in the Bellman backup process.

In sequential VI, the complete state space should be backed up in each iteration, which requires the exploration of all states sequentially. More specifically, the term $\sum_{s' \in S} P(s' \mid s,a) \cdot x_{s'}^{(n-1)}$ in Equation 1 requires the sequential VI to compute the reachability probabilities for each state by exploring each enabled action and reachable states. This exploration process creates a bottleneck for sequential VI, since the time required for the exploration process grows exponentially with respect to the state space of an MDP. Hence, if we can accelerate the exploration process, we can significantly improve VI's efficiency.

As we show in Figure 1, an MDP can be represented using a matrix structure $\mathcal{M}$. Each row involves a set of sub-vectors, where each sub-vector represents the immediate reacability probabilities of the states with respect to actions. In VI, the calculation of Equation 1 updates x_i by selecting the action a that maximize/minimize the vector to vector multiplication $Succ_{a_i} \times \mathcal{X}$. This computation is independent in each state. Furthermore, we collect the vector to vector multiplication of the complete state space together, and then divide by the ac-

[1] Nanyang Technological University, Singapore
[2] Institute of Software, Chinese Academy of Sciences, China

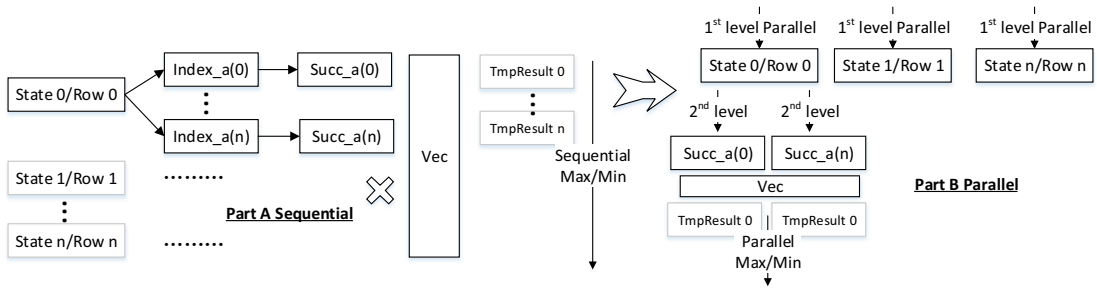


Figure 1. Parallelization

tions. We can find in global perspective that computation is indeed a substantial amount of synchronized sub-matrices to vector multiplications. The sub-matrices are the matrices representation of MDPs by actions. Thus, we define the *Action-based Matrices for MDP*.

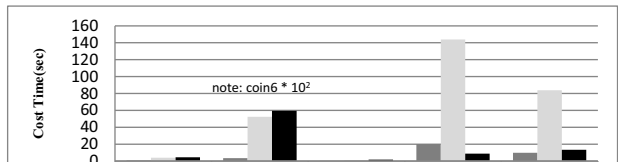
Definition 1 ***Action-based Matrices*** *Given a MDP* $M = (S, s_{init}, Act, P, R)$ *and the representation matrix* $\mathcal{M}$*, for each action* $a \in Act$*, an action-based matrix is a tuple* $(\mathcal{M}_a = \{S_a, S'_a, P(S' \mid S, a))$*, where* S_a *is the set of states in which action* a *is activated,* S'_a *is the set of states reached via action* a *from states in* S_a*, and* $P(S' \mid S, a) \rightarrow (0, 1]$ *is the probability array.*

Each $\mathcal{M}_a$ is a m_a by m_a matrix, where $m_a = |S_a|$. Intuitively, an $\mathcal{M}_a$ represents the transition relationship of the states in S_a with respect to action a. Using the action-based matrices, the value iteration process can be transformed into several interleaving action-based matrix to vector multiplications and subsequent minimisation, where each action-based matrix to vector multiplication is an independent computation. Hence, the action-based matrices and the described partitioning of matrix operations allow us to develop an efficient parallel VI that works well on GPUs. We also design compact data structures and parallel convergence detection for our approach.

4 IMPLEMENTATION AND EXPERIMENTS

We evaluate the performance of our approach by comparing it with VI and TVI implementations of Dai et al. [4]. We implemented our approach in CUDA C. We conducted our experiments on a computer with two Intel(R) Xeon(R) CPU E5-2670, 2.60GHz, 16GB RAM and a Geforce Titan Black GPU. We set the parallelism to 512 threads in one block, which can reach 100% occupancy per multiprocessor according to the CUDA Occupancy Calculator [5].

The results are shown in Figure 2. Our approach achieves around 10X$\sim$ speed up comparing to VI in MDPs with different structure. It can be observed from Figure 2 that TVI performs slightly worse with MDPs that have large number of small SCCs, since under this condition, TVI cannot reduce the backup times significantly, and the SCC detection brings additional cost. This situation is also addressed by Dai *et al.* [4]. We can see our approach can achieve up to 17X speedup compared to TVI under this condition. For the layered MDPs, we consider two strong layered MDPs. *Layered1* only has one transition between any two SCCs. The initial state is a state in the first layer and the goal state is an end state in the last layer. *Layered2* has the same size with *Layered1*. But the number of layers is considerably less than Layered1. The experiment results show that for *Layered1*, TVI outperforms our GPU accelerated VI slightly due to the large number of layers. But with *Layered2*, our approach outperforms TVI with around 1.5X speedup since the number of layers



	csma3	coin6	rabin4	layer1	layer2
GPU-VI	0.58	3.45	0.1	20.1	9.79
VI	3.9	52.29	1.29	143.8	83.8
TVI	4.5	59.6	1.79	8.8	13.4

Figure 2. Performance Evaluation

decreases considerably. Both our approach and TVI outperforms the VI for these two MDPs. In conclusion, our approach's performance is substantially better compared to VI for MDPs with all types of structures. We also achieves considerable speedup compared to TVI in MDPs which has small number of large SCCs or only SCCs with just one state. Although our approach performs slightly worse than TVI in MDPs with a deep layer, we can still conclude from the results that our approach can be more general to a wide range of MDPs.

5 CONCLUSION

We presented a novel GPU accelerated parallel value iteration approach for the efficient computation of reachability probabilities of MDPs. The main idea of our approach is to utilize the algebraic features of MDPs to divide the computation process of reachability probabilities into partitions, which can be computed in a massively parallel manner with an efficient parallelization granularity on GPUs. Our evaluation shows that we achieve 10X$\sim$ speedup compared to sequential, and up to 17X speedup compared to TVI.

References

[1] Ron Bekkerman, Mikhail Bilenko, and John Langford, *Scaling up machine learning: Parallel and distributed approaches*, Cambridge University Press, 2011.
[2] Richard Bellman, 'Dynamic programming and lagrange multipliers', *PNAS*, **42**(10), 767, (1956).
[3] Craig Boutilier, Thomas Dean, and Steve Hanks, 'Decision-theoretic planning: Structural assumptions and computational leverage', *JAIR*, **11**, 1–94, (1999).
[4] Peng Dai, Daniel S Weld, Judy Goldsmith, et al., 'Topological value iteration algorithms', *JAIR*, 181–209, (2011).
[5] CUDA NVIDIA, 'Gpu occupancy calculator', *CUDA SDK*, (2015).
[6] Anton Wijs, 'Gpu accelerated strong and branching bisimilarity checking', in *TACAS*, 368–383, Springer, (2015).
[7] Zhimin Wu, Yang Liu, Jun Sun, Jianqi Shi, and Shengchao Qin, 'Gpu accelerated on-the-fly reachability checking', in *ICECCS*, pp. 100–109, (2015).

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

doi:10.3233/978-1-61499-672-9-1728

Sets of Contrasting Rules to Identify Trigger Factors

Marharyta Aleksandrova [1 2] and **Armelle Brun** [1] and **Oleg Chertov** [2] and **Anne Boyer** [1]

Abstract. In this paper we introduce a new pattern, referred to as "set of contrasting rules". The main originality of this pattern is that it allows to easily identify trigger factors: factors that can bring some event state changes. In real applications this pattern can thus be used to influence the values of some attributes, what was shown through the experiments conducted on a real dataset of census data.

1 INTRODUCTION

As trigger factors we understand those factors that explain or that can bring some events or system state changes. Researches from different domains like medicine [9, 7], sustainable entrepreneurship [4], finance [3], etc. are interested in the identification of trigger factors, mainly through data analysis. There exists a wide variety of data analysis methods. Data mining techniques are of particular interest, as they were designed to discover interesting and unpredictable patterns and interconnections [5], contrary to statistical methods, that are mainly used to check the validity of predefined hypotheses [6].

One of the very popular data mining techniques is association rules mining, where an association rule is a pattern of the form "if X then Y". In this paper we introduce a new type of pattern, referred to as "set of contrasting rules". This pattern is made up of a set of association rules, which highlight the differences between different groups of the dataset. "Set of contrasting rules" patterns have several advantages: 1) by identifying and grouping highly informative rules they allow to structure the huge set of association rules; 2) as these patterns highlight the differences between data groups through the introduction of the notions of varying and invariant attributes, these differences can be interpreted as trigger factors. To the best of our knowledge, no approach in the literature allows to identify such trigger factors directly from the patterns or from rules.

2 A NEW PATTERN "SET OF CONTRASTING RULES"

Let D be a dataset defined on a set of n attributes $\left\{A^1, A^2, \ldots, A^n\right\}$. For each attribute A^j there is a set of possible values. We will call each pair $\{attribute, value\}$ an item. A set X of items is called an itemset. By $supp_D(X)$ we denote the support of the itemset X in the dataset D. The support value is calculated using the following formula $supp_D(X) = \frac{count_D(X)}{|D|}$, where $count_D(X)$ is the number of elements in D containing X and $|D|$ is the total number of elements in D.

An association rule is an induction rule of the form $X \rightarrow Y$, where X and Y are itemsets and $X \cap Y = \emptyset$ [1]. X is referred to as antecedent and Y as consequent. The support of the rule $X \rightarrow Y$ in D is calculated as $supp_D(X \rightarrow Y) = supp_D(X \cup Y)$ and its confidence as $conf_D(X \rightarrow Y) = \frac{supp_D(X \cup Y)}{supp_D(X)}$.

Assume that all elements in the dataset D are organized into k mutually exclusive groups $G_1, G_2, \ldots, G_k$ with $G_1 \cup G_2 \cup \ldots \cup G_k = D$ and $G_i \cap G_j = \emptyset, \forall i \neq j$. Let the group of an element (its class) be specified by the value of the target attribute A^G with possible values $g_1, g_2, \ldots, g_k$.

The definition of a "set of contrasting rules" pattern relies on the introduction of two new types of attributes: varying attributes and invariant attributes. An attribute is considered to be varying if its value can be changed externally to the system within the specified application task, and invariant otherwise. For example, when analysing census data the attribute $income\_level$ can be considered as varying if, for instance, the government can provide with financial assistance to the citizens. At the opposite the value of the parameter $ancestry$ can not be changed, that is it belongs to the set of invariant attributes. Thereby we divide the set of all attributes (except A^G) into two subsets: the set of varying attributes and the set of invariant attributes.

Definition 1. For a specified parameter α ($\alpha > 0.5$), a set of rules $R_1, R_2, \ldots, R_k$ is called a set of α-contrasting rules if:

1. $conf(R_1) \geq \alpha \;\&\; conf(R_2) \geq \alpha \;\&\; \ldots \;\&\; conf(R_k) \geq \alpha$;
2. in each rule, the itemsets of the consequents are formed of only one attribute: the target attribute A^G, with different values of A^G in the set;
3. the antecedents of the rules are made up of the same attributes, within which there is at least one varying and one invariant attribute;
4. the values of all invariant attributes are the same for all rules;
5. at least one varying attribute has different values in the set of rules.

The varying attributes with different values in the pair of contrasting rules define the trigger factors: they can influence the move of the elements from one group to another. The invariant attributes and those varying attributes having the same values in the pair of contrasting rules specify the application subset that is the subset of elements, that can be influenced by these trigger factors. In real applications, the "set of contrasting rules" pattern can be used to solve a wide range of tasks depending on the underlying meaning of the attributes. An example of this pattern usage will be presented in the next section.

3 EXPERIMENTAL RESULTS

The goal of the experiments conducted here is to show to what extent "set of contrasting rules" patterns allow to automatically discover trigger factors. The experiments are conducted on the California 2000 census dataset [8], which contains records of 610,369 family households on more than 100 attributes. For mining association

[1] Université de Lorraine - LORIA, Campus Scientifique, 54506 Vandoeuvre les Nancy, France, email: {marharyta.aleksandrova,armelle.brun,anne.boyer}@loria.fr
[2] National Technical University of Ukraine Kyiv Polytechnic Institute, 37, Prospect Peremohy, 03056, Kyiv, Ukraine, email: chertov@i.ua

Table 1. Some chosen "sets of contrasting rules" patterns, p =10,000$

#	Antecedent		Cons.	Conf
	Subgroup	Trigger Factors	Child=	
1	**HAge=]24,27]** & **WAge=]22,25]** & **WEdu=school** & HIncome=]1p,2p]	Vehicle=1	YES	0.71
		Vehicle=0	NO	0.70
2	**HAge=]24,27]** & **WAge=]22,25]** &**WEdu=noCollege** & HouseOwn=no	HIncome=]2p,4p] & HouseType=Det	YES	0.73
		HIncome=]0,1p] & HouseType=Apart	NO	0.70
3	**HAge=]30,31]** & **WAge=]28,30]** & **HAnc=OtherAmerican**	HIncome=]2p,4p]	YES	0.72
		HIncome=]1p,2p]	NO	0.86
4	**HAge=]30,31]** & **WAge=]28,30]** &**WAnc=Latino**	HouseOwn=yes & HouseType=Det	YES	0.71
		HouseOwn=no & HouseType=Apart	NO	0.75

rules we used Apriori algorithm [2]. The grouping of the rules and the formation of the "set of contrasting rules" patters is done automatically through the analysis of the rules obtained with the algorithm Apriori (post-processing).

The target attribute, which we focus on, is the one related to the children in the family. Our goal here is to *identify which factors can influence the human desire to give birth to a baby*. In order to find an answer to the question that formulates our experimental goal, we divide the dataset into two contrasting groups G_1 and G_2. G_1 is made up of families (elements) with one or two children aged from 0 to 2 years. G_2 contains families without any children. Such a definition of contrasting groups lets us track the change of the family state from childless to a family with small children.

Some "sets of contrasting rules" derived from the dataset are presented in the Table 1. The antecedents of the rules in the patterns are given in the second and third columns of the table. As stated in the previous section, the antecedents of contrasting rules have a common part that specifies the subgroup of the elements; it is presented in the second column with the invariant attributes given in bold. The antecedents have also another part that differs in the values of varying attributes (this part represents the trigger factors); it is presented in the third column of the table. The fourth column indicates the value of the consequent (the attribute $Child$) of each rule in our patterns, and the fifth column reveals the confidence value of the corresponding rules.

When analyzing the sets of contrasting rules given in Table 1, we can note that it can correspond to very precise recommendations for specific subgroups of elements in the dataset. For example, let us look on the first pattern. It indicates that if we provide young families (husband age $HAge =]24, 27]$ and wife age $WAge =]22, 25]$) in which the wife's education level is $WEdu = school$ and husband's income is in the range $]10000, 20000]$ with a vehicle, then with a high probability (71%) they will decide to have a baby. However, analyzing the second pattern formed for the families of the same age group where the wife has started the college but did not finish education there ($WEdu = noCollege$) and that do not have their own house ($HouseOwn = no$), it is not the number of vehicles that can trigger a child birth, but rather the combination of the type of the house (it should be changed from 'apartment' $HouseType = Apart$ to 'detached house' $HouseType = Det$) and the increase of the income level. Moving to another age group ($HAge =]30, 31]$ and $WAge =]28, 30]$) we can see that here the subgroups are formed basing on the ancestry of parents. If husband ancestry ($HAnc$) is *'OtherAmerican'* then the increase of income level can trigger a birth of a baby (see pattern 3). But if wife ancestry ($WAnc$) is *'Latino'*, then the family should be provided with an own detached house (pattern 4). We can see that the presented "sets of contrasting rules" form the subgroups and trigger factors that are meaningfully different. This proves the ability of the proposed pattern to extract valuable knowledge from the dataset.

4 CONCLUSION

In this paper we introduced a new pattern "set of contrasting rules". This pattern has the characteristic of being made up of several rules. It is formed with the aim to directly identify specific knowledge: trigger factors, through the introduction of the notions of invariant and varying attributes. A trigger factor is a factor that can bring some event state changes (in our case it moves elements from one group of the dataset to another). On the application level, this pattern can be interpreted as a way to search the levers of influence, which can force or trigger this move. This characteristic is the key point of the proposed pattern, and according to our knowledge, there are no other patterns in the literature, dedicated to this goal.

We showed on a real dataset of census data that trigger factors are actually identified, and that they can be easily interpreted and used to reach the desired objective. In our experiments the objective was to find factors that can influence the birth rate of families. The analysis of the factors identified through the conducted experiments showed that they are meaningful and can actually have such an influence.

REFERENCES

[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami, 'Mining association rules between sets of items in large databases', *ACM SIGMOD Record*, **22**(2), 207–216, (1993).

[2] Rakesh Agrawal, Ramakrishnan Srikant, et al., 'Fast algorithms for mining association rules', in *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pp. 487–499, (1994).

[3] H Kent Baker, E Theodore Veit, and Gary E Powell, 'Factors influencing dividend policy decisions of nasdaq firms', *Financial Review*, **36**(3), 19–38, (2001).

[4] Progress Choongo, Elco Van Burg, Leo J Paas, and Enno Masurel, 'Factors influencing the identification of sustainable opportunities by smes: Empirical evidence from zambia', *Sustainability*, **8**(1), 81, (2016).

[5] Jiawei Han, Micheline Kamber, and Jian Pei, *Data mining: concepts and techniques*, Elsevier, 2011.

[6] David J Hand, 'Statistics and data mining: intersecting disciplines', *ACM SIGKDD Explorations Newsletter*, **1**(1), 16–19, (1999).

[7] Anders Hougaard, Faisal Amin, Anne Werner Hauge, Messoud Ashina, and Jes Olesen, 'Provocation of migraine with aura using natural trigger factors', *Neurology*, **80**(5), 428–431, (2013).

[8] "u.s. census 2000. 5-percent public use microdata sample files". http://www.census.gov/Press-Release/www/2003/PUMS5.html, 2000.

[9] Jorunn Sundgot-Borgen, 'Risk and trigger factors for the development of eating disorders in female elite athletes', *Medicine & Science in Sports & Exercise*, **26**(4), 414–419, (1994).

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

doi:10.3233/978-1-61499-672-9-1730

Link Prediction by Incidence Matrix Factorization

Sho Yokoi[1] and **Hiroshi Kajino**[2] and **Hisashi Kashima**[3]

Abstract. Link prediction suffers from the data sparsity problem. This paper presents and validates our hypothesis that, for sparse networks, incidence matrix factorization (IMF) could perform better than adjacency matrix factorization (AMF), which has been used in many previous studies. A key observation supporting the hypothesis is that IMF models a partially-observed graph more accurately than AMF. A technical challenge for validating our hypothesis is that, unlike AMF approach, there does not exist an obvious method to make predictions using a factorized incidence matrix. To this end, we newly develop an optimization-based link prediction method adopting IMF. We have conducted thorough experiments using synthetic and real-world datasets to investigate the relationship between the sparsity of a network and the performance of the aforementioned two methods. The experimental results show that IMF performs better than AMF as networks become sparser, which strongly validates our hypothesis.

1 Introduction

Link prediction attempts to predict missing links based on other observed links and attributes of nodes [6, 2, 10, 13]. We focus on link prediction based on a graph structure, which is formulated as follows: given a partially-observed graph $G=(\mathcal{V},\mathcal{E}_\mathrm{P})$ with the set of nodes $\mathcal{V}$ and the set of *positive links* (observed links) $\mathcal{E}_\mathrm{P}\subset\mathcal{V}\times\mathcal{V}$, its goal is to learn a scoring function $s:\mathcal{V}\times\mathcal{V}\to\mathbb{R}$ to predict a new link on an *unlabeled pair of nodes* in a set $\mathcal{E}_\mathrm{U}:=(\mathcal{V}\times\mathcal{V})\setminus\mathcal{E}_\mathrm{P}$.

As pointed out by many researchers, one of the central issues in link prediction is the *sparsity* of positive links [5, 12, 9]. Our idea to counter the problem is to employ incidence matrix factorization (**IMF**) as a building block of a link prediction method, instead of adjacency matrix factorization (**AMF**), which has been used in various previous studies [8, 1, 7, 11, 4]. A key observation supporting the idea is that IMF can model a partially-observed graph more accurately than AMF.

First of all, We briefly introduce the previous AMF-based approach (Fig. 1 [TOP]). Given a partially-observed graph $G=(\mathcal{V},\mathcal{E}_\mathrm{P})$, AMF learns latent feature vectors $\{\boldsymbol{x}_k\}_{v_k\in\mathcal{V}}$ of nodes using both positive links and unlabeled node pairs such that $\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle\approx 1$ if $(v_i,v_j)\in\mathcal{E}_\mathrm{P}$, 0 otherwise holds in its simplest instantiation. This modeling has a little flaw. Let us consider a pair of nodes (v_i,v_j) that is not linked in a partially-observed graph but is actually positive in its fully-observed graph. In the ideal case, latent vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ obtained from the fully-observed graph satisfy $\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle\approx 1$, while those obtained from the partially-observed graph satisfy $\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle\approx 0$. As the observed part of a graph becomes sparser, the number of such node pairs will increase, and therefore, this inconsistency issue can lead to the poor performance. On contrary, IMF can avoid the inconsistency issue because it learns a model by utilizing only positive links. IMF learns latent feature vectors of nodes $\{\boldsymbol{x}_k\}_{v_k\in\mathcal{V}}$ and those of positive links $\{\boldsymbol{y}_l\}_{e_l\in\mathcal{E}_\mathrm{P}}$ such that $\langle\boldsymbol{x}_i,\boldsymbol{y}_j\rangle\approx 1$ if $v_i\in e_j$, 0 otherwise holds in its simplest instantiation. Since this modeling does not utilize unlabeled node pairs, the model obtained from a partially-observed graph is consistent with that obtained from its fully-observed one; therefore, the performance of IMF is expected to be robust to the sparsity of a graph. In this light, we arrive at the hypothesis that IMF can counter the sparsity problem better than AMF.

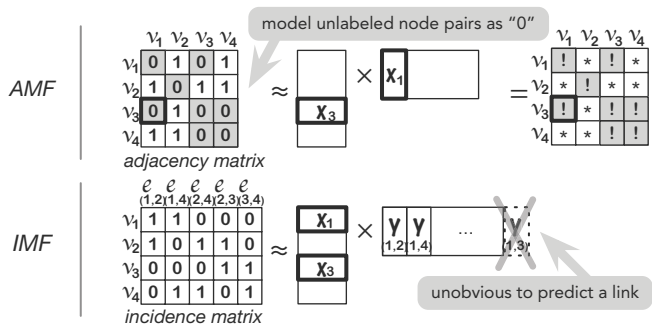


Figure 1. [TOP] Link prediction using AMF. [BOTTOM] It is not trivial to predict a link using IMF. We learn a latent vector $\boldsymbol{y}_{(i,j)}$ for any observed link $e_{(i,j)}\in\mathcal{E}_\mathrm{P}$ in addition to a latent vector $\boldsymbol{x}_k$ for any node. Predicting a link between v_1 and v_3 requires a latent vector of the unlabeled pair of nodes $(v_1,v_3)\notin\mathcal{E}_\mathrm{P}$, which we cannot obtain through IMF.

While the IMF approach is promising, it is not trivial to predict a new link using a factorized incidence matrix (Fig. 1 [BOTTOM]). The main purpose of this paper is (i) to develop a new link prediction method based on IMF and (ii) to confirm the hypothesis by thorough experiments with synthetic and real-world datasets.

2 IMF-based Link Prediction

In this section, we newly propose an optimization-based efficient link prediction method adopting the IMF approach.

Algorithm. Figure 2 illustrates the overview of our method. Given an incidence matrix $B\in\mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}_\mathrm{P}|}$ of the graph $G=(\mathcal{V},\mathcal{E}_\mathrm{P})$, IMF first factorizes B into two matrices X and Y using truncated SVD:

$$B\approx U_k\Sigma_k{V_k}^\top=XY^\top, \tag{1}$$

where $X:=U_k\Sigma_k$ and $Y:=V_k$. It provides us latent vectors of nodes X and those of positive links Y. Here, for any positive link (v_i,v_j), $\boldsymbol{b}_{(i,j)}\approx X\boldsymbol{y}_{(i,j)}$ holds, where $\boldsymbol{b}_{(i,j)}:=(0,\dots,0,\underset{i}{1},0,\dots,0,\underset{j}{1},0,\dots,0)^\top$ is a column vector of B. Our idea is to predict a link on an unlabeled node pair (v_i',v_j') by how well we can recover its latent vector $\boldsymbol{y}_{(i',j')}$ that is consistent with the factorization, i.e., $\boldsymbol{b}_{(i',j')}\approx X\boldsymbol{y}_{(i',j')}$. This

[1] Tohoku University, Japan, email: yokoi@ecei.tohoku.ac.jp
[2] IBM Research - Tokyo, Japan, email: KAJINO@jp.ibm.com
[3] Kyoto University, Japan, email: kashima@i.kyoto-u.ac.jp

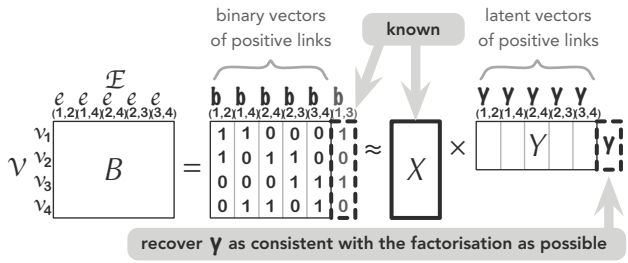


Figure 2. Overview of our link prediction method based on IMF.

idea boils down to the following scoring function, and this optimization problem can be solved in a closed form:

$$s_{\mathrm{IMF}}(v_i,v_j) := -\min_{\boldsymbol{y}\in\mathbb{R}^k} \left\|\boldsymbol{b}_{(i,j)} - X\boldsymbol{y}\right\|_2^2 = -\|\boldsymbol{w}_i+\boldsymbol{w}_j\|_2^2, \quad (2)$$

$$(\boldsymbol{w}_1,\ldots,\boldsymbol{w}_{|\mathcal{V}|}) := X(X^\top X)^{-1}X^\top - I_{|\mathcal{V}|} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}. \quad (3)$$

Computational Efficiency. At first sight, the computational cost of the IMF-based method seems more expensive than the AMF-based method because the size of an incidence matrix is larger than that of an adjacency matrix generally. However, with a simple contrivance, the cost of the matrix factorization of our method can be as small as that of the AMF-based method. Observing that we only need the matrices U_k and Σ_k in the matrix factorization (Eq. (1)), it is sufficient to factorize the positive semi-definite symmetric matrix $BB^\top$ into $Q_k\Lambda_k{Q_k}^\top$ by truncated SVD to obtain $U_k = Q_k$ and $\Sigma_k = \Lambda_k^{1/2}$. Since the size of $BB^\top$ is the same as the adjacency matrix A, the computation time of matrix factorization of our method is the same as that of AMF.

Moreover, the construction of the matrix $BB^\top$ requires almost the same computation time as that of the adjacency matrix A because $BB^\top = A + D$ holds, where D denotes $\mathrm{diag}(d_1,\ldots,d_{|\mathcal{V}|})$, and each d_i corresponds to the degree of a node v_i.

3 Experiments

To demonstrate that IMF actually counters the sparsity problem better than AMF, we conducted comparative experiments with synthetic and real-world datasets.

Datasets. In the first experiment, we generated 10 synthetic graphs ($|\mathcal{V}| = 10^4, |\mathcal{E}_\mathrm{P}| \simeq 10^4,\ldots,10^6$) by the Barabási–Albert model [3], which possess scale-free and small-world properties. In the second experiment, we extracted all the unweighted and undirected real-world graphs from KONECT[4] and chose the 24 smallest graphs in terms of the size $|\mathcal{V}|$ ($|\mathcal{V}| \simeq 10^1,\ldots,10^4, |\mathcal{E}_\mathrm{P}| \simeq 10^1,\ldots,10^6$).

Performance Measure. We used ROC–AUC to evaluate the performance of the scoring function, which is known to be a proper performance measure in link prediction [9].

Experimental Procedure. We conducted five-fold cross validation to measure the performance of IMF and AMF by repeating the following process, and then reported the mean of AUC. First, given $G = (\mathcal{V},\mathcal{E}_\mathrm{P})$, we randomly divide $\mathcal{E}_\mathrm{P}$ into $\mathcal{E}_\mathrm{P}^{(\mathrm{train})}$, $\mathcal{E}_\mathrm{P}^{(\mathrm{dev})}$, and $\mathcal{E}_\mathrm{P}^{(\mathrm{test})}$ by a ratio of $3:1:1$. Second, with $\mathcal{E}_\mathrm{P}^{(\mathrm{train})}$, we learn a scoring function s_k for each $k \in \{2^0, 2^1, \ldots, \min\{2^{14}, 2^{\lfloor \log_2(\mathrm{rank}\, M)\rfloor}\}\}$, where k is the rank of truncated SVD, and M is the incidence or adjacency matrix. Then we select the best hyperparameter k in terms of AUC of s_k. Third, with $\mathcal{E}_\mathrm{P}^{(\mathrm{test})}$, we calculate AUC of $s_{\mathrm{best}k}$ as results.

[4] http://konect.uni-koblenz.de/

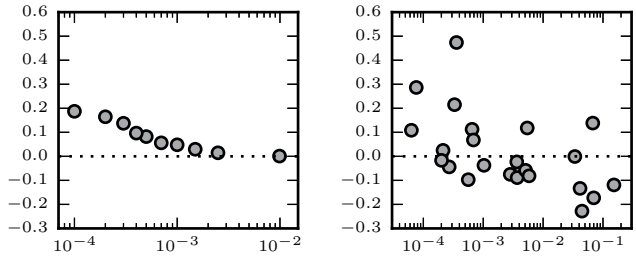


Figure 3. Scatter plot illustrating the relation between the sparsity (x-axis, $|\mathcal{E}_\mathrm{P}|/|\mathcal{V}|^2$) and the performance improvement of IMF over AMF (y-axis, $\mathrm{AUC}_\mathrm{IMF} - \mathrm{AUC}_\mathrm{AMF}$). Each point corresponds to each graph. [LEFT] Synthetic datasets. [RIGHT] Real-world datasets.

Experimental Results. Figure 3 [LEFT] shows the experimental results on the synthetic datasets. The Spearman's ρ between the sparsity measure and the AUC improvement of IMF over AMF is $-1.0 < 0$ ($p = 0.0 < 0.01$); i.e., the performance gain of IMF over AMF increases as the original graph becomes sparser, and the hypothesis is strongly supported. Furthermore, the AUCs of IMF on all the synthetic graphs are nearly constant (0.70), while that of AMF becomes worse as the graph becomes sparser ($0.72,\ldots,0.50$). It implies that the IMF approach is potentially capable of capturing scale-free or small-world properties of networks.

Figure 3 [RIGHT] shows the experimental results on the real-world datasets. Similar to the former experiments, Spearman's $\rho = -0.55 < 0$ ($p = 0.0054 < 0.01$), which also supports our hypothesis.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Number 15H01704.

REFERENCES

[1] E. Acar, D. M. Dunlavy, and T. G. Kolda, 'Link Prediction on Evolving Data Using Matrix and Tensor Factorizations', in *LDMTA at ICDM*, pp. 262–269, (2009).
[2] M. Al Hasan and M. J. Zaki, 'A Survey of Link Prediction in Social Networks', in *Social Network Data Analytics*, 243–275, (2011).
[3] A. L. Barabási and R. Albert, 'Emergence of Scaling in Random Networks', *Science*, **286**(5439), 509–512, (1999).
[4] E. Dong, J. Li, and Z. Xie, 'Link Prediction via Convex Nonnegative Matrix Factorization on Multiscale Blocks', *J. Appl. Math.*, **2014**, 786156:1–786156:9, (2014).
[5] L. Getoor, 'Link Mining: A New Data Mining Challenge', *SIGKDD Explor.*, **5**(1), 84–89, (2003).
[6] L. Getoor and C. P. Diehl, 'Link Mining: A Survey', *SIGKDD Explor.*, **7**(2), 3–12, (2005).
[7] J. Kunegis and A. Lommatzsch, 'Learning Spectral Graph Transformations for Link Prediction', in *ICML*, pp. 561–568, (2009).
[8] D. Liben-Nowell and J. Kleinberg, 'The Link-Prediction Problem for Social Networks', *J. Am. Soc. Inf. Sci. Technol.*, **58**(7), 1019–1031, (2007).
[9] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, 'New Perspectives and Methods in Link Prediction', in *KDD*, pp. 243–252, (2010).
[10] L. Lü and T. Zhou, 'Link prediction in complex networks: A survey', *Physica A*, **390**(6), 1150–1170, (2011).
[11] A. K. Menon and C. Elkan, 'Link Prediction via Matrix Factorization', in *ECML PKDD*, pp. 437–452, (2011).
[12] M. J. Rattigan and D. Jensen, 'The Case For Anomalous Link Discovery', *SIGKDD Explor.*, **7**(2), 41–47, (2005).
[13] P. Wang, B. Xu, Y. Wu, and X. Zhou, 'Link prediction in social networks: the state-of-the-art', *Sci. China Inform. Sci.*, **58**(1), 1–38, (2014).

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

doi:10.3233/978-1-61499-672-9-1732

Dialogues as Social Practices for Serious Games

Agnese Augello[1] and **Manuel Gentile**[2] and **Lucas Weideveld** and **Frank Dignum**[3]

Abstract. The paper describes an architecture for a social conversational agent. The aim is to use the agent in a serious game to improve the social and communicative skills of the players, showing the social effects of conversational choices on the emotions and behavioural changes of the interlocutors.

1 Introduction

A growing body of research considers the possession of adequate interpersonal, social and communicative competences as necessary for ensuring social, psychological and occupational well-being [1]. Through "role playing" it is possible to practice the desired behaviours in a controlled setting [1]. However, this approach can be difficult and expensive; often actors are used to train students, who can only practice a limited number of times.

Serious games can be exploited as an innovative and valid approach by means of simulation of interactions with virtual characters. Virtual agents can be used to bring social elements of interactions into simulations [2] [3][4]. The players can interact with the agents to experience the social effects of a conversation [5]. In order to bound the amount of social and dialogue information that has to be encoded and to bundle social interactions into standard packages we propose the use of *social practices* [7]. A social practice refers to a routinized type of behaviour typically and habitually performed in a society. Social practices are used by people as well to direct and limit the interactions and set expectations. In [8], the theory is analyzed considering an individual perspective in order to formalize it into an agent architecture. In this paper we will show how social practices can be used to implement a serious game to support the learning of social and communicative skills by medical students.

2 A social practice model

The social practice model proposed in [8] allows for the implementation of cognitive agents able to use the social practice as a first-class construct in their deliberation processes. According to this model a social practice is characterized by a *Physical Context* that describes physical objects and individuals with a meaningful role in the practice, a *Social Context* that describes the social interpretation of the environment, the *Activities* that an agent could perform, the *Plan Patterns* that the agent can use to construct a plan to reach a specific goal, a *Meaning* of the agent's activities and plans within the social practice, and finally the *Competences* that an agent should have to perform the activities of the social practice. Table 1 summarizes the components of a specific social practice.

Table 1. An example of social practice formalization: Consultation with a doctor

Abstract Social Practice	**Doctor Patient Dialogue**
Physical Context	
Resources	current time,medical instruments
Places	hospital,office
Actors	user, agent
Social Context	
Social interpretation	consulting room, consulting time
Roles	doctor, patient
Norms	patient is cooperative (gives truthful and complete answers), doctor is polite
Activities	welcome, presentation, data gathering, symptom description, speech acts
Plan patterns	Welcome, Presentation, Data Gathering, Symptom Description, Therapy
Meaning	support the patient, create trust, eliciting patient's problems and concerns,empathic response
Competences	listening effectively, being empathic, use effective explanatory skills, adapt conversation

3 A Social Agent Architecture for serious games

The proposed architecture (fig. 1), is composed of three main modules; a complete description of the *Social Practice Selection* module module can be found in [9], while in this work we focus on the formalization of the agent's *Identity* and its *Deliberation Engine*.

3.1 Identity

The identity of the agent formalizes his beliefs, the information related to the possible social practices, the state of the dialogue the rules for the generation of plans, the analysis of norm violations and state variables updating. A formalization based on the concept of social practice allows the agent to interpret the context from a social point of view and to perform the more suitable plan pattern contextualizing the dialogue. The identity of the agents includes also the linguistic knowledge required to manage the conversation: a set of question-answers modules (called categories) described according to S-AIML. It is an extension of the AIML language, that allows to bind the categories to specific practices and their activities. Specific tags have been introduced to contextualize the dialogue inside a social practice (*social practice* tag), according to a specific activity (*activity* tag), when specific preconditions are satisfied (*precondition* tag) and to have more freedom in the effect management. The enhancement of the AIML language was required because the AIML dia-

[1] ICAR-CNR, Italy, email: augello@pa.icar.cnr.it
[2] ITD-CNR, Italy, email : manuel.gentile@itd.cnr.it
[3] Utrecht University, The Netherlands, email: F.P.M.Dignum@uu.nl

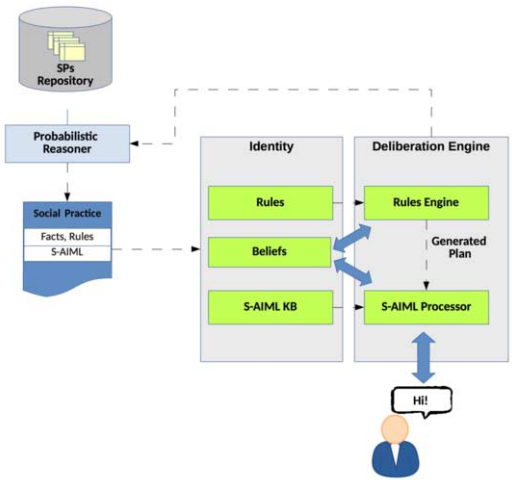


Figure 1. Social Chatbot Architecture

logue designer cannot use only the variables/parameters to control the evolution of the dialogue.

3.2 Deliberation Engine

This module allows the agent to deliberate according to the recognized social practice. It is composed of two main components. The first is a reasoner that exploits facts and rules to understand the proper action to execute(i.e. the updating of a fact, the execution of a plan or a specific activity). The other is the S-AIML processor that processes the S-AIML KB. These modules allow for a dynamic activation of a set of S-AIML categories related to the current social context with a consequent reduction of the number of categories that can match the user sentences and a simplification of the agent deliberation. Without a social practice oriented approach at every step of the dialogue all dialogue variables values and all the rules should be checked. Let us suppose that the player tells to the agent "You should make a computerized axial tomography". Several conditions must be considered, as highlighted in the following category (for shortness only few conditions have been considered).

```
<category>
  <pattern>You should make a computerized axial tomography
  </pattern>
  <template>
    <condition name="interlocutor" >
     <li value="familiar">Who gave you the medical degree?
       <think><set name="emotion">annoyed</set></think>
     </li>
     <li value="doctor">
       <condition name="doctor_type" >
        <li value="faimily_doctor">Tel me doctor, could i have
              something of serious?
        <think> <set name="emotion">fear </set> </think>
          </li>
        </condition>
       </condition>
    </li>
  </template>
</category>
```

Is the other speaker a doctor or is he a family member or another patient? And if he is a doctor and the agent does not know him, was the appointment scheduled or is it unexpected? The same recommendation to make a CAT examination, triggers different behaviours. Instead, using a social practice approach, when the social practice is correctly identified the dialogue is managed according to the rules bound to the practice that are satisfied, as highlighted in the following category.

```
<social practice name="unknown_doctor_consultation">
  <category>
    <precondition> <el>trust>low</el> </precondition>
    <pattern>You must make a computerized axial tomography</pattern
        >
    <template>
      Why should i make this examination?
      <think>
        <el>emotion=fear</el>
        <el>trusting=trusting−3</el>
      </think>
    </template>
  </category>
</social practice>
```

The activation of a social practice determines meaningful changes also on the entire dialogue path. From time to time, new information is acquired and the agent can re-plan according to the new situation. Moreover, it continuously monitors possible violations of social practice norms. In case of a violation the agent will act in a proper manner, stopping the execution of that practice if necessary.

4 Conclusion

The proposed architecture puts social practice at the heart of the deliberative process of a conversational agent. We presented a few examples based on the case study of medical consultations. We discussed some preliminary steps in the formalization of the agent's knowledge and the introduction of the new S-AIML language. Future work will regard a more developed implementation and the validation of the assumption by means of an experimental evaluation following a proper learning design approach.

REFERENCES

[1] Segrin, C., & Givertz, M. (2003). Methods of Social Skills Training and Development. In J. O. Greene & B. R. Burlesono (Eds.), Handbook of Communication and Social Interaction Skills. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

[2] Swartout, W. , Artstein, R., Forbell, E., Foutz, S., Lane, H. C., Lange, B., Morie, J., Noren, D, Rizzo, S., and Traum, D.. Virtual humans for learning. *AI Magazine 34(4): 13-30, 2013.*

[3] Babu, S. V., Suma, E. , Hodges , L. F., and Barnes, T. (2011). Learning cultural conversational protocols with immersive interactive virtual humans. *The International Journal of Virtual Reality, 2011, 10(4): 25-35.*

[4] Dargis, M., Koncevics, R., Silamikelis, A., Kirikova, M. *Game-based Training of Communication Skills in Requirements Engineering*. REFSQ Workshops 2014: 147-148

[5] Fairclough, Norman. *Analysing discourse: Textual analysis for social research*. Psychology Press, 2003.

[6] Jeuring, J., Grosfeld, F., Heeren, B., Hulsbergen, M., IJntema, R., Jonker, V., Mastenbroek, N., Van Der Smagt, M., Wijmans, F. , Wolters, M., Van Zeijts, H. *Demo: Communicate! - a serious game for communication skills*. Proceedings EC-TEL 2015: 10th European Conference on Technology Enhanced Learning, 2015.

[7] Reckwitz, Andreas. Toward a theory of social practices a development in culturalist theorizing. *European journal of social theory 5.2 (2002): 243-263.*

[8] Dignum, Virginia, and Frank Dignum. *Contextualized Planning Using Social Practices*. Coordination, Organizations, Institutions, and Norms in Agent Systems X. Springer International Publishing, 2014. 36-52.

[9] Augello, A. , Gentile, M., Dignum, F. *Social Practices for Social Driven Conversations in Serious Games*. Proceedings of the Games and Learning Alliance conference, GaLA 2015, Roma, December 9-11, 2015, in press.

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

doi:10.3233/978-1-61499-672-9-1734

Abducing Workflow Traces: A General Framework to Manage Incompleteness in Business Processes

Federico Chesani[1] and **Riccardo De Masellis**[2] and **Chiara Di Francescomarino**[2] and **Chiara Ghidini**[2] and **Paola Mello**[1] and **Marco Montali**[3] and **Sergio Tessaris**[3]
[1] **University of Bologna, Viale Risorgimento 2, 40136 – Bologna, Italy**
[2]**FBK-IRST, Via Sommarive 18, 38050 Trento, Italy**
[3]**Free University of Bozen–Bolzano, piazza Università, 1, 39100 Bozen-Bolzano, Italy**

Abstract. The capability to store data about Business Process executions in so-called Event Logs has brought to the identification of a range of key reasoning services (consistency, compliance, runtime monitoring, prediction) for the analysis of process executions and process models. Tools for the provision of these services typically focus on one form of reasoning alone. Moreover, they are often very rigid in dealing with forms of incomplete information about the process execution. While this enables the development of ad hoc solutions, it also poses an obstacle for the adoption of reasoning-based solutions. In this paper we exploit the power of abduction to provide a flexible, and yet computationally effective framework able to reinterpret key reasoning services in terms of incompleteness and observability in a uniform and effective way.

1 Introduction

The proliferation of IT systems able to store process executions in so-called event logs has originated, in the Business Process (BP) community, a quest towards tools for discovering, checking the conformance and enhancing process models based on actual behaviours. Focusing on conformance, that is, to assess how a *prescriptive* (or "de jure") process model relates to the execution traces, this general notion can be declined in specific "use cases", such as *model consistency*, *trace compliance*, *runtime monitoring* and *prediction/recommendation*.

A number of different tools offer ad hoc solutions that cover only few of the "use cases" and do not easily adapt to different workflow languages. This poses a problem, given the current trend of enriching BP languages with new constructs complementing the control flow knowledge. A second rigidity is in dealing with *incomplete information* about the process execution: the presence of not monitorable activities or errors in the logging procedure makes handling *incomplete event data* one of the main challenges of the BP community

In this work we exploit the paradigm of Abductive Constrained Logic Programming (ACLP, [4]), and the SCIFF abductive framework [1] to provide a general purpose environment able to support *conformance in its different "use cases"* in the presence of *incomplete event data*. Indeed, abduction fits in a natural manner: facts are observed in the execution traces, and need to be explained/diagnosed with respect to what is envisaged by the process model.

2 Process Models, Reasoning, and Incompleteness

We focus on structured process models in the spirit of [5], enriched with temporal constraints. We illustrate our investigation with an example of temporal workflow taken from [6], and reproduced in the left hand side of Figure 1. This workflow contains: 14 activities ($\mathsf{A1}, \ldots \mathsf{A14}$), 2 pairs of exclusive gateways ($\langle \mathsf{X1}, \mathsf{X4} \rangle, \langle \mathsf{X2}, \mathsf{X3} \rangle$), and 1 pair of parallel gateways ($\langle \mathsf{P1}, \mathsf{P2} \rangle$). In addition, activities are labelled with expressions of the form $[d_{min}, d_{max}]$ (*duration range* of the activity), while dashed arrows between activities expresses *inter-task constraints* involving the start or end of the activities. A sample trace that logs the execution of the aforementioned process is:

$$\begin{gathered}\{(\mathsf{A1}, [2, 7]), (\mathsf{A2}, [10, 15]), (\mathsf{A3}, [16, 46]), (\mathsf{A4}, [50, 200]), \\ (\mathsf{A13}, [300, 317]), (\mathsf{A14}, [320, 330])\}\end{gathered} \tag{1}$$

To incorporate incompleteness into the picture, the process models we consider are also equipped with *observability information*. Activities can be: (i) *observable*, if they are always explicitly logged, even if some information (e.g., the activity name or the starting time) may be missing; (ii) *non-observable*, if they are never logged, even if executed; (iii) *partially observable*, if they may or may not have been logged. We introduce the notion of *structured process with observability and time* (SPOT) as the tuple $\langle \mathcal{A}, obs, P, dur, tcon \rangle$, where $\mathcal{A}$ is a finite set of *activity (names)*; $obs : \mathcal{A} \to \{\mathtt{o}, \mathtt{n}, \mathtt{p}\}$ provides observability about each activity; P is the *top process block*, inductively defined as either a $\mathtt{a} \in \mathcal{A}$, or a combination of $\{\mathtt{seq}, \mathtt{xor}, \mathtt{and}, \mathtt{or}\}$ blocks; dur and $tcon$ captures the activity duration and the inter-task constraints.

A trace is a set of triples $\langle \mathtt{a}, s, e \rangle$ denoting the happened event $\mathtt{a}$, and the start/end timestamps s, e. Missing information units will be denoted with the special symbol "_". A trace is *partially specified* if some of its event triples contain the symbol _, *fully specified* otherwise. Moreover, we distinguish between *total* and *partial traces*, where in the latter additional event triples may be implicitly present even if not explicitly listed in the trace.

The notion of **Strong Compliance** applies to total traces, and requires a trace to be compliant to the blocks structure, and also the duration and inter-task temporal constraints. It reflects the usual notion of compliance for business processes, where it is assumed that the trace represents a complete end-to-end execution, such as in (1).

Other resoning services supported in our framework are the **Model Consistency**, that checks if a SPOT enables acceptable executions from start to end.; and, more related to incompleteness, the **Conditional Compliance**, that handles the case where the trace under analysis is indeed partial and/or partially specified: if such incompleteness hinders the possibility of replaying it on the process model, strong compliance might be regained by assuming that the trace included additional information on the missing or partially specified

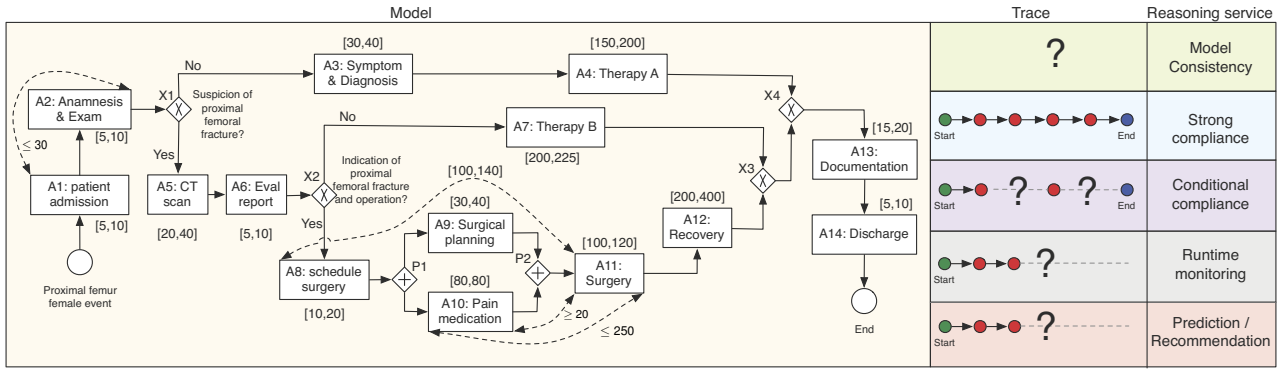


Figure 1. A process for femoral fracture treatment, reasoning services and incomplete execution traces.

events. The trace:

$$\{(\mathsf{A1},[2,7]),(\mathsf{A2},\_)(\mathsf{A4},[50,200]),(\mathsf{A13},[300,317]),(\mathsf{A14},[320,330])\} \quad (2)$$

is conditionally compliant, i.e. it might be considered strong compliant under the hypotheses that A2 was executed (satisfying the temporal constraints) and A3 (missing in the trace) was executed as well. Note that the set of assumptions needed to reconstruct strong compliance is not necessarily unique. This because alternative strongly compliant real process executions might have led to the recorded partial trace.

Finally, when dealing with ongoing executions, **Runtime Monitoring** aims to detect early violations of compliance / ensure the existence of a positive outcome, while **Prediction/Recommendation** provides (if possible) a completion that satisfies the model.

3 Abduction and Incomplete Processes

Abductive Logic Program (ALP) [4] is a non-monotonic reasoning process where hypotheses are made to explain observed facts. Given a set Γ of logical assertions known to hold, and a formula ϕ (corresponding to observed facts), abduction looks for a further set Δ of hypothesis, taken from a given set of abducible $\mathcal{A}$, which complements Γ in such a way that ϕ can be inferred (in symbols $\Gamma \cup \Delta \models \phi$). The set Δ is called *abductive explanation* (of ϕ).

In this paper we leverage on ACLP and on the SCIFF abductive logic programming framework [1], efficiently implemented using the CHR framework [2]. Beside the general notion of abducible, the SCIFF framework has been enriched with the notions of *happened event*, *expectation*, and *compliance* of an observed execution with a set of expectations. In our context, happened events account for events that have been logged in the trace, while abducibles are used to make hypothesis on events that are not recorded in the examined trace.

Our solution consists on translating a SPOT model into an ACLP program encoded using the SCIFF notation: observable activities are always expected to be observed, while partially obsefvable activities can be abduced (hypothesized) if not presente in the trace. Then, the SCIFF proof procedure is queried for possible abductive answers Δ_i that explain the observations (the given trace), under the conditions imposed by the process model. Temporal constraints are mapped by means of CLP predicates, that are fully supported within the SCIFF framework, thus allowing also the required temporal reasoning.

The different reasoning services are all supported within the SCIFF framework: *Model consistency* is directly supported by assuming that all the activities are partially observable, and the trace to be verified is empty. *Runtime monitoring* is alike, with the difference that the trace captures the events already happened. *Strong compliance* corresponds to the original reasoning task of the SCIFF, and it is still supported in our context: indeed, it corresponds to looking for at least an empty abductive answer Δ_i. *Conditional compliance* is supported as well, by looking at non-empty abductive answers. Similarly, *prediction/recommendation* is supported by simply applying out approach at run-time, and by interpreting the Δ_i as suggestions. A prototype implementation of the framework is currently available for download at http://ai.unibo.it/AlpBPM.

4 Conclusions

We have presented an abductive framework to support business process compliance, in its different forms, by attacking the different forms of incompleteness that may be present in an execution trace. Empirical evaluation using the model in Figure 1 shows that the different reasoning services can be computated with times spanning from few millisec. (when looking for a solution) up to several minutes (when looking for all the solutions). Such difference is directly related to the degree of incompleteness in the model and in the trace.

Concerning future development, the SCIFF framework is based on first-order logic, thus paving the way towards the incorporation of data [3] and the management of more sophisticated forms of incompleteness. A further reasoning service on temporal workflows is the one of controllability. Moreover, an extension of our work to deal with dynamic controllability, by integrating constraint propagation and filtering, would be an interesting and feasible future work.

Acknowledgements. This research has been partially carried out within the Euregio IPN12 KAOS, funded by the "European Region Tyrol-South Tyrol-Trentino" (EGTC) under the first call for basic research projects.

REFERENCES

[1] M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni, 'Verifiable agent interaction in abductive logic programming: The SCIFF framework', *ACM Trans. Comput. Log.*, **9**(4), (2008).
[2] M. Alberti, M. Gavanelli, and E. Lamma, 'The chr-based implementation of the SCIFF abductive system', *Fundamenta Informaticae*, **124**(4), 365–381, (2013).
[3] R. De Masellis, F. M. Maggi, and M. Montali, 'Monitoring data-aware business constraints with finite state automata', in *Proc. of ICSSP*. ACM Press, (2014).
[4] A. C. Kakas, R. A. Kowalski, and F. Toni, 'Abductive logic programming', *J. Log. Comput.*, **2**(6), (1992).
[5] Bartek Kiepuszewski, Arthur Harry Maria ter Hofstede, and Christoph J. Bussler, 'On structured workflow modelling', in *Seminal Contributions to Information Systems Engineering*, Springer, (2013).
[6] A. Kumar, S. R. Sabbella, and R. R. Barton, *Business Process Management: 13th Intl. Conf., BPM 2015, Innsbruck, Austria, August 31 – September 3, 2015, Procs.*, chapter Managing Controlled Violation of Temporal Process Constraints, 280–296, Springer International Publishing, Cham, 2015.

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

doi:10.3233/978-1-61499-672-9-1736

Optimal Simple Strategies for Persuasion

Elizabeth Black and **Amanda Coles** and **Christopher Hampson**[1]

1 Introduction

Argument dialogues provide a principled way of structuring rational interactions between participants (be they human or machine), each arguing over the validity of certain claims, with each agent aiming for an outcome that achieves their dialogue goal (*e.g.*, to persuade the other participant to accept their point of view [8], or to reach agreement on an action to perform [2]). Achievement of an agent's dialogue goal typically depends on both the arguments that the agent chooses to make during the dialogue, determined by its *strategy*, and the arguments asserted by its interlocutor. The strategising agent—the *proponent*—thus has the difficult problem of having to consider not only which arguments to assert but also the possible responses of its *opponent*. This problem is compounded since the opponent may exploit knowledge inferred from those arguments asserted by the proponent to construct new arguments. Hence, the proponent must take care not to divulge information that is advantageous to its opponent.

The important challenge of how to generate strategies for such a proponent has not been widely explored [10]. Notable exceptions are the work of Hadoux *et al.* [7], which employs mixed observability Markov decision processes to generate optimal policies for the proponent to follow; the work of Rienstra *et al.* [9], which applies a variant of the minimax algorithm to determine an effective proponent strategy; and the work of Black *et al.* [3], which employs heuristic planning techniques to determine an optimal proponent strategy for a simple asymmetric persuasion setting.

We highlight two types of uncertainty in the strategic argumentation problem: uncertainty over the arguments initially know to the opponent, captured by the *opponent model*, and uncertainty over how the opponent chooses to argue, given their initial knowledge base. Both [9] and [3] deal with uncertain models of the opponents initial knowledge base, where the opponent's strategy is known (*i.e.*, optimal [9] or deterministic [3]); while [7] considers the case in which we are certain of the arguments known to the opponent but have only a stochastic model of how they will behave.

The key novelties in our approach are that we deal with both of these types of uncertainty simultaneously the former through use of an *uncertain opponent model* and the latter by generating *conformant* strategies, that is strategies that are effective regardless of the opponent strategy. Further, our work is the first to generate strategies in a setting where the opponent may exploit information obtained during the dialogue to construct arguments unknown to it at the start of the dialogue, necessitating more cautious strategies.

2 Strategic Argumentation Problem

We consider a strategic argumentation setting in which both agents exchange arguments, with the proponent aiming to convince its opponent of some topic argument. Our problem comprises the following:

- An argumentation framework $(\mathcal{A}, \rightarrow)$, comprising a set of arguments $\mathcal{A}$ and a binary *attack relation* $\rightarrow \subseteq (\mathcal{A} \times \mathcal{A})$,
- The function $Acc : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ describes the set of *acceptable* arguments $Acc(\mathcal{B})$, for each subset $\mathcal{B} \subseteq \mathcal{A}$, subject to some agreed semantics (see [5], for details).
- A proponent model $\mathcal{K}_{\mathcal{P}} \subseteq \mathcal{A}$, comprising a set of arguments available to the proponent.
- An *uncertain* opponent model $(\mathcal{E}, \mathcal{K}_{\mathcal{O}}, p)$, comprising a finite set of labels $\mathcal{E} = \{\mathcal{O}_1, \ldots, \mathcal{O}_m\}$, a function $\mathcal{K}_{\mathcal{O}} : \mathcal{E} \rightarrow 2^{\mathcal{A}}$ associating each label $\mathcal{O}_i \in \mathcal{E}$ with a possible opponent model $\mathcal{K}^i_{\mathcal{O}} \subseteq \mathcal{A}$, and a probability distribution $p : \mathcal{E} \rightarrow \mathbb{Q}$, assigning to each $\mathcal{O}_i \in \mathcal{E}$ the likelihood that the opponent initially knows $\mathcal{K}^i_{\mathcal{O}}$.
- A closure operator $\mu : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ provides a description of how each agent may derive new arguments from knowledge acquired during the dialogue. *i.e.*. $\mu(\mathcal{B}) \subseteq \mathcal{A}$ is the set of all arguments that can be derived by $\mathcal{B} \subseteq \mathcal{A}$.

A *dialogue* is a sequence of *moves* $D = [M_0, M_1 \ldots, M_n]$ in which each move $M_k \subseteq \mathcal{A}$ is a finite set of arguments. A dialogue terminates when $M_{k-1} \cup M_k = \emptyset$, and is *successful* for the proponent w.r.t. a given topic $t \in \mathcal{A}$, if $t \in Acc(M_0 \cup \cdots \cup M_n)$.

A (general) strategy for $\mathsf{Ag} \in \{\mathcal{P}, \mathcal{O}\}$ is a function $\sigma_{\mathsf{Ag}} : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ such that $\sigma(\mathcal{B}) \subseteq \mu(\mathcal{K}_{\mathsf{Ag}} \cup \mathcal{B})$, for all $\mathcal{B} \subseteq \mathcal{A}$, that determines which move Ag should make, given the arguments asserted thus far. A *simple* strategy is a special case of strategy specified by a sequence of moves $S = [S_0, \ldots, S_n]$, with $S_k \subseteq \mathcal{K}_{\mathcal{P}}$, representing sets of arguments to be asserted in turn by the proponent, unless the opponent accepts the topic, or the sequence is exhausted, in which case we assert $\emptyset$. A pair of strategies $(\sigma_{\mathcal{P}}, \sigma_{\mathcal{O}})$ generates the dialogue $D_{(\sigma_{\mathcal{P}}, \sigma_{\mathcal{O}})} = [M_0, \ldots, M_n]$ given by $M_k = \sigma_{\mathsf{Ag}}(M_0 \cup \cdots \cup M_{k-1})$, where $\mathsf{Ag} = \mathcal{P}$ whenever k is even and $\mathsf{Ag} = \mathcal{O}$ whenever k is odd.

A strategy $\sigma_{\mathcal{P}}$ is effective against $\mathcal{O}_i \in \mathcal{E}$ if $D_{(\sigma_{\mathcal{P}}, \sigma_{\mathcal{O}})}$ is successful, for every possible opponent strategy $\sigma_{\mathcal{O}}$ for $\mathcal{O}_i$.

3 Persuasion Dialogues as Planning Problems

We provide a translation of a strategic argumentation problem into a propositional planning problem with (bounded) numerical variables, such that a solution to the planning problem yields an effective strategy for our proponent. Encoding the problem in the standard planning domain language PDDL2.1 [6] allows us to use an implementation of the planner POPF [4] to generate appropriate simple strategies. A key challenge in using a propositional planner to solve these problems is capturing the uncertainty about the opponent's initial beliefs. We use techniques inspired by the current state-of-the-art approach for solving conformant planning problems by compilation to classical planning [1]. The planning problem $\mathcal{P}$ is described as follows:

Variables We require the following numerical variables: (stage), $(\mathsf{probSuccess})$ and $(\mathsf{prob}(i))$, for each $\mathcal{O}_i \in \mathcal{E}$. The variable (stage)

[1] King's College London, Strand, London, UK, WC2R 2LS,
email: firstname.lastname@kcl.ac.uk
This work was partially supported by the EPSRC, grant ref. EP/M01892X/1

Action: proponent(a)	for all $a \in \mathcal{K}_\mathcal{P}$
pre	(i) canAssertP(a) (ii) (stage $= 0$) $\vee$ (stage $= 1$)
eff	(a) $\neg$canAssertP(a) (b) (stage $\leftarrow 2$) (c) **if** dialogueP($D_\mathcal{P}$), addP($a, D_\mathcal{P}, D'_\mathcal{P}$) **then** dialogueP($D'_\mathcal{P}$), $\neg$dialogueP($D_\mathcal{P}$) (d) **if** dialogueO($i, D_\mathcal{O}$) **then** temp($i, D_\mathcal{O}$), $\neg$dialogueO($i, D_\mathcal{O}$) (e) effective(i)

Figure 1: Actions in our Planning Model

regulates the order in which actions may be added to the plan, while the variables (probSuccess) and (prob(i)) are responsible for calculating the probability that a given strategy will be successful. In addition to these, we have the following propositional variables: canAssertP(a) and canAssertO(M, i, D) govern the arguments that $\mathcal{P}$ and $\mathcal{O}$ can assert, respectively, for $a \in \mathcal{K}_\mathcal{P}$, $\mathcal{O}_i \in \mathcal{E}$ and $D \subseteq \mathcal{K}_\mathcal{P}$; dialogueP($D$) and dialogueO($i, D$) describe the set of arguments that have been asserted by $\mathcal{P}$ and $\mathcal{O}_i \in \mathcal{E}$, respectively; temp($i, D$) a temporary 'storage' variable for dialogueO(i, D); successful($D_\mathcal{P}, D_\mathcal{O}$) if ($D_\mathcal{P} \cup D_\mathcal{O}$) is an acceptable set of arguments, where D_{Ag} are the arguments asserted by $\mathsf{Ag} \in \{\mathcal{P}, \mathcal{O}\}$; effective($i$) says that the strategy is effective against $\mathcal{O}_i \in \mathcal{E}$; addP($a, D, D'$) says that $D' = D \cup \{a\}$, and addO(M, D, D') says that $D' = D \cup M$.

Initial Conditions Our set of initial conditions comprises the following numerical conditions: (stage $= 0$), (probSuccess $= 0$), and (prob(i) $= p(\mathcal{O}_i)$), for all $\mathcal{O}_i \in \mathcal{E}$. We also require the following propositional initial conditions: canAssertP(a), for all $a \in \mathcal{K}_\mathcal{P}$, dialogueP($\emptyset$), and dialogueO($i, \emptyset$), for all $\mathcal{O}_i \in \mathcal{E}$, together with the following conditions that, once set, remain unaltered by the effects of any of the actions: canAssertO($M, i, D_\mathcal{P}$) iff $M \subseteq \mu(\mathcal{K}^i_\mathcal{O} \cup D_\mathcal{P})$; successful($D_\mathcal{P}, D_\mathcal{O}$) iff $t \in Acc(D_\mathcal{P} \cup D_\mathcal{O})$; addP($a, D_\mathcal{P}, D'_\mathcal{P}$) iff $D'_\mathcal{P} = D_\mathcal{P} \cup \{a\}$; addO($M, D_\mathcal{O}, D'_\mathcal{O}$) iff $D'_\mathcal{O} = D_\mathcal{O} \cup M$.

Goal Condition Our goal is a single condition (probSuccess $> \lambda$), where $\lambda \in [0, 1]$ is the required probability of success.

Actions Our set of actions comprises three types of action whose preconditions and effects are described in Figures 1–2, where all free parameters are universally quantified. For each argument $a \in \mathcal{K}_\mathcal{P}$ known to the proponent, there is a (proponent(a)) action, which emulates the act of the proponent asserting a. A single move is built up by iterated application of these actions (simulating the assertion of a set of arguments by the proponent). The opponent's move is captured by a single (opponent) action. This action simulates *all* possible responses for each possible opponent model $\mathcal{O}_i \in \mathcal{E}$, adding them to a 'pool' of possible dialogue states associated with that opponent model. Finally the action (probCount) must be applied after each (opponent) action and sums the total probability of *guaranteed* success, against each of the possible opponent models $\mathcal{O}_i \in \mathcal{E}$.

A solution to the planning problem $\mathcal{P}$, generated by a planner, is a sequence of actions that transforms the initial state into a state that satisfies the goal. Such a solution corresponds to a simple strategy that will be successful with probability $> \lambda \in [0, 1]$. To find an *optimal* simple strategy maximising λ, we first find a solution where $\lambda > 0$ and then iteratively seek solutions with strictly larger λ values.

4 Results

Owing to the lack of existing approaches for proponent strategy generation that do not depend on knowledge of the opponent's strategy, we instead benchmarked our approach against a naïve depth-first search, that exhaustively searches all simple strategies and evaluates their effectiveness with respect to the given opponent model. Our analysis showed that that for smaller examples the performance two approaches in comparable; however, for larger examples, comprising more than nine arguments, the planning approach outperforms the naïve search by several orders of magnitude.

Action: opponent	
pre	(i) (stage $= 1$)
eff	(a) (stage $\leftarrow 2$) (b) **if** dialogueP($D_\mathcal{P}$), temp($i, D_\mathcal{O}$), canAssertO($M, i, D_\mathcal{P}$), addO($M, D_\mathcal{O}, D'_\mathcal{O}$), $\neg$successful($D_\mathcal{P}, D'_\mathcal{O}$) **then** dialogueO($i, D'_\mathcal{O}$), $\neg$effective(i) (c) $\neg$temp($i, D_\mathcal{O}$)

Action: probCount	
pre	(i) (stage $= 2$)
eff	(a) (stage $\leftarrow 0$) (b) **if** effective(i) **then** (probSuccess $\leftarrow$ probSuccess $+$ prob(i)), (prob(i) $\leftarrow 0$)

Figure 2: Actions in our Planning Model

We performed further experiments to compare to the most closely related approach by Hadoux et al. [7] on the problems described in their paper. The two approaches are not directly comparable: we do not rely on knowledge of the opponents strategy, whilst they do; they generate policies, dependent on the opponent's response, whereas we generate simple strategies. We did however make two observations: first our approach finds solutions an order of magnitude faster than theirs; and second in all of the problems they propose a simple strategy is sufficient to guarantee 100% success rate, without the need for a policy, meaning that our approach is equally strong in these particular settings. Our problems, the planner we used, and the implementation of our translation and of our naïve algorithm are all available from: http://tinyurl.com/jfxotsg.

REFERENCES

[1] Alexandre Albore, Hector Palacios, and Hector Geffner, 'Compiling uncertainty away in non-deterministic conformant planning.', in *European Conference on Artificial Intelligence*, pp. 465–470, (2010).

[2] Elizabeth Black and Katie Bentley, 'An empirical study of a deliberation dialogue system', in *Theory and Applications of Formal Argumentation*, 132–146, Springer LNCS 7132, (2012).

[3] Elizabeth Black, Amanda Coles, and Sara Bernardini, 'Automated planning of simple persuasion dialogues', in *Computational Logic in Multi-Agent Systems*, 87–104, Springer LNAI 8624, (2014).

[4] Amanda J. Coles, Andrew I. Coles, Maria Fox, and Derek Long, 'Forward-chaining partial-order planning', in *International Conference on Automated Planning and Scheduling*, (2010).

[5] Phan Minh Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games', *Artificial Intelligence*, **77**(2), 321 – 357, (1995).

[6] Maria Fox and Derek Long, 'PDDL2. 1: An extension to PDDL for expressing temporal planning domains.', *Journal of Artificial Intelligence Research*, **20**, 61–124, (2003).

[7] Emmanuel Hadoux, Aurélie Beynier, Nicolas Maudet, Paul Weng, and Anthony Hunter, 'Optimization of probabilistic argumentation with markov decision models', in *International Joint Conference on Artificial Intelligence*, pp. 2004–2010, (2015).

[8] Henry Prakken, 'Formal systems for persuasion dialogue', *The Knowledge Engineering Review*, **21**(02), 163–188, (2006).

[9] Tjitze Rienstra, Matthias Thimm, and Nir Oren, 'Opponent models with uncertainty for strategic argumentation', in *International Joint Conference on Artificial Intelligence*, pp. 332–338, (2013).

[10] Matthias Thimm, 'Strategic argumentation in multi-agent systems', *Künstliche Intelligenz, Special Issue on Multi-Agent Decision Making*, **28**(3), 159–168, (2014).

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

doi:10.3233/978-1-61499-672-9-1738

Non-Utilitarian Coalition Structure Generation

Oskar Skibski[1] and **Henryk Michalewski**[2] and **Andrzej Nagórko**[3] and **Tomasz P. Michalak**[4,5] and **Andrew Dowell**[6] and **Talal Rahwan**[7] and **Michael Wooldridge**[8]

Abstract. The coalition structure generation problem is one of the key challenges in multi-agent coalition formation. It involves partitioning a set of agents into coalitions so that system performance is optimized. To date, the multi-agent systems literature has focused exclusively on the *utilitarian* version of this problem which seeks to maximize the sum of the values of the coalitions involved. However, there are many examples of situations in which other performance metrics are of interest. In particular, in games with non-transferable utility, we may be more interested in an egalitarian optimal coalition structure, or in minimizing the difference between the utilities of the most affluent and poorest agents. In this paper, we present a number of exact algorithms to solve such non-utilitarian formulations of the coalition structure generation problem.

1 INTRODUCTION

The coalition structure generation problem involves partitioning the set of agents into coalitions so that the performance of the system is optimized. It has been advocated for a variety of potential applications, including improving the surveillance of common areas by autonomous sensors [10]; reducing the uncertainty of the expected energy output of virtual power plants [3]; and increasing the throughput of cognitive radio networks [13].

To date, the multi-agent systems literature has entirely focused on the *utilitarian* version of the coalition structure generation problem in which the objective function to be maximized is the sum of the values of the coalitions involved.[9] Here, the underlying model is typically a game in which every coalition is assigned a single numerical value that represents its utility. Such games are called *transferable utility* (TU) games as an implicit assumption is that a coalition's utility is transferable among agents in the coalition.

While the ability to transfer payoff occurs naturally in many settings, it is not universal. For instance, in goal-oriented systems, agents may derive utility from accomplishing specific goals (e.g. saving lives in disaster response [11]) and one agent may not profit from another agent's goal. Similarly, in resource allocation problems, indivisible and non-transferable resources can be assigned to agents, not to coalitions [7, 6]. Also, non-transferable payoffs occur in various environmental and economic problems [15]. All such domains can be modelled with *non-transferable utility* (NTU) games, in which the utility of a coalition is expressed as a *vector* of real numbers with each entry representing the non-transferable utility of the particular agent within this coalition.

The utilitarian formulation of the coalition structure generation problem can be applied not only to the TU games, but also to NTU ones. In particular, it is enough to sum up the utility vectors for all the coalitions (see the next section for details). However, such an approach is based on an assumption that all agents are completely benevolent and, in the worst case, they agree to sacrifice all their individual utilities. This is because, under utilitarianism, the maximisation of the system welfare is achieved without any regard to the situation of individual agents. For instance, in the NTU with the following non-zero values of coalitions: $V(\{a_1\}) = 9$; and $v(\{a_1, a_2, a_3\}) = [3-\epsilon, 3-\epsilon, 3-\epsilon]$; two coalition structures are considered optimal: $\{\{a_1\}, \{a_2\}, \{a_3\}\}$ and $\{\{a_1\}, \{a_2, a_3\}\}$. In both, the entire value is generated by $\{a_1\}$, while all other coalitions have value zero. On the other hand, if the coalition structure $\{\{a_1, a_2, a_3\}\}$ was chosen then the value of the entire system would be only marginally lower, i.e., $9 - 3\epsilon$, but it would be equally distributed.

Against this background, in our research, we studied two non-utilitarian coalition structure generation problems: *Egalitarian* and *Balanced*. In the former one, the objective function to be *maximized* is the value of the smallest agent utility in the coalition structure (i.e., the poorest agent). In the latter one, the objective function to be *minimized* is the difference between the smallest and the largest agent utilities in the coalition structure (i.e., the difference between the richest and the poorest agents). While the notion of the egalitarian welfare is very well known in social sciences and has been already extensively discussed in the multi-agent systems literature, albeit in different contexts [8, 7, 9, 5], the concept of the balanced welfare is inspired by the economic concept of the Richest/Poorest average income ratio [2], i.e., the relative income difference between the (typically 10%) cohorts of the poorest and the richest members of human societies.[10] We first analysed exact dynamic programming for both problems. Next, we investigated how to solve them under two concise representations of coalitional games, namely MC-nets [12] and Decision Diagrams [17, 1], using linear programming techniques.

[1] University of Warsaw, Poland
[2] University of Warsaw, Poland
[3] University of Warsaw, Poland
[4] University of Warsaw, Poland
[5] University of Oxford, UK
[6] Independent Researcher, UK
[7] Masdar Institute of Science and Technology, UAE
[8] University of Oxford, UK

[9] A detailed overview of the literature on the coalition structure generation problem can be found in the work by [16].

[10] We note that the problem of balanced welfare optimization was also analysed in the OR literature [14] but under the assumption that the input may be *incomplete*, i.e. that not all coalitions are feasible. This is fundamentally different from the standard models of coalitional games, where all coalitions are feasible. We thank one of the anonymous reviewers for highlighting the work by Martell et al.

2 Preliminaries

Let $N = \{a_1, a_2, \ldots, a_n\}$ be a set of agents. A characteristic function game with *transferable utility (TU)* is a pair, (N, v), where $v : 2^N \to \mathbb{R}$ is the utility function that assigns a real value to every coalition $C \subseteq N$. As the utility is transferable, it is shared by all members of any coalition.

Conversely, a characteristic function game with *non-transferable utility (NTU)* is a pair, (N, V), where $V : 2^N \to \mathbb{R}^N$ is the utility function that for every coalition $C \subseteq N$ assigns a n-bit vector $(x_1, x_2, \ldots, x_n) \in \mathbb{R}^N$. Here, x_i denotes the individual utility of agent $a_i \in C$. We will refer to x_i in $V(C)$ as $V_i(C)$ and assume that agents outside a coalition have a zero utility, *i.e.*, $V_i(C) = 0$ for every $a_i \notin C, C \subseteq N$.

A *coalition structure* over N is a partition of the agents to coalitions, $CS = \{C_1, C_2, \ldots, C_k\}$ such that $\bigcup_{i\in\{1,\ldots,k\}} C_i = N$, and $C_i \cap C_j = \emptyset$ for any $i, j \in \{1, \ldots, k\}$ where $i \neq j$. The set of all coalition structures over N is denoted Π^N.

The (standard) utilitarian coalition structure generation (CSG) problem is defined as follows:

Definition 1 UTILITARIAN CSG: *Given a game (N, V), find a coalition structure CS^* that maximizes the sum of values of all agents. Formally:*

$$CS^* \in \arg\max_{CS\in\Pi^N} \left(\sum_{C_j\in CS} \sum_{i\in C_j} V_i(C_j) \right).$$

While the characteristic functions can encode any game, their exponential size means that they can be used to model only relatively small systems. Given this, a number of concise representations of coalitional games have been proposed in the literature.[11] In Section 5, we consider two such concise formalism: MC-nets [12] and Decision Diagrams [17] on which we focused in our research.

3 PROBLEM DEFINITIONS

In this paper, we consider two non-utilitarian CSG problems for the NTU games: EGALITARIAN CSG and BALANCED CSG. The former one assesses the performance of a coalition structure with respect to the value of smallest agent utility within it.

Definition 2 EGALITARIAN CSG: *Given a game with non-transferable utility, (N, V), find a coalition structure CS^* with the maximal value of the smallest agent utility. Formally:*

$$CS^* \in \arg\max_{CS\in\Pi^N} \left(\min_{i\in C_j\in CS} V_i(C_j) \right).$$

Conversely, the BALANCED CSG problem assesses the performance of a coalition structure with respect to the difference between the value of an agent with biggest utility and value of an agent with the smallest utility.

Definition 3 BALANCED CSG: *Given a game with non-transferable utility, (N, V), find a coalition structure CS^* that minimizes the difference between values of the smallest and the largest agents utilities. Formally:*

$$CS^* \in \arg\min_{CS\in\Pi^N} \left(\max_{i\in C_j\in CS} V_i(C_j) - \min_{i\in C_j\in CS} V_i(C_j) \right).$$

[11] For more on concise representations of coalitional games see the book by [4].

ACKNOWLEDGEMENTS

Tomasz Michalak and Michael Wooldridge were supported by the European Research Council under Advanced Grant 291528 ("RACE").

REFERENCES

[1] Karthik V. Aadithya, Tomasz P. Michalak, and Nicholas R. Jennings, 'Representation of coalitional games with algebraic decision diagrams', in *AAMAS'11: Proceedings of 12th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1121–1122, (2011).

[2] David Audretsch, Erik Lehmann, Aileen Richardson, and Silvio Vismara, *Globalization and Public Policy: A European Perspective*, 2015.

[3] Eilyan Y. Bitar, Enrique Baeyens, Pramod P. Khargonekar, Kameshwar Poolla, and Pravin Varaiya, 'Optimal sharing of quantity risk for a coalition of wind power producers facing nodal prices', in *Proceedings 31st IEEE American Control Conference*, (2012).

[4] 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).

[5] Yann Chevaleyre, Paul E. Dunne, Ulle Endriss, Jérôme Lang, Michel Lemaitre, Nicolas Maudet, Julian Padget, Steve Phelps, Juan A Rodriguez-Aguilar, and Paulo Sousa, 'Issues in multi-agent resource allocation', *Informatica*, (30), 3–31, (2006).

[6] Paul E. Dunne, Sarit Kraus, Efrat Manisterski, and Michael Wooldridge, 'Solving coalitional resource games', *Artificial Intelligence*, **174**(1), 20–50, (2010).

[7] Ulle Endriss and Nicolas Maudet, 'Welfare enginering in multiagent systems', in *Engineering Societies in the Agents World IV*, pp. 93–106, Springer-Verlag, (2004).

[8] Ulle Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni, 'Resource allocation in egalitarian agent societies', *Secondes Journes Francophones sur les Modles Formels d'Interaction (MFI-2003)*, 101–110, (2003).

[9] Ulle Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni, 'Negotiating socially optimal allocations of resources: An overview', *An Overview. Technical Report 2005/2, Department of Computing, Imperial College London*, (2005).

[10] Zhu Han and H. Vincent Poor, 'Coalition games with cooperative transmission: a cure for the curse of boundary nodes in selfish packet-forwarding wireless networks', *IEEE Transactions on Communications*, **57**(1), 203–213, (2009).

[11] Greg Hines, Talal Rahwan, and Nick R. Jennings, 'An anytime algorithm for finding the ϵ-core in nontransferable utility coalitional games', in *ECAI'12: Twentieth European Conference on Artificial Intelligence*, volume 242, p. 414, (2012).

[12] Samuel Ieong and Yoav Shoham, 'Marginal Contribution Nets: a Compact Representation Scheme for Coalitional Games', in *ACM EC '05: 6th ACM Conference on Electronic Commerce*, pp. 193–202, (2005).

[13] Zaheer Khan, Janne Lehtomäki, Matti Latva-aho, and Luiz A DaSilva, 'On selfish and altruistic coalition formation in cognitive radio networks', in *Fifth International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM-10)*, (2010).

[14] Silvano Martello, WR Pulleyblank, Paolo Toth, and Dominique De Werra, 'Balanced optimization problems', *Operations Research Letters*, **3**(5), 275–278, (1984).

[15] Joseph Plasmans, Jacob Engwerda, Bas Van Aarle, Giovanni Di Bartolomeo, and Tomasz Michalak, *Dynamic Modelling of Monetary and Fiscal Cooperation Among Nations*, Springer, 2006.

[16] Talal Rahwan, Tomasz P Michalak, Michael Wooldridge, and Nicholas R Jennings, 'Coalition structure generation: A survey', *Artificial Intelligence*, **229**, 139–174, (2015).

[17] Yuko Sakurai, Suguru Ueda, Atsushi Iwasaki, Shin-Ichi Minato, and Makoto Yokoo, 'A compact representation scheme of coalitional games based on multi-terminal zero-suppressed binary decision diagrams', in *Agents in Principle, Agents in Practice*, volume 7047 of *Lecture Notes in Computer Science*, 4–18, (2011).

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

doi:10.3233/978-1-61499-672-9-1740

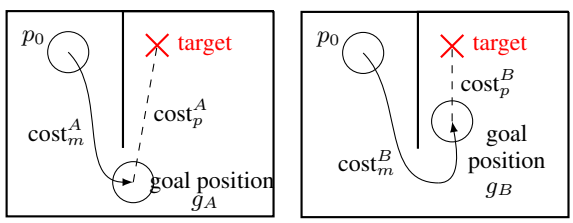
PA*: Optimal Path Planning for Perception Tasks

Tiago Pereira[1 2] and **Manuela Veloso**[1] and **António Moreira**[2 3]

Abstract. In this paper we introduce the problem of planning for perception of a target position. Given a sensing target, the robot has to move to a goal position from where the target can be perceived. Our algorithm minimizes the overall path cost as a function of both motion and perception costs, given an initial robot position and a sensing target. We contribute a heuristic search method, PA*, that efficiently searches for an optimal path. We prove the proposed heuristic is admissible, and introduce a new goal state stopping condition.

1 INTRODUCTION

The problem of motion planning has been widely studied before, but usually perception is not considered when determining the cost of a path. In this work we plan for motion and sensing, finding a path for the robot that minimizes both the distance traveled and the distance to a sensing target. As we show in Figure 1, a sensing target can be perceived from multiple locations. We show two possible paths that perceive the target from different goal positions, resulting in different motion and perception costs. Our proposed approach solves the problem of finding a path to a position g in a 2D gridmap that minimizes the total cost, $cost_m + \lambda cost_p$, where λ is a weight parameter.



(a) Path 1 (b) Path 2

Figure 1. The cost of a path is the sum of the motion cost ($cost_m$) and the perception cost ($cost_p$), proportional to path size and perception distance.

Many robotic applications consider perception separately from planning, with both being computed interleaved. It has been used for tasks as varied as robot localization [1] or object recognition [2], where perception is controlled in order to achieve a goal. However, recently perception got a more active role in planning. An example is object detection, where the next moves of the robot should be planned to maximize the likelihood of correct object classification [5]. Another example is the inspection problem. In order to determine a path that can sense multiple targets, a neural network approach was used to solve the NP-hard Watchman Routing Problem [4]. In the same topic, it has also been shown that perception planning and path planning can be solved together [3], selecting the most relevant perception tasks depending on the current goal.

Our contribution for solving the perception planning problem is the PA* algorithm, a heuristic search based on A* for gridmaps, and extended to deal with perception tasks. In the next section we explain the PA* algorithm in more detail.

2 PA*: OPTIMAL PERCEPTION PLANNING

We have to find a path ρ that not only minimizes distance traveled, but also minimizes the perception cost. The path is a sequence of positions in a grid, $\{s_0, s_1, ..., s_n\}$, such as the sensing target T is perceived from some position in ρ. The total cost of path ρ is

$$\text{cost}(\rho) = \text{cost}_m(\rho) + \lambda \text{cost}_p(\rho, T) \tag{1}$$

where λ is a weight parameter that trades-off motion and perception. The motion cost is proportional to the path size, and the perception cost increases with the minimum distance between ρ and T.

Theorem 1. *For the optimal ρ^*, the position that minimizes the distance from the path to sensing target T is the final position s_n.*

Proof. If there were another position s_i in the middle of the path that had the smallest distance to the target, then there would be a different path ending in s_i with minimal cost, contradicting the hypotheses that the path ρ^* from s_0 to s_n is the one that minimizes cost. □

In PA* the total cost is given by the sum of $g(s_0, n)$, the path distance from the starting position s_0 to the current node n, and $h(n, T)$, a heuristic of both the motion and perception costs from n to T.

$$f(n) = g(s_0, n) + h(n, T) \tag{2}$$

If the heuristic used is admissible, i.e., always less or equal than the true value, then the path returned is guaranteed to be optimal. Therefore, the choice for the heuristic is based on the euclidean distance between the current node and the target, without considering any obstacles, as shown in Figure 2. We assume that from position n the robot can still move to q, from where it senses the target.

$$h(n, T) = \min_q \Big(||n - q|| + \lambda c_p(||q - T||) \Big) \tag{3}$$

The sensor accuracy is modeled by c_p, with cost increasing with sensing distance. The variable α represents the percentage of the distance to the target that is traveled, and $1 - \alpha$ the percentage of the distance d that is sensed, where $d = ||n - T||$. In order to have the optimal solution, we need to find α that makes cost minimal.

$$\alpha^* = \underset{\alpha}{\text{argmin}}\, \alpha d + \lambda c_p\big((1 - \alpha)d\big) \tag{4}$$

[1] Carnegie Mellon University, USA, tpereira@cmu.edu, mmv@cs.cmu.edu
[2] INESC-TEC, INESC Technology and Science, Porto, Portugal
[3] Faculty of Engineering of University of Porto, Portugal, amoreira@fe.up.pt

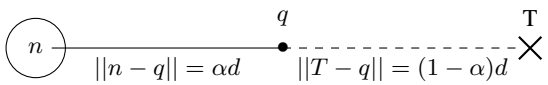


Figure 2. From robot position n and perception target T, without obstacles, the optimal goal lies in the straight line between those points. The image shows a solution with motion αd and sensing distance $(1-\alpha)d$.

Theorem 2. *If using the straight line solution in PA*, the heuristic is admissible, i.e. it is always less than the true cost.*

Proof. The direct distance between robot position n and target T is d. The motion distance plus the sensing distance equals d'. Because it might be a non-straight path to the target, $d' = d + \epsilon$, with $\epsilon \geq 0$. The robot moves a percentage of this path $\alpha d'$, and senses the rest $(1-\alpha)d'$. The overall cost is

$$\begin{aligned} &\alpha d' + \lambda c_p((1-\alpha)d') = \alpha(d+\epsilon) + \lambda c_p((1-\alpha)(d+\epsilon)) \\ &\geq \alpha d + \lambda c_p(d - \alpha d) \geq \alpha^* d + \lambda c_p((1-\alpha^*)d) \end{aligned} \quad (5)$$

proving the straight line solution yields minimum cost. □

For any specific perception function, it is possible to find the optimal sensing position as a function of the distance $||n-T||$. With α^* known before-hand, the heuristic is easy to use during search. In our model we assume circular omnidirectional sensing, with a limited range r_p, so the optimal sensing distance d_s^* is $(1-\alpha^*)d$ if $(1-\alpha^*)d \leq r_p$, or r_p if $(1-\alpha^*)d > r_p$. The heuristic becomes:

$$h(n,T) = (||n-T|| - d_s^*) + \lambda c_p(d_s^*) \quad (6)$$

It is possible to adapt the cost functions to the problem in hand (e.g., sensor and target properties), and then determine the heuristic for that specific problem just by solving for α^* offline.

2.1 Stopping Condition

In perception planning, it is possible to have paths with minimum cost that have a non zero heuristic at the goal state. We introduce a function that represents the cost of sensing from the current node:

$$f^s(n) = g(s_0, n) + \lambda c_p(||n-T||) \quad (7)$$

This function accounts for the cost of moving to the current node n, and sensing the target from that position. It takes no obstacles into consideration. Because the heuristic is admissible, $f^s(n) \geq f(n)$. When $f^s(n)$ and $f(n)$ are the same, sensing from the current position is equal to the optimal. At goal positions, besides testing for line-of-sight with ray casting, the following condition has to hold:

$$f^s(n) - f(n) = 0 \Leftrightarrow \lambda c_p(||n-T||) = h(n,T) \quad (8)$$

2.2 Addition of Expanded Nodes to Priority Queue

In the solution presented until now, nodes are expanded using a heuristic that estimates how much the robot should move in order to have an optimal path. However, as search continues, the estimated optimal path might not be feasible. We propose a solution that turns PA* into an optimal search algorithm. When a node n is added to the priority queue, its priority is given by $f(n)$. However, as node n is expanded for the first time, not only its neighbors will be added to the priority queue, but the node n itself will be added again, now using as priority the cost $f^s(n)$. This approaches makes it possible to "backtrack" to previous nodes. The stopping criteria becomes $priority(n) = f^s(n)$, with priority(n) being $f(n)$ for the first time a node is added in the priority queue, and $f^s(n)$ for the second time.

3 RESULTS

In order to test the performance of our algorithm, we tested it against a breadth-first search (BFS), comparing node expansion. Our results are based on 1470 feasible search instances. As expected theoretically and shown in Figure 3, PA* has always a better performance than BFS. For large λ, sensing cost has a big weight, so the robot will move as close as possible to the target. In the limit, all the state space is searched for large λ, trying to find a better solution. We can see that the best performance of PA* is for lower λ, which makes robot sense from further away.

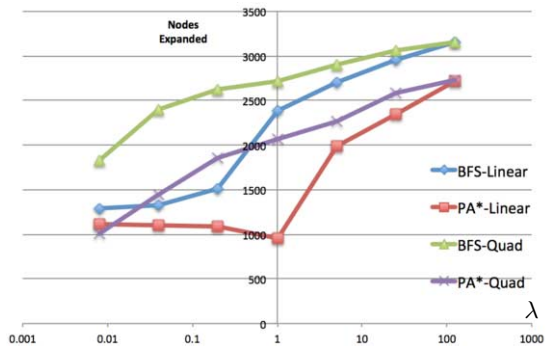


Figure 3. Number of nodes expanded by PA* and Breadth-First Search (BFS) as a function of λ, for linear and quadratic perception functions.

4 CONCLUSIONS

In this work we introduced the problem of motion planning for perception tasks, considering both motion and perception costs. We proposed PA*, an extension of A* for perception problems, and contributed heuristics to solve the planning problem, proving their admissibility. In the future we want to extend our approach with more complex perception cost functions.

ACKNOWLEDGEMENTS

This work is financed by the Carnegie Mellon Portugal Program grant SFRH/BD/52158/2013.

REFERENCES

[1] Joydeep Biswas and Manuela M Veloso, 'Localization and navigation of the cobots over long-term deployments', *The International Journal of Robotics Research*, (2013).

[2] Staffan Ekvall, Patric Jensfelt, and Danica Kragic, 'Integrating active mobile robot object recognition and slam in natural environments', in *International Conference on Intelligent Robots and Systems*, (2006).

[3] Jeremi Gancet and Simon Lacroix, 'Pg2p: A perception-guided path planning approach for long range autonomous navigation in unknown natural environments', in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (2003).

[4] Petr Janousek and Jan Faigl, 'Speeding up coverage queries in 3D multi-goal path planning', *IEEE International Conference on Robotics and Automation (ICRA)*, (2013).

[5] Christian Potthast and Gaurav S. Sukhatme, 'A probabilistic framework for next best view estimation in a cluttered environment', *Journal of Visual Communication and Image Representation*, (2014).

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

doi:10.3233/978-1-61499-672-9-1742

Cross-Domain Error Correction in Personality Prediction

Işıl Doğa Yakut Kılıç [1] and **Shimei Pan** [2]

Abstract. In this paper, we analyze domain bias in automated text-based personality prediction, and proposes a novel method to correct domain bias. The proposed approach is very general since it requires neither retraining a personality prediction system using examples from a new domain, nor any knowledge of the original training data used to develop the system. We conduct several experiments to evaluate the effectiveness of the method, and the findings indicate a significant improvement of prediction accuracy.

1 Introduction

Recently, an array of automated text-based personality analysis tools and services has emerged as the amount of user generated data, such as social media posts, have increased significantly [3]. Such tools analyze textual data authored by an individual (e.g., one's social media posts), and generate a personality profile based on the results (i.e. IBM's personality insights [2]). These results can be used to infer consumer behavior patterns or brand preferences [4] which can be used by marketing and public relations teams in their decision making processes [1]. Varying to contradictory results due to domain difference (e.g., social media posts versus emails) however render the tool ineffective and untrustworthy. Typically, personality traits are measured by standard psychometric surveys (e.g., IPIP survey) which are questionnaire-based, independent of domain, situation and text. Since personality traits obtained from psychometric survey are frequently used as the ground truth to train and evaluate an automated personal trait prediction system, the discrepancy in the predicted results is mainly due to domain overfitting in machine learning (ML) rather than situation-dependent personality change [5]. Thus, if an automated systems could predict personality accurately, in principle they will output the same personality values as those obtained from psychometric surveys, regardless of the situation. In reality, most personality tools are developed and trained using text samples from one particular domain (e.g., IBM's Personality Insights was trained using tweets [5]). Since most machine learning algorithms work under the assumption that the test data will be drawn from the same population as the training data, when an application domain is very different from the training domain —as is often the case in real-world applications— accuracy suffers.

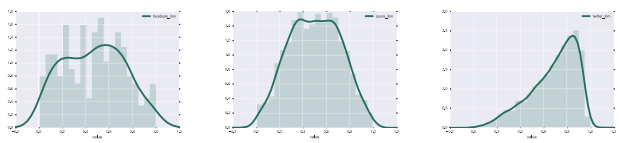
This short paper introduces a novel approach that identifies and reduces domain bias in text-based personality prediction systems. Since users of a personality prediction tool (e.g., a retailer) often do not have access to the training data (e.g., the dataset used to train IBM's Personality Insights), the proposed method employs a black-box approach that assumes access to neither the training data used to develop the tool nor training data from the new target domain (e.g., the application domain).

[1] University of Maryland, Baltimore County, USA, email: yakut1@umbc.edu
[2] University of Maryland, Baltimore County, USA, email: shimei@umbc.edu

2 Assessing Domain Bias in Personality Predictions

Domain difference in personality predictions can be evaluated at two levels: individual level and population level. At the individual level, for each person, we compute his trait scores based on his writing samples (e.g., social media posts). We compare the differences between the inferred trait scores and the ground truth personality of the same person. The larger the differences, the more severe the domain bias. We use Mean Square Error (MSE) as individual-level evaluation measure. Aside from observations at the individual level, the discrepancy at the population level can be shown between the distribution of predicted traits and the ground truth of a population. We use Kolmogorov-Smirnov test of equality between two distributions as the population-level evaluation metrics. Figure 1 shows the distributions of the Big Five Personality traits of the users from three datasets: Facebook, Quora, and Twitter. As can be seen, all three distributions are very different. The discrepancy of the predicted traits between Twitter and Quora (Figures 1b and 1c) is even more disturbing since the same set of individuals were used in collecting both datasets.

Figure 1: The Distributions of the Derived Conscientiousness Scores from three datasets: Facebook, Quora and Twitter.



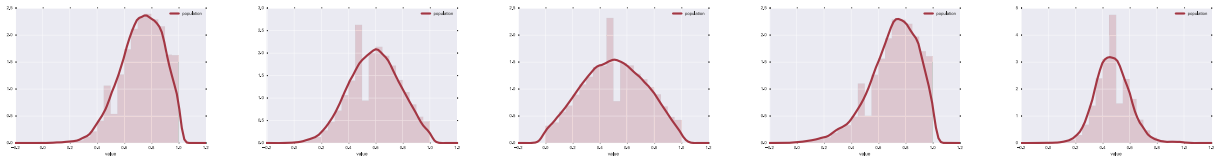
(a) Facebook (b) Quora (c) Twitter

3 Domain Bias Correction

Our method consists of two key operations: distribution parameter estimation and domain weight estimation based on domain similarity. Using these two processes, the method creates a linear transformation model based on the similarity between the application and the reference ground truth distributions.

The reference personality ground truth dataset is created to provide a domain-independent personality ground truth data from which we draw statistics to support domain bias correction. For this purpose, we use a personality ground truth dataset obtained from psychometric surveys (e.g., IPIP survey). The dataset is relatively big and contains 20,000+ people in total. Figure 2 shows the reference ground truth distributions of the Big Five personality traits. As one would expect, all of the personality traits have approximately normal distributions based on D'Augustino and Pearson's Normality test.

Figure 2: Reference ground truth distributions by aggregating the ground truth personality scores containing more than 20,000 data points. The data is aggregated from three separate sources: Personality Questionnaire results of individuals who are Facebook users, Mechanical Turkers and general public.



(a) Openness (b) Consciencious (c) Extraversion (d) Agreeableness (e) Emotional Range

Distribution Parameter Estimation In order to correct the predicted trait scores so that the inferred distributions fit the reference ground truth distributions, we first need to parametrize the distributions. Parameters of distributions can be used to scale and shift one distribution to fit another.Distribution parameters are calculated by first using Box-Cox Transformation and then applying Maximum Likelihood Estimation. Box-Cox transformation is a process that creates a normal distribution for given data using power functions. Given any distribution, the Box-Cox transformation will find an appropriate exponent λ and transform it into a normal distribution using the following formula:

$$y = \frac{(x^{\lambda} - 1)}{\lambda} \qquad (for\ \lambda > 0) \qquad (1)$$

$$log(x) \qquad (for\ \lambda = 0) \qquad (2)$$

where λ is the value that maximizes the log-likelihood function. Using the resulting stabilized data, we apply Maximum Likelihood Estimation to get the parameters.

The Maximum Likelihood Estimation (MLE) estimation of the distribution parameters μ (expectation) and σ^2 (variance) are calculated using mean *m* and standard deviation *s* of the data samples.

Domain Weight Estimation In cases where trait analysis results from multiple domains are available, we propose to weight the distribution parameters. The motivation for weighting application domains comes from the intuition that as the difference between predictions from an application domain and the reference ground truth increases, the prediction power of a system on that particular application domain decreases. To measure domain similarity between an application domain and the reference ground truth, we used the Kolmogorov-Smirnov statistic (KS statistic). The KS-statistic between two samples is given by

$$D_{n,n'} = \sup_x |F_{1,n}(x) - F_{2,n'}(x)| \qquad (3)$$

where F_1 and F_2 are the *empirical distribution functions* of the two samples of size n and n', with μ and σ as their mean and standard deviation, and sup is the *supremum function* (or the Least Upper Bound).

Linear Transformation With distribution parameters and associated weight for each domain acquired in previous steps, we can create a linear transformation function to map a source value to a corrected value. The linear transformation function is unique to the application datasets at hand. The goal of this transformation is to make the distributions of the corrected values more similar to the reference ground truth distribution.

Specifically, using the estimated reference population parameters and domain weights, we can now design the following linear function to derive the corrected trait values V_{*t}

$$V_{*t} = \sum_{i=1}^{n} [((S^i_{*t} - \mu_{S^i})/\sigma_{S^i} * \sigma_G + \mu_G) * D_{S^i,S^G}] \qquad (4)$$

where S^i is one of the n application datasets. The μ and σ values are calculated mean and standard deviation of an application dataset, and μ_G and σ_G are the ground truth parameters. The D_{S^i,S^G} is the domain weight calculated using Equation 3 and normalizing them among the application datasets. The formula calculates the corrected values by transforming application datasets and calculating the weighted means.

4 Results

We conducted various evaluations to demonstrate the effective of our method. At the individual level, the relative MSE reduction ranging from 23% to 30% on different datasets. At the population level, our results indicate that the corrected trait distributions are much more similar to the ground truth distributions than those before bias reduction for all the traits on all the test domains.

5 Conclusion

This short paper presented an analysis of domain difference on personality prediction results, and proposed a method to correct the bias that require no knowledge about the training data. The algorithm uses parameter estimations, domain weighting, and linear transformations to correct the domain bias. The effectiveness of the method has been demonstrated based on both individual-level and population-level evaluation metrics. The proposed method is very general and can be used to correct other domain-bias problems.

REFERENCES

[1] Jacob B Hirsh, Sonia K Kang, and Galen V Bodenhausen, 'Personalized persuasion: Tailoring persuasive appeals to recipients personality traits', *Psychological Science*, **23**(6), 578–581, (2012).
[2] IBM. Personality insights, IBM Watson developer cloud, 2015.
[3] Jianqiang Shen, Oliver Brdiczka, and Juan Liu, 'Understanding email writers: Personality prediction from email messages', in *User Modeling, Adaptation, and Personalization*, 318–330, Springer, (2013).
[4] Chao Yang, Shimei Pan, Jalal Mahmud, Huahai Yang, and Padmini Srinivasan, 'Using personal traits for brand preference prediction'.
[5] Michelle X Zhou, Fei Wang, Tom Zimmerman, Huahai Yang, Eben Haber, and Liang Gou, 'Computational discovery of personal traits from social multimedia', in *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pp. 1–6. IEEE, (2013).

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

doi:10.3233/978-1-61499-672-9-1744

Classical Planning with Communicative Actions

Tânia Marques and **Michael Rovatsos**[1]

Abstract. Explicit communication planning is an increasing necessity for agent systems. Furthermore, there has also been a renewed interest in using classical planning to perform this planning in addition to physical actions in a goal-directed way. Existing approaches, however, are not applicable to a broad spectrum of domains. We present several generic pre-processing strategies and adaptations of the Fast-Forward (FF) planner that we compare in two different domains in terms of time complexity.

1 Introduction

Recently, there has been an increasing interest in planning communication explicitly, contrasting with traditional approaches where communication is disregarded [3] or implicit [7, 12]. In fields where agents communicate with humans, the relevance of explicit planning communication over the task is obvious, but even in more artificial settings, this has been shown to be increasingly important, especially for tasks where information exchange is not naturally implicit [2] or tasks where the agents have different sets of beliefs [11].

Symbolic approaches to explicitly plan communication make it easy to incorporate the semantics of speech acts commonly used in agent communication languages [4], and allow for expressing the mechanics of communication in a domain-independent way that avoids enumerating all possible messages at design time. As a consequence, there has been renewed interest in symbolic approaches for dialogue planning (e.g. [9, 13]), and epistemic planning that can be employed in communicative situations to model uncertainty over other agents' mental states, which determine their expected behaviour towards the planning agent [5, 11].

In this paper, we investigate how a symbolic planner can be used to incorporate explicit communicative actions, while keeping the common assumptions of classical planning and its great strength of being generic and applicable to a broad range of domains. To focus on the fundamental problem of modeling communicative actions in such a way that they can be seamlessly combined with domain actions, we assume a deterministic setting where the other agent will always accept any request coming from the planning agent. While ignoring the uncertainty inherent to communication, this allows us to expose the complexity problems that arise even in this simple case without being biased by additional layers of complexity raised by contingency and probabilities. We focus on two broad classes of methods that extend classical planning to support communicative actions: pre-processing methods that can be easily implemented without changing the planner itself; and an integration of communication inside a classical planners that helps us exploit the machinery it provides. We present different variations of these methods and show how using the planner machinery is important to reduce the complexity of search, independently of the domain.

[1] The University of Edinburgh, email: {tmarques,mrovatso}@inf.ed.ac.uk

2 Integrating Communication

Our planning approach resembles speech act theory [1, 4], where utterances are actions whose effects are transfers of beliefs. If we assume two agents $a, b \in Agents$ and a set of actions A, divided into physical ground actions $Acts$ and communicative ground actions $Coms$. Then, a communicative action is a planning action defined by its name and a set of predicates as preconditions and effects taken from a set of *communicative fluents* such as `intends` (agent a has intention of performing act act) and `knows_wants` (agent b knows that agent a wants act to be performed) which allows us to build "mediating actions" following Cohen and Perrault's [4] concept, stating that a "mental" action will cause an intention in the hearer, and consequently to the actual performance of the action intended by the speaker. As arguments the communicative actions will take agents $requester, receiver \in Agents$ and any grounded physical action $request \in Acts$. We assume that only grounded physical actions can be taken as arguments. This makes our communicative actions *second-order actions*, but with a depth limit of one. Assuming this construction, actions *request* and *accept* would look as follows in PDDL [10]:

```
(:action request
:parameters   (?requester - agent ?receiver - agent ?request - act)
:precondition (and (not (= ?requester ?receiver))  (me ?requester))
:effect       (knows_wants ?receiver ?requester ?request))

(:action accept
:parameters   (?receiver - agent ?requester - agent ?request - act)
:precondition (and (not (= ?receiver ?requester))  (not (me ?recceiver))
(knows_wants ?receiver ?requester ?request))
:effect       (intends ?receiver ?request))
```

The integration of communicative actions can be done in a pre-processing phase or can be achieved by incorporating them inside the planning algorithm. The latter is more complex, because it assumes knowledge of the planner, but it allows the use of the machinery already provided by the planner.

2.1 Pre-Processing Methods

We present three possible pre-processing methods to add communication to classical planning.

- **Specific Communicative Actions:** This method automatically creates *request* and *accept* communicative actions that are tailored specifically to each physical actions contained in the domain file.
- **Grounded Physical Actions:** Here, the physical actions and the objects are used to generate all the possible objects of the type *act* that can be used in the communicative actions. An action is also generated which is an interface between the static object created and the possibility of using it in planning as a physical action.
- **Conditional Actions:** Similar to the previous one, but instead of using an action for every static object of the type *act*, we use an encoding in a single action that has *when* in its effects. Basically it works as a *switch* to decide which effects as an action are associated to a specific act object.

2.2 Adapting the planning algorithm

Two adaptations of the FF planner [8] were implemented, each based on a specific way of using the original planner:

- **Action Templates:** FF has already machinery in place to verify which objects can fill the schematas or not based on several factors. This method integrates communication into the FF code to allow the use of this machinery to define the communicative. actions to be created.
- **Relevant facts:** FF has also a reachability analysis to choose the relevant facts (facts that are reached with the actions provided). If an action does not reach any relevant fact, then we disregard it in this method by not creating the respective communicative action.

3 Experimental Results

3.1 Domains

We explored two domains: the Colored Trails [6] and the Deliver Letters domain. In the Colored Trails domain, agents play a game on a grid-like board with coloured squares. Each player needs to move to a goal location, but can only move to an adjacent square if it owns a chip of the same colour as that square, and the agent "spends" that chip. In ou scenario, we consider single row grid, where the requester is in the leftmost location and its goal is located at the opposite end. The requester has no chips, which are all with the receiver, so the agent will have to request the exchange of a number of chips equal to the length of the row. In the Deliver Letters domain, two agents are in a initial storage place and receive letters to deliver to the post offices. Every agent can only deliver letters to a certain post office, but may receive letters directed to any of them. For simplicity, we assume the requester only receives letters it cannot deliver, so it has to request their delivery from the other agent. All instances have two agents, three locations and the number of letters is a power of 2. Both the domains seem similar, but they are conceptually different. While the importance of a physical action in Colored Trails is defined by their potential use during planning, in Deliver Letters, the relevance of an action is completely defined by the object (the letter *ID*). There are eight instances for each domain that vary in terms of the size of the world (squares and letters) by powers of 2 from 4 to 512. These instances were run on a Intel Core i7-4770 CPU @ 3.40GHz.

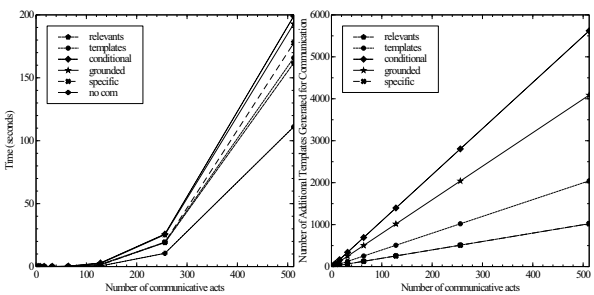


Figure 1. Time performance in Colored Trails, when the number of communicative actions equals the number of chips. Left: runtime; Right: number of additional templates created compared to using no communication.

The plots in figure 1 and 2 present the time taken for each approach and the number of additional grounded actions (templates) created when compared with having no communication (*no com*).

The results are consistent across all scenarios. The *grounded* physical actions and *conditional* cations approaches always produce the worst performance (and higher number of templates), while the *specific* communicative actions and *relevant* facts approach had the best performance. Time was mainly spent on the search, and adding communicative actions had the greatest effect on time when the actions can only be identified as relevant in the search as in Colored Trails.

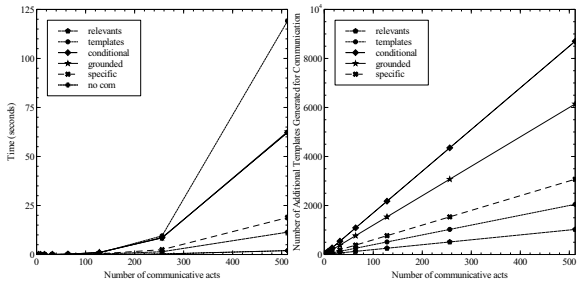


Figure 2. Plot with the time performance and additional templates obtained in the Delivering Letters domain.

4 Conclusions

We have shown that using specific actions or generating well-informed templates are the best approaches for integrating communication in classical planning. This is due to the link between communicative and physical actions which makes the search process critical in reducing the number of communicative actions to be considered. In the future, it will be interesting to find solutions to fully integrate communication in the plan search, and to test different methods in more complicated plan settings such as contingent planning.

ACKNOWLEDGEMENTS

The research presented in this paper has been funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 607062 *ESSENCE: Evolution of Shared Semantics in Computational Environments* (http://www.essence-network.com/).

REFERENCES

[1] J. L. Austin, *How to do things with words*, Oxford, Clarendon, 1962.
[2] T. Balch and R. C. Arkin, 'Communication in reactive multiagent robotic systems', *Autonomous Robots*, **1**(1), 27–52, (1994).
[3] R. I. Brafman and C. Domshlak, *From One to Many: Planning for Loosely Coupled Multiagent Systems*, 28–35, ICAPS'08, AAAI, 2008.
[4] P.R. Cohen and C.R. Perault, 'Elements of a plan-based theory of speech acts', *Cognitive Science*, **3**, 177–212, (1979).
[5] T. Engesser et al., *Cooperative epistemic multiagent planning with implicit coordination*, 68–77, DMAP'15, 2015.
[6] Y. Gal et al., *Colored Trails: a Formalism for Investigating Decision-Making in Strategic Environments*, 25–30, Workshop on Reasoning, Representation, and Learning in Computer Games, AAAI, 2005.
[7] M. R. Genesereth et al., 'Cooperation without communication', *Heuristic Programming Project*, 51–57, (1984).
[8] J. Hoffman, 'The fast-forward planning system', *AI*, **22**(3), 57–62, (2001).
[9] A. Koller and R. P. Petrick, 'Experiences with planning for natural language generation', *Computational Intelligence*, **27**(1), 23–40, (2011).
[10] D. McDermott et al., 'The PDDL planning domain definition language', *AIPS98 Planning Competition Comitee*, (1998).
[11] C. Muise et al., *Towards Team Formation via Automated Planning*, COIN'15, 2015.
[12] E. Pagello et al., 'Cooperative behaviors in multi-robot systems through implicit communication', *Robot Auton Syst*, **29**(1), 65–77, (1999).
[13] M. Steedman and R. P. A. Petrick, *Planning dialog actions*, 265–272, SIGdial'07, 2010.

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

doi:10.3233/978-1-61499-672-9-1746

Mixed Strategy Extraction from UCT Tree in Security Games

Jan Karwowski[1] and **Jacek Mańdziuk**[1 2]

Abstract. In this paper a simulation-based approach to finding optimal defender strategy in multi-act Security Games (SG) played on a graph is proposed. The method employs the Upper Confidence Bounds applied to Trees (UCT) algorithm which relies on massive simulations of possible game scenarios. Three different variants of the algorithm are presented and compared with each other as well as against the Mixed Integer Linear Program (MILP) exact solution in terms of computational efficiency and memory requirements. Experimental evaluation shows that the method has a few times lower memory demands and is faster than MILP approach in majority of test cases while preserving quality of the resulting mixed strategies.

1 INTRODUCTION

This paper describes simulation based approach to SG which can be used to find reasonably good strategies in SG with many rounds. Contrary to currently used methods which require a game to be represented in normal form or extensive form used in game theory, our method uses games represented by a set of game rules defining states, moves and results. This way we avoid memory demanding explicit payoff representations (typically used in SG) which assign a particular payoff for each possible move sequence and quickly become intractable as the number of possible move sequences grows exponentially with the number of rounds.

1.1 Security Games

SG is a field of science which applies mathematical models to patrolling schemes and other security operations in order to find optimal strategies for security forces against adversaries [3, 10, 7]. Possible applications include homeland security, fighting crime, or securing industrial objects. Usually SG are played by two sides: the defender (representing security forces) and the attacker (representing terrorists, criminals, etc.). The game is of imperfect information and players make their moves simultaneously. Instead of searching for the best move in a single game, the goal is to find a strategy (probability distribution of moves to play) for the defender that will maximize the expected reward. Almost all SG use Stackelberg Game [6] to model the game and are often referred as Stackelberg Security Games (SSG). Players in Stackelberg Game are asymmetric. One of them is called a Leader (the defender in SG) and the other one is a Follower (the attacker in SG). Asymmetry is introduced by the fact that the Follower knows the Leader's strategy before committing to their strategy. Such asymmetry models real-life cases in which the attacking side can observe the defending side sufficiently long to learn their strategy (probability distribution of actions).

Strong Stackelberg Equilibrium [6], which is the optimal solution for a defender can be expressed as a solution to bi-level optimization problem. The state-of-the-art solutions for SG transform this problem to a form suitable for popular Mathematical Programming solvers and use them to compute the strategy.

1.2 Upper Confidence Bounds applied to Trees

UCT method [5] is a variant of Monte Carlo Tree Search (MCTS) method used for searching for a single best move in perfect-information game tree [9, 8]. UCT uses large number of game simulations to estimate reward value for each move. Internally UCT keeps a tree consisting of game states (nodes) and actions or moves (edges). Each edge is labeled with two values: $Q(s,a)$ – current estimate of action a played in state s, and $N(s,a)$ – number of times the action was hitherto simulated. During simulations the method chooses actions based on the following formula:

$$a^* = \arg\max_{a \in A} \left\{ Q(s,a) + C\sqrt{(\ln N(s))/N(s,a)} \right\}, \qquad (1)$$

where $N(s) = \sum_a N(s,a)$ and C is the method's parameter. Formula (1), often called UCB1 [2], maintains a balance between exploitation of currently good estimated moves and exploration of rarely visited ones. After the simulation phase a move with the best Q estimation is chosen to be played.

2 Mixed-UCT METHOD

While UCT can be used for finding the best *single* move in perfect information game, the Mixed-UCT method, proposed in this paper,

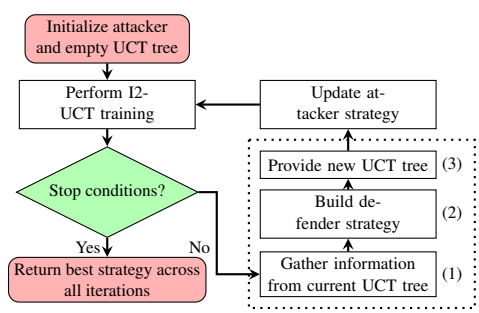


Figure 1. Outline of proposed Mixed-UCT method. Dotted line encompasses a generic procedure for the defender's strategy definition, which is the heart of the method.

[1] Faculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw, Poland, email {jan.karwowski, j.mandziuk}@mini.pw.edu.pl
[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore

is a UCT-based approach to finding the best mixed strategy for the defender in SSG. It relies on a UCT modification, referred to as I2-UCT, applicable to the task of finding the *best move* in imperfect-information games [4]. Basically I2-UCT extends vanilla UCT by adding a method of sampling (estimating) the current node (position) in imperfect information games before the actual UCT simulations take place (see [4] for details).

The proposed Mixed-UCT method iteratively applies I2-UCT procedure in order to find the best *mixed strategy* in SSG. The outline of the method is presented in Figure 1. For the sake of space limits we will solely concentrate on the most interesting part of the method, i.e. transformation of the data gathered by I2-UCT (intended to find a single best move) into coherent mixed strategy. Three variants of such a transformation are proposed. Each of them consists of three procedures denoted by (1), (2) and (3) in the figure.

Single tree variant uses UCT statistics gathered in a single (common) tree during training against many attacker strategies. **(1)** simply stores the UCT tree obtained from the recently completed training. **(2)** works as follows: in the stored UCT tree each path from the initial state (tree root node) to a terminal state (a leaf node) is assigned a weight being a product of visit counters of all edges (moves) on this path. Each such path represents a sequence of moves in the game – i.e. a pure strategy. The resulting mixed strategy is a probability distribution of these pure strategies, where probability of each sequence (pure strategy) is proportional to its weight – the weights are normalized to sum up to 1. **(3)** provides the tree stored by (1).

Best path variant collects best move sequences, one per each attacker it played against. **(1)** finds the best path from the root to a leaf in the trained tree according to Q estimates and stores a move sequence (pure strategy) extracted from this path. **(2)** returns probability distribution of these collected move sequences with probabilities proportional to number of their occurrences in these stored paths. **(3)** provides an empty tree for each I2-UCT training.

In **Tree slice** variant **(1)** prunes a tree by removing all paths from a given node that have Q estimate lower than 50% of the best Q value assigned to this node. Such pruned tree is stored and called a *tree slice*. **(2)** is a two-step procedure. First all stored *tree slices* are joined into one tree by summing visit counters in respective vertices, then a procedure from **Single tree** variant is applied to this joint tree. **(3)** provides an empty tree for each subsequent training.

	Single tree		Tree slice		Best seq.		Uniform	Optimal	
game	t [s]	*sc*	t [s]	*sc*	t [s]	*sc*	R	R	t [s]
1	2382	1	1559	0.96	1344	0.99	-10.46	0.54	28327
2	2614	1	2117	0.99	1587	1	-7.21	0.08	110
3	3188	1	7643	0.99	2486	0.97	-13.97	-4.27	17394
4	2115	1	3191	1	1379	1	-13.97	-4.5	18734
5	2325	1	3359	1	2484	1	-0.87	2.58	283
6	2058	1	3183	1	2151	1	-9.29	-1.06	18579
7	2241	1	4062	1	2897	1	-4.61	0.9	4695
8	1973	1	2364	1	1579	1	-13.91	-4.87	5823
9	2058	1	2007	1	1261	1	-13.84	-6	5915
10	1393	1	1448	0.99	1585	0.99	-0.81	0.79	5519
11	2579	1	2416	1	2093	1	-9.23	-2.85	5928
12	1713	1	2686	1	1775	1	-4.55	0.17	5149

Table 1. Defender's scores and computation times (t) of presented variants of Mixed-UCT in 12 test games (each averaged over 10 trials) compared with optimal solver-based strategy and the result of uniformly playing defender. *Score (sc)* represents the relative payoff assessment, which is placed on a scale from 0 (Uniform player) to 1 (Optimal player).

3 EXPERIMENTAL RESULTS

Each of the three variants of the method was tested against a benchmark set of 12 patrolling games defined on a graph. In each round both players were moving their units to adjacent vertices or let them remain in the current node. The goal of the attacker was to reach particular target node, while the defender's task was to catch the attacker (on the way or in the target) by being in the same vertex as them. Each game was played on a different graph with various reward and penalty distributions in the vertices.

Table 1 presents the average results of 10 game repetitions in each of the 12 test games, compared to the state-of-the-art method (MILP solutions calculated with SCIP [1]) and to a simple attacker that uses uniform strategy. Memory usage was comparable across all test cases and equal to about 5GB in MILP and about 1.2 in Mixed-UCT.

The results show that quality of Mixed-UCT solutions is close to optimal in all variants. Computational times of Mixed-UCT are shorter than those of the solver and vary between variants. **Tree slice** tends to be the slowest and **Best path** the fastest, in most of the cases.

4 CONCLUSIONS

This work introduces Mixed-UCT method and presents its initial experimental evaluation in SG domain. The results indicate general suitability of UCT-based methods to finding optimal strategies in SG. The proposed method, due to more compact (rule-based) game representation requires significantly less memory than MILP, which is currently the state-of-the-art approach. The proposed method is also faster than MILP solver. The results show that **Tree slice** version is inferior to the two other variants both in time and results quality, but at the same time, are inconclusive about superiority among the two other variants. Promising initial results lay foundations for further development of the method, in particular its evaluation on larger game graphs and games with multiple units on each playing side.

REFERENCES

[1] Tobias Achterberg, 'SCIP: Solving constraint integer programs', *Mathematical Programming Computation*, **1**(1), 1–41, (July 2009).

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer, 'Finite-time analysis of the multiarmed bandit problem', *Machine learning*, **47**(2-3), 235–256, (2002).

[3] Manish Jain, Jason Tsai, James Pita, Christopher Kiekintveld, Shyamsunder Rathi, Milind Tambe, and Fernando Ordóñez, 'Software assistants for randomized patrol planning for the lax airport police and the federal air marshal service', *Interfaces*, **40**(4), 267–290, (2010).

[4] Jan Karwowski and Jacek Mańdziuk, 'A new approach to security games', in *International Conference on Artificial Intelligence and Soft Computing*, volume 9120 of *LNAI*, 402–411, Springer-Verlag, (2015).

[5] Levente Kocsis and Csaba Szepesvári, 'Bandit based monte-carlo planning', in *Machine Learning: ECML 2006*, 282–293, Springer, (2006).

[6] George Leitmann, 'On generalized stackelberg strategies', *Journal of Optimization Theory and Applications*, **26**(4), 637–643, (1978).

[7] Eric Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer, 'PROTECT: A deployed game theoretic system to protect the ports of the United States', in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 13–20, (2012).

[8] Maciej Świechowski and Jacek Mańdziuk, 'Self-adaptation of playing strategies in General Game Playing', *Computational Intelligence and AI in Games, IEEE Transactions on*, **6**(4), 367–381, (2014).

[9] Karol Waledzik and Jacek Mańdziuk, 'An automatically-generated evaluation function in General Game Playing', *Computational Intelligence and AI in Games, IEEE Transactions on*, **6**(3), 258–270, (2014).

[10] Yue Yin, Bo An, and Manish Jain, 'Game-theoretic resource allocation for protecting large public events', in *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI14)*, pp. 826–834, (2014).

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

doi:10.3233/978-1-61499-672-9-1748

Boolean Negotiation Games

Nils Bulling[1] and **Koen V. Hindriks**[1]

Abstract. We propose a new strategic model of negotiation, called Boolean negotiation games. Our model is inspired by Boolean games and the alternating offers model of bargaining. It offers a computationally grounded model for studying properties of negotiation protocols in a qualitative setting. Boolean negotiation games can yield agreements that are more beneficial than stable solutions (Nash equilibria) of the underlying Boolean game.

1 Introduction

There are at least two prominent methodologies to analyse negotiations between agents: off-line, e.g. [8], using game theoretic techniques, and online, e.g. [2], using heuristic and evolutionary models. Most work in the game theoretic approach is based on Rubinstein's bargaining model of alternating offers [9], often making the assumptions of perfect rationality and perfect information.

We propose a new, compact model that allows us to investigate strategic aspects of negotiation protocols. The model we propose is inspired by Boolean games [4] (BG) which have become a popular model in the multi-agent domain. The many variants and extensions of BGs related to, e.g., knowledge [1], control and manipulation [6, 7], secret goals [5], dependencies [3], and pre-play negotiations about payoffs [10], just to name a few, make them an ideal starting point for our purposes. Boolean games, for example, allow us to also study aspects of control and power in a negotiation.

The main *contributions* of the work we present here consist of a model of negotiation called *Boolean negotiation game* (BNG) and a formal analysis of a protocol that does not allow repeating offers using this model. In the context of BGs the non-repetition of offers naturally yields finite games, which arise in many practical contexts. We introduce negotiation equilibria and are able to show that they always exist and that they can yield agreements which are more beneficial than the Nash equilibria of the underlying BG. In this context, the negotiation protocol plays a crucial role. A negotiation protocol gives rise to a specific unfolding of a BG with similarities to extensive games, but this unfolding is more general as it may not result in a complete agreement on all outcomes resulting in a smaller BG being played after the negotiation phase. As such, different properties of negotiation protocols greatly affect the game being played.

2 Boolean Negotiation Games

Boolean negotiation games (BNGs) allow players to interact in BGs by exchanging proposals sequentially. A negotiation protocol is imposed on a given BG affecting the possible actions of agents. Such a protocol adds a new layer of strategic interaction as not just the plain selection of a specific proposal is important but also the timing is of crucial significance, for example, in the setting where proposals cannot be offered more than once.

In order to introduce our model we first define the notion of a *generalized extensive Boolean game* (GBG). These games generalise the standard game theoretic notion of an extensive (Boolean) game [9]: (i) agents' actions are not limited to setting a single variable at a time but may *propose settings for multiple variables*, in principle including those which they *do not control*, (ii) at terminal histories, not all variables must be assigned a truth assignment. We assume a vector $\mathbf{Act}$ consisting of sets of actions. The idea is that agent i draws its actions from the set $\mathbf{Act}_i$, where the agent's valuations of propositions are typically also taken to be actions. We say that an action act_1 is said to *conflict* with action act_2 performed earlier during the game if the actions assign different truth values to a proposition. As in extensive form games, a protocol determines which agent's turn it is as well as the enabled actions at the current situation. A (possibly empty) sequence of actions is called a *history*. A *protocol* P maps a history h to a tuple (i, V) indicating that it is agent i's turn in h and that actions $V \subseteq \mathsf{Act}_i$ are enabled.

A GBG is a BG together with a protocol P. Given a protocol P, a *P-strategy* for agent i for that game is a (partial) function π_i the domain of which consists of all non-terminal P-histories h — histories consistent with P — at which it is player i's turn and which assigns a P-enabled action to such histories. A *profile* of P-strategies $\boldsymbol{\pi}$ yields a unique P-run $\rho_{\boldsymbol{\pi}}$, i.e. a sequence of actions consistent with the strategy.

The unique P-run $\rho_{\boldsymbol{\pi}}$ yielded by strategy profile $\boldsymbol{\pi}$ may not set the truth of all variables. In addition to that, some of the performed actions might be conflicting, e.g. a player may set a variable true and the same variable false later during the game. To determine an outcome of a GBG, we need to resolve these conflicts. The general rule that we use to determine the outcome is that any action that conflicts with an action performed later during the game is reverted and ignored in the computation of the outcome. Moreover, if the resulting outcome $\boldsymbol{\xi}$ is not a (full) valuation for all variables in Π, then the agents need to settle on the remaining variables in some other way. To settle on variables for which the agents did not settle on a valuation, the agents establish values by means of the ξ-reduced BG of the GBG, which is the BG in which each variable the truth value of which is defined by ξ is replaced by that truth value. Thus, a strategy of an agent in a GBG also needs to define which variables the agent sets in the resulting reduced BG. A formal treatment is out of the scope of this abstract, however, once this is defined formally it allows to introduce the notion of a *generalized equilibrium* taking into account that a strategy consists of two parts: a P-strategy and a strategy for the reduced BGs. We observe that a generalised equilibrium does not have to exist as some reduced BGs may not have a stable outcome. To see this, consider the trivial case in which the GBG consists of a single root node after which normal form games not having any Nash

[1] TU Delft, The Netherlands, email: {n.bulling, k.v.hindriks}@tudelft.nl.

equilibria are played: then, there is no generalised equilibrium either.

A *Boolean negotiation game* (BNG) is a special GBG in which the underlying protocol satisfies specific properties tailored towards negotiation settings. It is well accepted that there are at least two minimal requirements most negotiations should satisfy: (i) agents can make proposals and are able to respond to them; (ii) agents need to approve a possible agreement before it is concluded [9]. The latter point also implies that taking part in a negotiation should be individually rational for each agent. Based on these two properties we now introduce negotiation protocols and BNGs. In the negotiation setting, actions should be thought of as *proposals* made to the other players. As a consequence, a proposal conflicting with a proposal made earlier implicitly *rejects* the earlier proposal and serves the purpose of a *counter-proposal*. To implement point (ii) above, we identify a sub-class of protocols that requires all agents except for the agent who made the last proposal to explicitly approve the agreement that is on the table. Therefore, players which are happy with the current proposals can accept. Players have to be cautious, though, because if a proposal is made all other players could accept it which concludes the negotiation. To this end, the empty valuation $\xi_\emptyset$ is now interpreted as an accept action and is identified with the special action accept. In order to allow agents to accept, the protocol needs to support this. In that case we say that a protocol *supports agreeing*. Similarly a protocol *supports quitting* if a quit action allowing the agent to leave the negotiation is enabled after each non-terminal history. Then, a run is *closing* iff its final action is quit and it does not contain any further quit actions. There are different approaches how to deal with a quitting agent. We assume that in that case no deal is reached and the whole negotiation ends. Finally, a *negotiation protocol* (NP) is a protocol P which is turn-taking (i.e. players act in turns), supports agreeing as well as quitting and in which each P-run ρ is either agreeing containing no quit action, or is closing. Consequently, a BNG is a GBG including a negotiation protocol.

The definition of NPs is very general. It often makes sense to put some restrictions on the proposals that can be made. For example, it is usually not helpful to make the same proposal again and again. If a proposal has not resulted in an agreement it will, under reasonable assumptions, also not do so if it is made over again, only if the proposer counts on wearing out his/her opposite. The assumption is also reasonable in terms of real negotiations where it is often difficult to get back to a previously rejected proposal. Therefore, we focus on non-repeating protocols. A NP P is *non-repeating* if no proposal can be made twice with the exception of $\xi_\emptyset$ playing the role of the accept action. Furthermore, in order to investigate agents' interactions we focus on two types of protocols: one in which agents make proposals concerning their own variables only; and one where agents propose full valuations only. In the following let $\mathcal{N}$ be a BNG.

We are especially interested in the question whether agents have an $\mathcal{N}$-strategy profile $\boldsymbol{\sigma}$ which yields an agreement which is acceptable for all agents, given the possible outcomes of the underlying BG of $\mathcal{N}$. As usual in negotiation settings agents have a reservation value which corresponds to the payoff below which a player would refuse any proposal. A rather strict notion of reservation value would be a player's maxmin-strategy defining an outcome which the player can guarantee on its own. We call the corresponding reservation value the *maxmin reservation value*. In the strategic setting we consider here, it makes good sense to relate the reservation value to outcomes of Nash equilibria, as they give a payoff at least as good as the maxmin reservation value. In general, there can be more than one Nash equilibrium, therefore, we define a weak and a strong notion. The *greedy reservation value* (resp. *modest reservation value*) is the player's maximal (resp. minimal) payoff received by any Nash equilibrium. If a game does not have any Nash equilibria both reservation values are defined as the player's maxmin reservation value. We refer to *greedy* (resp. *modest*) agents as such which use as baseline their greedy (resp. modest) reservation values. Whereas the existence of a subgame perfect equilibrium in finite extensive form games is guaranteed by Kuhn's theorem (cf. [9]) this is not obvious in our setting. Indeed, it does not hold for the notion of generalised equilibrium put forward in the context of GBGs. The reason is that agents can quit the negotiation which results in a Boolean game over the not yet fixed variables. This BG may not have any Nash equilibria which also explains why a generalised equilibrium may not exist. In general the solution concept of generalised equilibrium is too strong as players base their decision on reservation values. Therefore, we introduce the weaker solution concept of a *negotiation equilibrium*. This solution concept makes no further assumption about the players' behavior if a (complete) agreement is not reached apart from assuming that each player can be ensured to receive a payoff at least as good as its reservation value in the resulting reduced BG. We can show that such an equilibrium always exists.

3 Future Work

We have proposed a formal framework for studying strategic aspects of negotiations in the compact framework of Boolean games. We used this to study negotiation protocols that do not allow repeating offers. There are many other interesting constraints on protocols that we could study within our framework. There are also many questions that we would like to study in more detail. For example, which protocols guarantee Pareto optimal outcomes? We are also particularly interested in studying negotiation with partial knowledge. Although a lot of research on negotiation with incomplete information has already been done, analysing this setting theoretically remains a challenge. Our model provides a starting point for creating a theoretical model of negotiation with incomplete information in future work.

REFERENCES

[1] Thomas Ågotnes, Paul Harrenstein, Wiebe van der Hoek, and Michael Wooldridge, 'Boolean games with epistemic goals', in *Logic, Rationality, and Interaction*, 1–14, Springer, (2013).

[2] Tim Baarslag, Mark J.C. Hendrikx, Koen V. Hindriks, and Catholijn M. Jonker, 'Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques', *Autonomous Agents and Multi-Agent Systems*, 1–50, (2015).

[3] Elise Bonzon, Marie-Christine Lagasquie-Schiex, and Jérôme Lang, 'Dependencies between players in boolean games', *International Journal of Approximate Reasoning*, **50**(6), 899–914, (2009).

[4] Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang, and Bruno Zanuttini, 'Boolean games revisited', in *ECAI*, pp. 265–269, (2006).

[5] Nils Bulling, Sujata Ghosh, and Rineke Verbrugge, 'Reaching your goals without spilling the beans: Boolean secrecy games', in *PRIMA*, pp. 37–53, Dunedin, New Zealand, (2013).

[6] Ulle Endriss, Sarit Kraus, Jérôme Lang, and Michael Wooldridge, 'Designing incentives for boolean games', in *AAMAS*, pp. 79–86, (2011).

[7] John Grant, Sarit Kraus, Michael Wooldridge, and Inon Zuckerman, 'Manipulating boolean games through communication', in *IJCAI*, pp. 210–215, (2011).

[8] Sarit Kraus, *Strategic Negotiation in Multiagent Environments*, MIT Press, 2001.

[9] Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory*, The MIT Press, 1994.

[10] Paolo Turrini, 'Endogenous boolean games', in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pp. 390–396. AAAI Press, (2013).

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

doi:10.3233/978-1-61499-672-9-1750

Toward Addressing Collusion Among Human Adversaries in Security Games

Shahrzad Gholami and **Bryan Wilder** and **Matthew Brown** and **Dana Thomas** and **Nicole Sintov** and **Milind Tambe**[1]

Abstract. Security agencies including the US Coast Guard, the Federal Air Marshal Service and the Los Angeles Airport police are several major domains that have been deploying Stackelberg security games and related algorithms to protect against a single adversary or multiple, independent adversaries strategically. However, there are a variety of real-world security domains where adversaries may benefit from colluding in their actions against the defender. Given the potential negative effect of these collusive actions, the defender has an incentive to break up collusion by playing off the self-interest of individual adversaries. This paper deals with problem of collusive security games for rational and bounded rational adversaries. The theoretical results verified with human subject experiments showed that behavior model which optimizes against bounded rational adversaries provides demonstrably better performing defender strategies against human subjects.

1 Introduction

Models and algorithms based on Stackelberg security games have been deployed by many security agencies including the US Coast Guard, the Federal Air Marshal Service, and Los Angeles International Airport [12] in order to protect against attacks by strategic adversaries in counter-terrorism settings. More recently, security games research has explored new domain such as wildlife protection, where planning of effective strategies is needed to tackle sustainability problems such as illegal poaching and illegal fishing [3]. Most of these previous works on security games assumes that different adversaries can be modeled independently [8]. However, there are many real-world security domains in which adversaries may collude in order to more effectively evade the defender. Three example domains are:

i) *Wildlife Protection Domain*: International trade of illicit wildlife products is growing rapidly and the most common types of illicitly traded wildlife products include elephant ivory, rhino horn, tiger parts, and caviar. Biodiversity loss, species extinction, invasive species introduction, and disease transmission resulting from illicit wildlife trade can all have disastrous impacts on the environment. Additionally, connections have been observed between illicit wildlife trade and organized crime as well as terrorist organizations, and thus activities such as poaching can serve to indirectly threaten national security [16]. Different forms of collusion have been observed among different groups of poachers. For example, these groups may coordinate to reduce the cost for storage, handling, and transportation of goods as well as to gain access to trade markets. This coordination can result in overall higher levels of poaching and damage to the environment [15].

ii) *Illegal Drug Trade:* Due to an ever growing demand for drugs, international organized crime syndicates have increased cooperation in order to facilitate drug trafficking, expand to distant markets, and evade local law enforcement [1]. In some cases, drug traders must cooperate with terrorist organizations to send drugs through particular areas. More broadly, expansion of global transportation networks and free trade has motivated cooperation between criminal organizations across different countries [11].

iii) *Finance - "rent-a-tribe" model:* Authorities in the US attempt to regulate payday lenders. These are lenders which offer extremely high interest rates to low-income borrowers, who cannot obtain loans from traditional banks. Recently, payday lenders have begun to operate in partnership with Native American tribes, which are exempt from state regulations. In this domain, the defender (a regulator) seeks a policy which prevents collusion between the adversaries (payday lenders and Native American tribes) [6].

Despite mounting evidence of the destructive influence of collusive behavior, strategies for preventing collusion have not been explored in the security games literature.

2 Background and Related Work

To better understand how humans make decisions regarding collusion, the following frameworks and theories are helpful in analysing the problem of collusive security game for rational and bounded rational adversaries.

Stackelberg Security Game model: The Stackelberg Security Game model, introduced almost a decade ago, has led to a large number of applications and has been discussed widely in the literature [12]. All of these works consider adversaries as independent entities and the goal is for a defender (leader) to protect a set of targets with a limited set of resources from a set of adversaries (followers)[2]. The defender commits to a strategy and the adversaries observe this strategy and each select a target to attack. The solution concept for security games involves computing a strong Stackelberg equilibrium which assumes that the adversaries maximize their own expected utility and break ties in favor of the defender. Security game models where an adversary is capable of attacking multiple targets simultaneously have been explored in [17]. To address cooperation between adversaries, [5] introduced a communication network based approach for adversaries to share their skills and form coalitions in

[1] University of Southern California, Los Angeles, CA 90089

[2] We use the convention in the security game literature where the defender is referred as "she" and an adversary is referred to as "he".

order to execute more attacks. However, no previous work on security games has conducted behavioral analysis or considered the bounded rationality of human adversaries in deciding whether to collude in the first place.

We now introduce key behavioral models and concepts that are useful for modeling and analyzing adversary behaviors in collusive security games.

Quantal Response and Subjective Utility Quantal Response: In real-world settings, human adversaries do not strictly maximize their expected utility, rather, they choose strategies stochastically [9]. Quantal Response (QR) model is a solution concept based on the assumption of bounded rationality. SUQR [10] has been proposed as an extension to QR and is the model used in this paper to predict the probability of attack at each target. In SUQR, subjective utility replaces expected utility and is defined as a linear combination of key domain features including the defender's coverage probability and the adversary's reward and penalty at each target. These features are assumed to be the most salient factors in the adversary's decision-making process.

Prospect Theory and Probability Weighting Functions: Another aspect of bounded rationality is the misperception by the adversary of key factors that influence decision making. Prospect Theory provides a descriptive model of how humans make decisions among alternative choices in the presence of probabilistic risk [7, 14]. According to this model, individuals overestimate low probability and underestimate high probability. Following this idea, literature in this domain proposes parametric models which capture different non-uniform weighting schemes including both inverse S-shaped as well as S-shaped probability curves. Based on these curves, the adversaries perceive a modified coverage probability and this fact can be exploited to the benefit of the defender. Human subject experiments have been conducted for security games to test both bounded rationality and probability perception [8], but such work never considered the type of collusive actions of concern in this paper.

Inequity Aversion Theory: Decisions regarding the interaction between humans in strategic settings can be influenced by the relative advantage of participants. According to Inequity Aversion theory humans are sensitive to inequity of outcome regardless of whether they are in the advantaged or disadvantaged situation and they make decisions in a way that minimizes inequity [4]. Inequity aversion has been widely studied in economics and psychology and is consistent with observations of human behavior in standard economic experiments such as the dictator game and ultimatum game in which the most common choice of people is to split the reward 50-50 [2]. Along these lines and contrary to the theoretical predictions, the IA theory also supports our experiments and analyses in security game domain.

Individualism Collectivism Analysis: Similarly, the personal attitudes and attributes of participants can also influence their interactions in strategic settings. A key characteristic is the well-established individualism-collectivism paradigm, which describes cultural differences in the likelihood of people to prioritize themselves versus their in-group. Specifically, those who identify as part of collectivistic cultures, compared to people in individualistic cultures, tend to identify as part of their in-groups, prioritize group-level goals, define most relationships with in-group members as communal, and are more self-effacing. Individualism-collectivism can be reliably measured using psychometrically-validated survey instruments [13].

3 Conclusion

We introduced a type of security games involving potential collusion among adversaries. Also we discussed the underlying frameworks and theories that are helpful in understanding how human makes decision regarding collusion in security games. Theoretical results verified by real human subject experiments showed that human adversaries are far from rational when deciding whether or not to collude and human behavioral model that incorporates bounded rationality of adversaries outperforms models assuming rational human adversaries.

REFERENCES

[1] Horace A Bartilow and Kihong Eom, 'Free traders and drug smugglers: The effects of trade openness on states' ability to combat drug trafficking', *Latin American Politics and Society*, **51**(2), 117–145, (2009).

[2] Nathan Berg, 'Behavioral economics', *21st century economics: A reference handbook*, **2**, 861–872, (2010).

[3] Fei Fang, Thanh H Nguyen, Rob Pickles, Wai Y Lam, Gopalasamy R Clements, Bo An, Amandeep Singh, Milind Tambe, and Andrew Lemieux, 'Deploying paws: Field optimization of the protection assistant for wildlife security', (2016).

[4] Ernst Fehr and Klaus M Schmidt, 'A theory of fairness, competition, and cooperation', *Quarterly journal of Economics*, 817–868, (1999).

[5] Qingyu Guo, Bo An, Yevgeniy Vorobeychik, Long Tran-Thanh, Jiarui Gan, and Chunyan Miao, 'Coalitional security games', (2016).

[6] Creola Johnson, 'America's first consumer financial watchdog is on a leash: Can the cfpb use its authority to declare payday-loan practices unfair, abusive, and deceptive', *Cath. UL Rev.*, **61**, 381, (2011).

[7] Daniel Kahneman and Amos Tversky, 'Prospect theory: An analysis of decision under risk', *Econometrica: Journal of the Econometric Society*, 263–291, (1979).

[8] Debarun Kar, Fei Fang, Francesco Delle Fave, Nicole Sintov, and Milind Tambe, 'a game of thrones: When human behavior models compete in repeated stackelberg security games', in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, (2015).

[9] Daniel L McFadden, 'Quantal choice analaysis: A survey', in *Annals of Economic and Social Measurement, Volume 5, number 4*, 363–390, NBER, (1976).

[10] Thanh Hong Nguyen, Rong Yang, Amos Azaria, Sarit Kraus, and Milind Tambe, 'Analyzing the effectiveness of adversary modeling in security games.', in *AAAI*, (2013).

[11] Andres Lopez Restrepo and Álvaro Camacho Guizado, 'From smugglers to warlords: twentieth century colombian drug traffickers', *Canadian Journal of Latin American and Caribbean Studies*, **28**(55-56), 249–275, (2003).

[12] Milind Tambe, *Security and game theory: Algorithms, deployed systems, lessons learned*, Cambridge University Press, 2011.

[13] Harry C Triandis and Michele J Gelfand, 'Converging measurement of horizontal and vertical individualism and collectivism.', *Journal of personality and social psychology*, **74**(1), 118, (1998).

[14] Amos Tversky and Daniel Kahneman, 'Advances in prospect theory: Cumulative representation of uncertainty', *Journal of Risk and uncertainty*, **5**(4), 297–323, (1992).

[15] Greg L Warchol, Linda L Zupan, and Willie Clack, 'Transnational criminality: An analysis of the illegal wildlife market in southern africa', *International Criminal Justice Review*, **13**(1), 1–27, (2003).

[16] Liana S Wyler and Pervaze A Sheikh, 'International illegal trade in wildlife: Threats and us policy'. DTIC Document, (2008).

[17] Zhengyu Yin, Dmytro Korzhyk, Christopher Kiekintveld, Vincent Conitzer, , and Milind Tambe, 'Stackelberg vs. nash in security games: Interchangeability, equivalence, and uniqueness', in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, (2010).

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

doi:10.3233/978-1-61499-672-9-1752

Detecting Communities Using Coordination Games : A Short Paper

Radhika Arava [1] and **Pradeep Varakantham**[2]

Abstract. Communities typically capture homophily as people of the same community share many common features. This paper is motivated by the problem of community detection in social networks, as it can help improve our understanding of the network topology. Given the selfish nature of humans to align with like-minded people, we employ game theoretic models and algorithms to detect communities in this paper. Specifically, we employ coordination games to represent interactions between individuals in a social network. We provide a novel and scalable two phased algorithm *NashOverlap* to compute an accurate overlapping community structure in the given network. We evaluate our algorithm against the best existing methods for community detection and show that our algorithm improves significantly on benchmark networks with respect to standard *normalised mutual information* measure.

1 Introduction

In social networks like Facebook, Google+, there can be overlap in user's high-school friends' circle and his university friends' circle. It is important to identify the overlapping community structure of a social network, as it helps in understanding the network topology, the spread of information, rumor in that network.

In this paper, we provide a novel, scalable two-phase algorithm *NashOverlap* to compute the overlapping community structure of a network. We evaluate our algorithm against the current state of the art and we find that it works significantly better than the best existing methods on the standard LFR benchmark networks [2]. To the best of our knowledge, this is the first game theory based scalable overlapping community detection algorithm that can detect an accurate community structure for a given social network.

2 Community Detection Problem

In a social network $G = (V, E, w)$, V is the set of vertices and E is the set of undirected edges that represents relationships between vertices in the network and $w : E \mapsto \mathbb{R}$ is the weight function on edges. A community in a social network is a non-empty connected subset of vertices with denser connections within themselves than with the rest of the network. Formally, if $C_j \subseteq V$ denotes a community j, then our goal is to identify the community structure $\Gamma = \{C_1, C_2, \ldots, C_r\}$, where $\bigcup_j C_j = V$.

[1] Nanyang Technological University,Singapore,email:radhika.arava@gmail.com
[2] Singapore Management University,Singapore,email:pradeepv@smu.edu.sg

3 Algorithm *NashOverlap*

We design a novel two-phased algorithm *NashOverlap* to detect overlapping communities in a social network as illustrated in Figure (1).

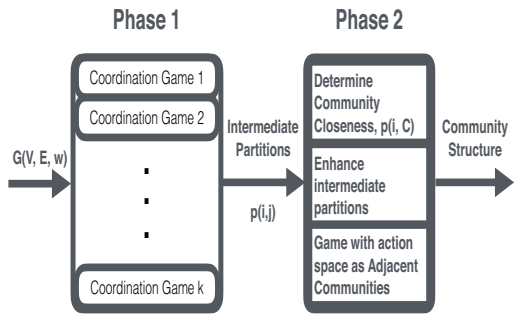


Figure 1. Algorithm: NashOverlap

3.1 First Phase

In this phase, we compute edge-closeness value $p(i, j)$ for each edge (i, j) and compute an intermediate partition of the network by solving k graph coordination games independently using local search.

Given a weighted network $G(V, E, w)$, we formulate each game, $\zeta^1 = (V, (S_i)_{i\in V}, (u_i)_{i\in V})$, where V is the set of players, S_i constitutes the set of r ($r \geq 2$) strategies for each player i and u_i is the utility function for each player i. Let $s_i \in \{1, \ldots, r\}$ denote the strategy that player i chooses; then $(s_i)_{i\in V}$ defines a strategy profile of the game. Let s_{-i} denote the strategies of all the other players other than i. Utility of a player i at a given strategy profile $s = (s_i, s_{-i})$ is given by

$$u_i(s_i, s_{-i}) = \sum_{\substack{j:(i,j)\in E \\ s_i = s_j}} t(i, j) \tag{1}$$

where tie-strength $t(i, j)$ of an edge (i, j) is defined as:

$$t(i, j) = w(i, j) + \sum_{\substack{k:(i,k)\in E \\ (k,j)\in E}} (w(i, k) + w(j, k)) . \tag{2}$$

We solve the above graph coordination game ζ^1 using local search in the following way. Initially, assign to each vertex $i \in V$, a strategy

picked uniformly from S_i. Pick a uniform random vertex-ordering and each vertex i during its turn picks the strategy $s_i \in S_i$, whichever gives him the maximum utility.

This is repeated with the same vertex-ordering until game ζ^1 converges to a Nash equilibrium. We define edge-closeness value $p(i,j)$ of an edge as the proportion of k games in which its vertices choose the same strategy at Nash equilibrium.

We now show that ζ^1 is a potential game with the potential function $\Phi_1 : \{1,2,\ldots,r\}^{|V|} \mapsto \mathbb{R}$ that is defined as:

$$\Phi_1 = \sum_{(i,j)\in E\setminus E_c} t(i,j),$$

where E_c is the set of cut-edges.

Theorem 1 *Φ_1 is a weighted potential function.*

We construct an intermediate partition which is the set of connected components containing only edges with $p(i,j) \geq 0.95$.

3.2 Second Phase

This phase takes care of the mistakenly identified edges in the intermediate partition of first phase using $p(i,j)$ values and outputs a stable overlapping community structure.

Definition 1 *Given a vertex i and a community C, define the community-closeness, $p(i,C)$ as the sum of edge-closeness values between i and C. That is,*

$$p(i,C) = \sum_{\substack{j:(i,j)\in E \\ j\in C}} p(i,j).$$

Intuitively, a vertex is said to be adjacent to a community, if it has any of its adjacent vertices in that community. If a vertex is not adjacent to a community, its community-closeness to that community is 0.

Consider a game $\zeta^2 = (V, (S_i)_{i\in V}, (u_i)_{i\in V})$ where S_i for each vertex i is the set of all communities, u_i is the sum of vertex $i's$ community-closeness values to all its adjacent communities.

We solve this graph coordination game using local search. Pick a random vertex-ordering and each vertex in its turn, computes its community-closeness to each of its adjacent communities. The vertex chooses to be a part of only those communities to which its community-closeness is at least α times its maximum community-closeness value, given that its utility increases with that decision. The vertex-ordering is repeated, until the game converges to a Nash equilibrium.

For a given overlapping community structure Γ, we shall duplicate every vertex in each of its overlapping communities. Let $\mathcal{C}_i$ be the set of communities of player i in a given overlapping community structure Γ. So, we put a copy of vertex i in each of its communities $C \in \mathcal{C}_i$. Observe that if the community structure is not overlapping, then $\mathcal{C}_i$ is of size 1 for each player i.

Let ζ^2 be a potential game with potential function $\Phi_2 : \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{|V|}\} \mapsto \mathbb{R}$ that is given by

$$\Phi_2 = \frac{1}{2} \cdot \sum_{\substack{i\in V \\ C\in\mathcal{C}_i}} p(i,C) \tag{3}$$

Theorem 2 *Φ_2 is a weighted potential function.*

α is a overlap parameter and lies in $[0,1]$. If $\alpha = 1$, the resulting community structure has no overlap.

3.3 Discussion

We show that the games ζ^1 and ζ^2 always converges to a local optimum. Though, the problem of computing the local optimum for all these games is PLS-Complete, we can allow for minor changes in the game parameters and show that we can compute a stable overlapping community structure in linear time (linear in the number of edges).

For every network, given its community structure, mixing parameter μ is defined as the fraction of a vertex's links that connect to vertices sharing no communities. For fuzzier networks($\mu > 0.5$), our algorithm is likely to converge to global optimum. However for networks with $\mu < 0.5$, our algorithm detects an accurate overlapping community structure. We choose $k = 100$ and vary α in $[0,1]$ for our experiments.

We compared the performance of our algorithm against current best algorithms: CFinder [5], OSLOM [4], COPRA [1], SLPA [6] with respect to a standard measure *Normalized Mutual Information* [3] on networks with varying network sizes, community sizes, overlapping membership and mixing parameter. *NashOverlap* outperformed all the other algorithms, with respect to NMI measure in all experimental settings. We provide results for a setting that is representative of the comparison results observed over all the parameter settings in Table 1.

om	SLPA	COPRA	CFinder	OSLOM	NashOv
2	0.87444	0.95993	0.36046	0.97164	0.968639
3	0.79596	0.89308	0.3135	0.89679	0.94078
4	0.75394	0.80646	0.37866	0.82748	0.877891
5	0.692	0.74409	0.36277	0.76214	0.787768
6	0.65122	0.69011	0.32862	0.7129	0.727489
7	0.60913	0.64374	0.34251	0.66702	0.678984
8	0.56703	0.59485	0.34953	0.62428	0.63227

Table 1. Comparison of NMI measure for all algorithms on 5000 vertex networks varying overlapping membership (*om*) from 2 to 8 and community sizes in the range [20, 100] for $\mu = 0.5$ and $on = 10\%$.

We believe that these results along with the simplicity of our algorithm can help in understanding the topology of networks efficiently.

REFERENCES

[1] Steve Gregory, 'Finding overlapping communities in networks by label propagation', *New Journal of Physics*, **12**(10), 103018, (oct 2010).

[2] Andrea Lancichinetti and Santo Fortunato, 'Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities', *Physical Review E*, **80**(1), 016118, (2009).

[3] Andrea Lancichinetti, Santo Fortunato, and Jnos Kertsz, 'Detecting the overlapping and hierarchical community structure in complex networks', *New Journal of Physics*, **11**(3), 033015, (mar 2009).

[4] Andrea Lancichinetti, Filippo Radicchi, Jos J. Ramasco, and Santo Fortunato, 'Finding statistically significant communities in networks', *PLoS ONE*, **6**(4), e18961, (apr 2011).

[5] Gergely Palla, Imre Dernyi, Ills Farkas, and Tams Vicsek, 'Uncovering the overlapping community structure of complex networks in nature and society', *Nature*, **435**(7043), 814–818, (june 2005).

[6] Jierui Xie, Xiaoming Liu, and Boleslaw K. Szymanski, 'SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process', *Proceedings of Data Mining technologies for computational collective intelligence workshop, at ICDM, Vancouver, CA*, 344–349, (dec 2011).

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

doi:10.3233/978-1-61499-672-9-1754

Pricing Options with Portfolio-Holding Trading Agents in Direct Double Auction

Sarvar Abdullaev[1] and **Peter McBurney**[2] and **Katarzyna Musial**[3]

Abstract. Options constitute integral part of modern financial trades, and are priced according to the risk associated with buying or selling certain asset in future. Financial literature mostly concentrates on risk-neutral methods of pricing options such as Black-Scholes model. However, it is an emerging field in option pricing theory to use trading agents with utility functions to determine the option's potential payoff for the agent. In this paper, we use one of such methodologies developed by Othman and Sandholm to design portfolio-holding agents that are endowed with popular option portfolios such as bullish spread, butterfly spread, straddle, etc to price options. Agents use their portfolios to evaluate how buying or selling certain option would change their current payoff structure, and form their orders based on this information. We also simulate these agents in a multi-unit direct double auction. The emerging prices are compared to risk-neutral prices under different market conditions. Through an appropriate endowment of option portfolios to agents, we can also mimic market conditions where the population of agents are bearish, bullish, neutral or non-neutral in their beliefs.

1 Introduction

Option is the type of financial derivative that enables its holder (i.e. owner) to buy or sell specified assets at certain future price to writer (i.e. issuer) of the option. Holder of the option buys for an additional cost (i.e option premium) determined by the market or the writer of the option. On the other hand, the writer of the option sells by taking future obligation to trade assets if holder chooses to exercise his right to buy or sell. Option contract must specify the underlying asset to be traded, its *volume*, *strike price* and expiration date. European options can be exercised only on their maturity date, while American options on any date until expiration. We will use only European options in the scope of this paper.

Traders can take different positions with options of different moneyness and create option portfolios which can align with their forecast and at same time limit their loss in case if their forecast is not true. Cohen counts more than 40 option portfolios and classifies them based on their market direction (i.e. bullishness or bearishness), volatility level, riskiness and gain [2]. Let us consider, butterfly spread. This type of spread involves taking positions in options with three different strike prices. In butterfly call spread, trader has an estimate that the price is not going to change sharply, so he wants to stay neutral. He buys 2 call options: one ITM with low K_1 and one OTM with high K_3. At the same time, he sells 2 ATM calls with K_2, where K_2 is halfway between the range of K_1 and K_3. This spread leads to a profit if the asset price will not go far from its current spot price. It will incur in fixed loss if the asset price changes sharply in either directions. Butterfly spread can be created using put options as well.

Pricing options is mostly based on Black-Scholes framework [1] which values options from the perspective of no arbitrage assumption. There is an analytical solution for finding the risk-neutral value of the option. However, in agent-oriented approach, traders can have different assumptions and pricing strategies (not necessarily risk-neutral) based on their private utility functions. Gerber and Pafum described risk-averse traders based on an exponential utility function which could produce a bid-ask spread around risk-neutral option prices [3]. We will use Othman and Sandholm's indifferent option pricing method which takes into account agent's inventory [7].

In this paper, we study how option prices may differ from the risk-neutral prices if the traders come to the market already endowed with some option portfolio. We developed an agent-based system which uses direct double auction mechanism to run option traders. We used inventory-based Logarithmic Market Scoring Rule (LMSR) option trader developed by Othman and Sandholm [7] to enable the option pricing based on the payoff structure of their current portfolio. We endow the LMSR traders with commonly used option portfolios such as *bullish spread*, *bearish spread*, *butterfly spread* etc. and run them in our proposed mechanism. This allows us to set up different sentiment in the market such as more bullish traders (or more bearish traders) and observe the resulted option prices in comparison to risk-neutral price. We explain our key findings from this experiment.

2 Portfolio-holding Trading Agent

In prediction markets, the agents are allowed to change the market maker's payoff structure for a corresponding payment. For example, if market maker is accepting bets for teams A and B on a football match, and his current payoff structure is (300,200) meaning that the aggregator has to pay \$300 in total if team A wins, and \$200 if team B wins. However one would like to bet on team A, and he expects to receive \$50 if his bet is achieved. The aggregator changes his payoff structure to (350,200) by accepting the bet, and he also needs to decide how he can charge the client for accepting his bet. The most common method for evaluating the cost of accepting the bet in prediction markets, LMSR [4] and it is defined as a cost function for the vector of payoffs $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ on the probability space of events $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$:

$$C(\mathbf{x}) = b \log \left(\sum_i \exp(x_i/b) \right) \quad (1)$$

[1] King's College London, UK, email: sarvar.abdullaev@kcl.ac.uk
[2] King's College London, UK, email: peter.mcburney@kcl.ac.uk
[3] Bournemouth University, UK, email: kmusialgabrys@bournemouth.ac.uk

where $b > 0$ is a liquidity parameter. The larger values of b produce tighter bid/ask spreads, but may also incur larger worst-case losses capped by $b \log(n)$ [8]. The agent who wishes to change the payoff from $\mathbf{x}$ to $\mathbf{y}$ has to pay the difference between the costs $C(\mathbf{y}) - C(\mathbf{x})$. In our above example, given that $b = 100$, the aggregator accepting bets must charge the client $C((350, 200)) - C((300, 200)) \approx \$39$ for the bet.

The same principle can be used for the option trader who holds a certain portfolio of options that generate certain payoff for different asset price outcomes in future. The agent can virtually simulate buying or selling particular type of option and compute the changes it makes to its payoff structure. For example, let agent take butterfly call spread with buying ITM call at strike $K_1 = 80$ and OTM call at $K_3 = 120$, and selling 2 ATM calls at $K_2 = 100$. We can compute his discounted payoffs for the range of possible prices where the asset price can end up at time T. Let this payoff structure be $\mathbf{x}$. Trader feels bullish and wants to buy one more call option at strike $K_4 = 130$, so his bid for buying an OTM option at $K_4 = 130$ can be computed from the difference of his payoff structures $C(\mathbf{y}) - C(\mathbf{x})$, and in our particular case, it is \$1.42 given that the liquidity parameter is $b = 2500$. We can also compute the Black-Scholes price of such option given parameters $T = 1, r = 0.05, S_0 = 100, \sigma = 0.02$ and it is \$2.52. This would mean that the trader places a bid for given OTM option less than its risk-neutral value.

3 Experimental Results

We have simulated the asset prices using the Geometric Brownian Motion with a calibrated parameters according to the historic data of NASDAQ-100 index in 2014. The daily mean drift is computed as $\mu = 0.0007$, and and the volatility is $\sigma = 0.0089$. The initial asset price is the same as NASDAQ-100 on 2 January 2014, $S_0 = \$3563.57$, and it is $S_T = \$3597.59$ at the end of the year. We also analyse option with strike \$3563.00 which expires in one year. We use only call options for the simulation, because put prices can be directly computed from the call price using *put-call parity* relationship. Mechanism simulates 365 trading days going up to the point when the option expires. Because mechanism is direct, it clears orders in one round and moves to the next trading day. Every trading day, the traders are re-instantiated with the same distribution of portfolios so they do not remember their previous choices.

LMSR traders create a positive bid-ask spread, which forbids them from trading if the market is uniformly populated with LMSR traders holding the same portfolio. Therefore LMSR trader should be also simulated in mixed groups each holding different set of portfolios, and thus produce different prices. It is also important to note that LMSR trader is deterministic in their pricing, because the only factor which affects their pricing decision is their portfolio and fixed range of events horizon that they use to compute their final payoff. Therefore two LMSR traders holding the same portfolio produce same bids or same asks. To make market more heterogeneous, I use multiple option portfolios to mimic neutral, non-neutral, bullish and bearish traders. Some of the option portfolios used are bearish spread, bullish spread, butterfly spread, ladder, strangle straddle, strip, etc [5]. After running several experiments with LMSR traders, we found out that liquidity $b = 100$ provides reasonable range of bids and asks which are likely to produce trades in the market. LMSR trader picks random quantities between -2000 and 2000 while submitting orders, so agent's decision to buy or sell is uniformly distributed. The negative quantities stand for asks, and the positive ones are bids.

Figure 1 shows the trade between neutral and non-neutral portfolio holders. It can be seen that the prices are volatile around Black-Scholes prices. This is explained using the deterministic nature of LMSR traders. The whole market consists of 2 neutral LMSR traders and 2 non-neutral LMSR traders who output all together 8 different pricing quotes, 4 for bids and 4 for asks. Because mechanism has very few choices to determine the clearing price among mostly homogeneous quotes, the option price for each trading day differs significantly. In the market of bearish traders as shown in Figure 2, the call options are initially underpriced, as they are considered less profitable for the traders expecting the drop in asset prices. However the prices cross the risk-neutral price only after option lives the half of its lifespan. This is because the payoff from the option becomes more certain, as the asset prices continue to grow defying the bearish belief of the trader.

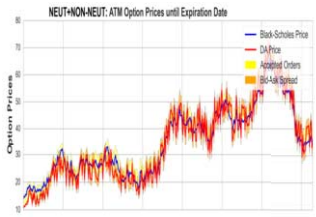


Figure 1. Option prices with neutral and non-neutral LMSR traders

Figure 2. Option prices with more bearish LMSR traders

4 Concluding Remarks

In this paper, we simulated LMSR-based option trading agents in direct double auction mechanism and were able to observe option prices under different population of traders with bearish, bullish, neutral and non-neutral portfolios. Our simulation results have shown that pricing option via double auctions is a valid technique as the obtained prices were close to risk-neutral solution if the market was truly populated with traders having different beliefs. Moreover, we saw that the neutral and the non-neutral traders create volatility as their portfolios consist of opposite payoff structures. Also in more bearish market population, the calls were initially underpriced until the option reached half-way its maturity. This approach can be further improved using other incentive compatible mechanisms such as McAfee's double auction [6] in future researches. Also the traders can be more sophisticated in making buying/selling decisions instead of randomly choosing either action.

REFERENCES

[1] Fischer Black and Myron Scholes, 'The pricing of options and corporate liabilities', *The journal of political economy*, 637–654, (1973).
[2] G. Cohen, *The Bible of Options Strategies: The Definitive Guide for Practical Trading Strategies*, Prentice Hall, 2005.
[3] Hans U Gerber and Gérard Pafum, 'Utility functions: from risk theory to finance', *North American Actuarial Journal*, **2**(3), 74–91, (1998).
[4] R. Hanson, 'Logarithmic market scoring rules for modular combinatorial information aggregation', *Journal of Prediction Markets*, **1**(1), 1–15, (2007).
[5] J. C. Hull, *Options, Futures and Other Derivatives*, Prentice Hall, 5 edn., 2001.
[6] R Preston McAfee, 'A dominant strategy double auction', *Journal of economic Theory*, **56**(2), 434–450, (1992).
[7] A. Othman and T. Sandholm, 'Inventory-Based Versus Prior-Based Options Trading Agents', *Algorithmic Finance*, **1**(2), 95–121, (2012).
[8] David M Pennock and Rahul Sami, 'Computational aspects of prediction markets', *Algorithmic game theory*, 651–674, (2007).

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

doi:10.3233/978-1-61499-672-9-1756

Efficient Semantic Tableau Generation for Abduction in Propositional Logic

Yifan Yang [1] and **Ricardo De Aldama** [1,2] and **Jamal Atif** [2] and **Isabelle Bloch** [1]

1 Introduction

Abduction is a backward chaining inference, finding the best explanations of an observation with regard to a knowledge base in a two-steps process: i) hypotheses generation, and ii) explanations selection according to a minimality criterion. A propositional abduction problem $\mathcal{P}$ is a tuple $\langle \mathcal{V}, \mathcal{O}, \mathcal{T} \rangle$ [3], where $\mathcal{V}$ is a set of variables, $\mathcal{T}$ denotes the background knowledge of an application domain, $\mathcal{O}$ represents the observation, which is not directly entailed by the knowledge base $\mathcal{T} \not\models \mathcal{O}$, and $\mathcal{H}$ is a hypothesis of the problem $\mathcal{P}$ if $\mathcal{T} \cup \{\mathcal{H}\} \models \mathcal{O}$.

Example 1
Let us consider $\mathcal{V} : \{r, c, s, w, n, d\}$, $\mathcal{O} : \{w \wedge d\}$, $\mathcal{T} = \{\phi_1, \cdots, \phi_4\} : \phi_1 = r \rightarrow w$, *if it rains, the ground will be wet;* $\phi_2 = s \rightarrow w$, *when the sprinkler is on, the ground will be wet;* $\phi_3 = c \rightarrow r \wedge d$, *if there are some heavy clouds, it will rain and get dark;* $\phi_4 = n \rightarrow d$, *during the night, it is dark. Potential explanations* $\mathcal{H}$ *can be* $c,\ r \wedge n,\ s \wedge n,\ r \wedge c,\ s \wedge c$, *and* $c \wedge n$.

In our paper, we restrict $\mathcal{O}$ and $\mathcal{H}$ to be a conjunction of a set of literals. A hypothesis is an explanation if it satisfies the consistency ($\mathcal{T} \cup \{\mathcal{H}\}$ is consistent) and explanatory ($\mathcal{H} \not\models \mathcal{O}$) conditions. Semantic minimality is one of the important criteria in propositional logic [5, 6] and Description Logics [2] to select the most general explanations. $\mathcal{H}$ is a semantic-minimal explanation if there does not exist an explanation $\mathcal{H}'$ of $\mathcal{P}$ such that $\mathcal{T} \cup \{\mathcal{H}\} \models \mathcal{H}'$. One concrete algorithm to solve the propositional abduction problem, based on semantic tableaux, is proposed in [1, 5] and [7]. In this paper, we focus on improving the performance of the computation process of abductive reasoning in propositional logic by developing an optimized semantic tableau method to find semantic minimal explanations for a propositional abduction problem.

2 An optimized semantic tableau for propositional logic

2.1 Optimized semantic tableau

Definition 2.1 (*Relevant formula*)
Given a literal l *and a formula* ϕ, ϕ *is a relevant formula for* l *if* l *appears in the normal form of* ϕ. *The set of relevant formulas for* l *is denoted by* $J(l)$.

In Example 1, we have for instance $J(r) = \{c \rightarrow r \wedge d\}$.

Definition 2.2 (*And-Or semantic tableau*)
Given a propositional abduction problem $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, \mathcal{T} \rangle$, *the And-Or semantic tableau, denoted by* $T_O(\mathcal{V}, \mathcal{O}, \mathcal{T})$ (T_O *for short), is an And-Or tree structure where an And-node (called rule node) consists of a formula* $\phi \in \mathcal{T}$ *or* $\neg O$ *and an Or-node (literal node) consists of a set of literals* $\{l_i\}$ *derived from its parent rule node. The level of a node is the distance from the root to this node.*

Remark 1. The semantic tableau proposed in [1], denoted by T_A, can be represented as a special structure using And-Or semantic tableau, denoted by T'_A: i) every literal node has a unique rule node as a child and this rule node is not necessarily a relevant formula of its parent literal node; ii) all the rule nodes appearing at the same level are identical formula; iii) all the rule nodes in the same branch are different formulas.

2.2 Tableau construction and hypotheses generation

We propose to construct a connected And-Or semantic tableau (denoted by T_G) exhibiting hierarchical connection structures such that, in contrast to the traditional tableau method, the construction process involves only relevant formulas in the knowledge base iteratively and intermediate hypotheses are generated in the node along with the development of the tableau. T_G is expanded step by step and terminated when a sub-tableau is closed or no more relevant formulas can be applied: i) level 0 is composed by the negation of the observation; ii) level 1 is composed by the union of negation of observation literals ($\neg o_i$) by applying expansion rules; iii) an edge derived from a rule node (ϕ) at level $2k$ to a literal node (a) at level $2k+1$ by applying expansion rules is denoted by $\phi \longrightarrow a$; iv) an edge derived from a literal node (a) at level $2k+1$ to a rule node (ϕ) at level $2(k+1)$ is denoted by $a \dashrightarrow \phi$, where ϕ is a relevant formula of $\neg a$ ($\phi \in J(\neg a)$). The connected And-Or semantic tableau of Example 1 is shown in Figure 1. An intermediate hypothesis $\mathcal{H}_{inter}$ of an elementary sub-tableau[3] is the complement of the literals in the open branches.

Definition 2.3 (*Hypothesis model*)
A hypothesis $\mathcal{H}$ *of* $\mathcal{P}$ *is the conjunction of* $\{\mathcal{H}^k_{inter} | k \in \{1, \cdots, n\}\}$ *where* $\{\mathcal{H}^k_{inter} | k \in \{1, \cdots, n\}\}$ *represents* n *terminal elementary sub-tableaux in a simple sub-tableau*[4] *of* T_O. *The intermediate hypothesis model* $\mathcal{M}(\mathcal{H}_{inter})$ *of a sub-tableau* S *consists of truth assignments of a set of literals* $\{l_i\}$, *where* l_i *is the complement of the literal of an open branch or the literal of a closed branch in the leaf*

[1] LTCI, CNRS, Télécom ParisTech, Université Paris-Saclay, Paris, France, email: {yifan.yang, isabelle.bloch}@telecom-paristech.fr, ricardo.de.aldama@gmail.com
[2] PSL, Université Paris-Dauphine, LAMSADE, UMR 7243, Paris, France, email: jamal.atif@dauphine.fr

[3] An elementary sub-tableau is a two-level sub-tableau rooted in a rule node.
[4] An And-Or semantic tableau is called *simple* if every literal node has at most a unique child node.

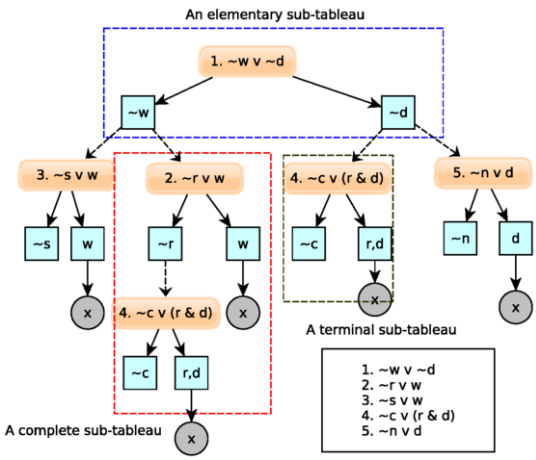


Figure 1. The tree-structure of the connected And-Or semantic tableau method for Example 1.

nodes including S and its ancestors. This set represents the semantic model of the intermediate hypothesis $\mathcal{H}_{inter}$ in the current elementary sub-tableau with respect to the ancestor elementary sub-tableau $\mathcal{T}_l = \bigwedge_{i \in I_l} \phi_i$. A hypothesis model is the union of the intermediate hypotheses models ($\mathcal{M}(\mathcal{H}) = \bigcup_{k \in \{1,\cdots,n\}} \mathcal{M}(\mathcal{H}^k_{inter})$).

A resolved knowledge base is a knowledge base with supplementary formulas derived by applying resolution on the original one $\mathcal{T}$, denoted by $\mathcal{T}_c$. $\mathcal{T}_c$ is represented by formulas rewritten in disjunctive normal form. For each literal l_i in $\mathcal{M}(\mathcal{H})$, we remove the conjunction containing its complement l_i in each formula. If one formula becomes an empty formula, the intermediate hypothesis is considered as inconsistent. We choose one rule node among alternative children nodes under each literal node, and the conjunction of the complements of all leaf literal nodes is one potential hypothesis.

In the exploration part, we update the initial explanation set once a new elementary sub-tableau is added in the tableau. The intermediate hypothesis model of this elementary sub-tableau is compared with the one in the hypotheses set. When $\mathcal{M}(\mathcal{H}_{inter}) \subseteq \mathcal{M}(\mathcal{H})$ is satisfied, the new elementary sub-tableau will be considered as a candidate to construct a hypothesis.

In Example 1, the initial hypotheses are $s \wedge n$, $s \wedge c$, $r \wedge n$ and $r \wedge c$. When the elementary sub-tableau associated with $\phi_3 = \neg c \vee (r \wedge d)$ is added below $\phi_1 = \neg r \vee w$, c is a new intermediate hypothesis. Therefore, a new explanation is a conjunction of the literal set obtained by replacing r by c in the initial hypothesis literal set $\{r, c\}$. The new explanation is c.

2.3 Soundness and completeness

Proposition 2. *The connected And-Or semantic tableau (T_G) and the semantic tableau proposed in [1] (T_A) provide the same sets of minimal hypotheses.*

Sketch of the proof: The proof relies on the following steps. We first show that T_G can be obtained from T'_A by a sequence Φ of transformations, combining deletion of branches that would lead to inconsistencies (closed branches), and "cut and paste", which consists in moving a semi-closed sub-tableau closer to the root of the tableau. We then prove that $\Phi(T'_A)$ and T'_A provide the same sets of hypotheses. Finally we prove that $\Phi(T'_A)$ and T_G provide the same sets of hypotheses.

3 Experimental evaluation

In this section, we present an empirical study to compare the computational cost of the proposed optimized semantic tableau with the algorithms in [1] and [4]. The theories used in our experiments to compare with semantic tableau method in [1] are generated in 3-CNF using a random generator[5]. The experiments are carried out by varying the number of formulas in the theory (namely for 10, 15 and 20 formulas). As evidenced in Table 1, our optimized semantic tableau based abduction has a significantly lower computation time for all the experiments compared to the semantic tableau method in [1].

Number of formulas	10	15	20
mean value (T_A)	999.64	3147.23	21435.59
std. deviation (T_A)	1083.80	2916.39	90244.14
mean value (T_G)	2.61	4.58	8.50
std. deviation (T_G)	4.29	7.24	13.63

Table 1. Computation cost of the two semantic tableau methods by varying the number of formulas in the theory.

The comparison between the semantic tableau method with the other abduction approaches using the diagnosis benchmark in [4] and Example 1 is shown in Table 2. The proposed method is competitive for the simple example presented in Example 1. However, the computation time increases along with the number of formulas in the theory. In the second example, the methods evaluated in [4] are more efficient. However, some explanations (c) are ignored for this example and the employed subset minimality is less restricted than semantic minimality.

Example (in ms)	T_A	T_G	ATMS	HS-DAG	MUS
Example 1	713	7	8	9	13
Diagnosis Horn theory	out of memory	85	17	17	30

Table 2. Computation time of all the methods.

Acknowledgements: This work has been supported by the French ANR project LOGIMA.

REFERENCES

[1] A. Aliseda-Llera, *Seeking explanations: abduction in logic, philosophy of science and artificial intelligence*, Ph.D. dissertation, University of Amsterdam, 1997.
[2] S. Colucci, T. Di Noia, E. Di Sciascio, F. M. Donini, and M. Mongiello, 'A uniform tableaux-based approach to concept abduction and contraction in $\mathcal{ALN}$', in *17th International Workshop on Description Logics (DL)*, volume 104, pp. 158–167, (2004).
[3] Nadia Creignou and Bruno Zanuttini, 'A complete classification of the complexity of propositional abduction', *SIAM Journal on Computing*, **36**(1), 207–229, (2006).
[4] R. Koitz and F. Wotawa, 'Finding explanations: an empirical evaluation of abductive diagnosis algorithms', in *International Workshop on Defeasible and Ampliative Reasoning (DARe-15)*, pp. 36–42, (2015).
[5] M. C. Mayer and F. Pirri, 'First order abduction via tableau and sequent calculi', *Logic Journal of IGPL*, **1**(1), 99–117, (1993).
[6] Gabriele Paul, 'Approaches to abductive reasoning: an overview', *Artificial Intelligence Review*, **7**(2), 109–152, (1993).
[7] F. Soler-Toscano, A. Nepomuceno-Fernández, and A. Aliseda-Llera, 'Abduction via C-tableaux and δ-resolution', *Journal of Applied Non-Classical Logics*, **19**(2), 211–225, (2009).

[5] Note that the generated inconsistent theories are not considered in our experiments.

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

doi:10.3233/978-1-61499-672-9-1758

Case-Based Classification on Hierarchical Structure of Formal Concept Analysis

Qi Zhang and **Chongyang Shi** and **Ping Sun** and **Zhengdong Niu**[1]

Abstract. We propose a novel Hierarchical CBC model (HCBC) based on Formal Concept Analysis (FCA). Firstly, Concept Lattice (CL), the hierarchical and conceptual structure in FCA, is adopted to represent cases. Thus a novel dynamic weight model is proposed from CL to measure similarities between cases and concepts. Then the similarity metric is applied to retrieve the top-K similar concepts which are used to vote for adaptive solutions for new cases by majority voting in case adaption. Experiments show our model shows good performance in terms of accuracy and outperforms the other classification methods.

1 INTRODUCTION

Case-Based Classification is a lazy methodology, aiming at retrieving similar situations to adapt solutions for new problems from the retained knowledge base through four stages, i.e. retrieval, reuse, revision and retention. Similarity assessment and case adaption are crucial to improve the performance of CBC. However, most similarity measures in CBC focus on surface features, which are provided as part of descriptions of cases and typically represented using attribute-value pairs [4]. Structural representations and Structural similarities, facilitating retrieving more relevant cases, are seldom considered. Besides, most existing structural representations rely on extensive domain knowledge, increasing computational expense and time complexity [4]. In this paper, we adopt CL in FCA [9] to represent cases in hierarchical conceptual clusters. In CL, all concepts are formed by naturally clustering of objects and attributes, indicating that CL is free of extend domain knowledge and shows implicit relevance among cases. Based on the hierarchical structure, a dynamic weight function for attributes is proposed to measure structural similarity, which is applied to searching the top-K similar concepts in retrieval stage. Actually, the structural similarity distinguishes HCBC from the existing FCA-based CBCs such as Frequency-weighted CBC (FCBC) in [5] where the extra information of the hierarchy and relation among concepts, facilitating similarity assessment, are not considered. Finally, the selected concepts vote for the final classification by majority voting, and the classified cases are retained and applied to update the CL. Experimental results on UCI data sets report HCBC achieves better performance in terms of accuracy than other well-known classification algorithms on categorical data.

2 PRELIMINARIES

Definition 1 *A formal context K is a triplet (G, M, I), where G and M denote nonempty, finite object set and attribute set respectively, and I is a binary relation of G and M, i.e. $I \subseteq G \times M$. For $g \in G$, $m \in M$, $(g, m) \in I$ denotes that object g has attribute m.*

Definition 2 *In formal context $K = (G, M, I)$, with $X \subseteq G$ and $B \subseteq M$, a pair of dual operators is defined below:*

$$X^* = \{m \in M \mid (g, m) \in I, \forall g \in X\}, \tag{1}$$

$$B^{'} = \{g \in G \mid (g, m) \in I, \forall m \in B\}. \tag{2}$$

Definition 3 *For $K = (G, M, I)$, a pair $C = (X, B)$ with $X \subseteq G$ and $B \subseteq M$ is called a formal concept if $X^* = B$ and $B^{'} = X$. X and B is respective the extent and intent of the concept. $L(G, M, I)$ denotes the set of all concepts and is called concept lattice.*

Thus, both $(X^{*'}, X^*)$ and $(B^{'}, B^{'*})$ are formal concepts. Formal concepts are generated by clustering cases through maximum inclusion between attribute set and object set, i.e. the dual operators. Concept lattice is a complete lattice and organized by a partial order $\leq$ given by: if $(X_1, B_1), (X_2, B_2) \in L(G, M, I)$,

$$(X_1, B_1) \leq (X_2, B_2) \Leftrightarrow X_1 \subseteq X_2 (B_1 \supseteq B_2). \tag{3}$$

If any concept (X_3, B_3) has $(X_1, B_1) \leq (X_3, B_3) \leq (X_2, B_2)$, then $(X_1, B_1) = (X_3, B_3)$ *or* $(X_2, B_2) = (X_3, B_3)$. We denote $(X_1, B_1) \prec (X_2, B_2)$.

The partial order organizes all concepts in a hierarchical structure, indicating concept lattice is informative that it depicts the generalization/specialization relationship between concepts.

3 HIERARCHICAL CASE-BASED CLASSIFICATION

This section firstly introduces some new definitions in CL. Then the weight model is proposed for similarity assessment. Afterwards, we introduce the workflow of HCBC.

Definition 4 *In concept lattice $L(G, M, I)$, concept C_T which satisfies: $\forall (X, B) \in L(G, M, I)$, $(X, B) \leq C_T$ is defined as top concept. (G, Φ) is the unique top concept in $L(G, M, I)$.*

Definition 5 *For $\forall (X, B) \in L(G, M, I)$, then*

$$L_{\downarrow}(X, B) = \{C \mid C \leq (X, B), C \in L(G, M, I)\}. \tag{4}$$

$(L_{\downarrow}(X, B), \leq)$ is called Down Induced Lattice (DIL).

$L_{\downarrow}(m)$ forms a lattice that aggregates all concepts whose attribute sets contain m. $L_{\downarrow}(m)$ helps to compute the frequencies of attributes in concept lattice by its cardinality.

Definition 6 *For C_i, C_j, $C_k \in L(G, M, I)$: if $C_i \leq C_j$,*

$$H(C_i, C_j) = \begin{cases} 0, \text{if } C_i = C_j, \\ 1, \text{if } C_i \prec C_j, \\ \min(H(C_i, C_k) + H(C_k, C_j)), \text{if } C_i \leq C_k \leq C_j. \end{cases}$$

$H(C_i, C_j)$ computes the minimal difference of hierarchies between concepts C_i and C_j. It reflects and quantizes generaliza-

[1] Beijing Institute of Technology, China, email: {zhangqi_cs, cy_shi, Peggy4Sun, zniu}@bit.edu.cn, Chongyang Shi is the corresponding author.

tion/specialization relation of concepts. Based on hierarchy measure and induced lattice, a novel weight metric for attributes is defined.

Definition 7 (Weight Metric) *For $C = (X, B) \in L(G, M, I)$ and $m \in B$, weight of attribute m in concept C is defined below:*

$$w(C,m) = \log \frac{|L|}{|L_{\downarrow}(m)|} * \frac{H([m], C_T) + H(C,[m]) + 1}{H(C,[m]) + 1}. \quad (5)$$

$|L|$ and $|L_{\downarrow}(m)|$ are cardinalities of the corresponding lattices. $[m]$ is short for (m', m'^{*}). In the metric, the logarithm part expresses that the more concepts contain an attribute, the less important the attribute is in similarity assessment. The fraction derives from the hierarchical relation and reflects structural relevances of concepts. The weight metric is applied in the similarity metric that is given by:

Definition 8 (Similarity Metric) *Let $C = (X, B)$ and $C \in L$, A is attribute set of a new problem or a case. Then similarity metric between concept C and the new case is defined below:*

$$S(C,A) = \frac{\sum_{m \in A \cap B} J(C,m)}{\sum_{m \in A \cap B} J(C,m) + \frac{1}{2} \sum_{i \in A - B} J(C,i) + \frac{1}{2} \sum_{j \in B - A} J(C,j)}. \quad (6)$$

m, i and j are attributes and $J(C,m)$ is shown below:

$$J(C,m) = \begin{cases} w(C,m), \text{ if } m \in B, \\ \frac{1}{|B|} \sum_{m \in B} w(C,m), \text{ if } m \notin B. \end{cases} \quad (7)$$

It is clear that $0 \leq S(C,A) \leq 1$ and $S(C,A) = 1$ if the attribute sets of the new case A and concept C are the same. The higher values the metric reaches, the more relevant the concept is to the new case. The metric is applied to retrieve top-K most similar concept in HCBC. Next we will introduce the workflow of HCBC below:

1. Generate formal context by *conceptual scale* [2], then construct CL to represent cases.
2. Retrieve the top-K similar concepts with the targeted new case by the proposed similarity metric.
3. Adapt an optimal solution for the new case by majority voting.
4. Retain the resolved cases and update the concept lattice by an incremental construction algorithm, Godin [3].

Since formal context is binary-valued, conceptual scale is adopted to transform multi-value data to binary formal context. To built CL, we apply the incremental algorithm, Godin, which incrementally constructs CL by generating new concepts and updating the old concepts. Then all concepts in CL are scanned to search the top-K similar concepts which are greatly influenced by similarity metric. Solutions or labels from the selected concepts are usually not identical, and it is more difficult to obtain the adaptive solutions from concepts than from cases. Thus we adopt majority voting to get the final solutions to resolve the new case which will be reserved to update the CL and facilitate solving next new cases.

4 EXPERIMENTS AND CONCLUSION

This section evaluates HCBC in term of accuracy on UCI data sets in which Spect and Monk problems are binary classification problems, and Car, Zoo, Balance and Hayes are multi-class problems. Experiments were conducted under Leave-One-Out cross validation in comparison with other excellent classification methods: k-Nearest Neighbour (kNN) [1], Sequential Minimal Optimization (SMO) [7] and C4.5 [8], Naive Bayes (NB) [6] and FCBC. We selected $k = 4$ in both kNN and HCBC, and the recommended settings from the corresponding authors are adopted for the others. The accuracy results are shown by percentages in Table 1. The best results of each data set are highlighted in bold. Thus, comments summarize below:

Table 1. Accuracy results of all the classification methods.

Datasets	*KNN*	*SMO*	*C4.5*	*NB*	*FCBC*	*HCBC*
Car	93.69	93.69	**96.99**	85.65	89.06	94.62
Zoo	94.06	**96.04**	91.1	94.06	92.38	95.05
Balance	85.76	90.56	78.08	**92.16**	78	88.32
Hayes	67.5	83.13	83.75	87.5	83.94	**90.63**
Spect	79.4	80.9	**84.64**	79.4	80.45	**84.64**
Monk-1	**100**	75	**100**	75	82.64	**100**
Monk-2	59.49	67.13	**97.22**	67.13	67.13	87.27
Monk-3	**100**	**100**	**100**	97.22	91.67	**100**
Average	84.99	85.81	91.34	84.76	83.16	**92.57**
Avg. Ranking	3.63	2.9	2.5	4.5	3.75	**1.63**

1. It is clear that HCBC has the highest average accuracy and ranks the topmost, which reportd that HCBC performs much better than the other methods overall.
2. It is observed that HCBC achieves much better results on the binary classification problems than the others and has comparable performance on the multi-class problems with the others.
3. HCBC shows all-sided advantages over FCBC and better performance than kNN, indicating our proposed similarity measure and adaption strategy are effective in CBC.

In conclusion, we have proposed hierarchical case-based classification based on concept lattice in FCA. The HCBC method combines similarity metric based on a novel dynamic weight model and majority voting to obtain adaptive solutions. It shows better classification performance than the selected methods. In future, the performance of HCBC on numeric data will be investigated and efficient retrieval algorithms are to be implemented.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (No. 61502033, 61272169, 61472034), Open Project from State Key Laboratory of Management and Control for Complex System (China,No.99S9021F4D), Basic Research Fund of Beijing Institute of Technology.

REFERENCES

[1] N. S. Altman, 'An introduction to kernel and nearest-neighbor nonparametric regression', *The American Statistician*, **46**(3), 175–185, (1992).
[2] Bernhard Ganter and Rudolf Wille, *Applications of Combinatorics and Graph Theory to the Biological and Social Sciences*, chapter Conceptual Scaling, 139–167, Springer US, New York, NY, 1989.
[3] Robert Godin, Rokia Missaoui, and Hassan Alaoui, 'Incremental concept formation algorithms based on galois (concept) lattices', *Computational Intelligence*, **11**, 246–267, (1995).
[4] RAMON LOPEZ DE MANTARAS, DAVID MCSHERRY, and DEREK et al. BRIDGE, 'Retrieval, reuse, revision and retention in case-based reasoning', *The Knowledge Engineering Review*, **20**, 215–240, (9 2005).
[5] Jirapond Muangprathub, Veera Boonjing, and Puntip Pattaraintakorn, 'A new case-based classification using incremental concept lattice knowledge', *Data Knowl. Eng.*, **83**, 39–53, (2013).
[6] Andrew Y. Ng and Michael I. Jordan, 'On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes', in *NIPS 14, December 3-8, 2001, Vancouver, Canada*, pp. 841–848, (2001).
[7] John C. Platt, 'Sequential minimal optimization: A fast algorithm for training support vector machines', Technical report, ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, (1998).
[8] J. Ross Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
[9] Rudolf Wille, 'Restructuring lattice theory: An approach based on hierarchies of concepts', in *Proceedings of the NATO Advanced Study Institute, Banff, Canada, August 28, 1981*, pp. 445–470. Springer, (1982).

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

doi:10.3233/978-1-61499-672-9-1760

Stochastic Area Pooling for Generic Convolutional Neural Network

Zhidong Deng[1] and **Zhenyang Wang**[2] and **Shiyao Wang**[3]

Abstract. This paper proposes a novel SAPNet model that incorporates a stochastic area pooling (SAP) method with a generic stacked T-shaped CNN architecture. In our SAP method, pooling area is randomly transformed and max pooling operation is then conducted on such areas, which are no longer regular identical fixed upright squares. It can be viewed as feature-level augmentation, substantially reducing model parameters while keeping generalization ability of CNN almost unchanged. Furthermore, we present a generic CNN architecture that structurally resembles three stacked T-shaped cubes. In such architecture, the number of kernels in convolutional layer preceding any pooling layer is doubled and all learnable weight layers are combined with batch normalization and dropout with a small ratio. Finally, on CIFAR-10, CIFAR-100, MNIST, and SVHN datasets, the experimental results show that our SAPNet requires fewer parameters than regular CNN models and still achieves superior recognition performances for all the four benchmarks.

1 Introduction

Data augmentation is a simple and useful way to improve recognition and generalization capabilities of CNN. It can help expand data space and further enhance diversity of input images. In this paper, we take advantage of such idea to feature-level augmentation to explore expansion on feature map space. Accordingly, we propose a novel stochastic area pooling (SAP) to improve feature representation capabilities of CNN. Instead of pooling on fixed pooling areas, pooling areas in our SAP method could be randomly translated, scaled, and rotated with a fluctuation. Jaderberg *et al*. [3] introduce similar idea in their spatial transformer network, which is to transform the input feature map. The difference is that the transformations in spatial transformer network are a deterministic function of the input and the transformation parameters and are explicitly optimized over, while the transformations in our SAP method are randomly drawn from pre-specified distributions that are not optimized over. In particular, Jaderberg *et al*.aim at generating scale and rotation invariance features by introducing an addition learning-based spatial transformer network, which requires a large amount of parameters and computational cost. SAP, however, plays the role of feature-level augmentation, having only a little extra computational burden.

Meanwhile, aiming to simplify the design procedure of new convolutional models, we present a generic CNN architecture that structurally looks like three stacked T-shaped cubes. In such architecture, the number of kernels in convolutional layer just preceding any pooling layer is doubled to overcome representational bottleneck [8]. All learnable weight layers are embedded with regularization of batch normalization, ReLU, and dropout with a small ratio of 0.1. Actually, all SAPNets with different convolutional kernel sizes have the same structures and similar parameter settings, which are demonstrated to be stable, reliable, and efficient.

[1] State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science, Tsinghua University, Beijing 100084, China, email: michael@tsinghua.edu.cn
[2] Tsinghua University, email: crazycry2010@gmail.com
[3] Tsinghua University, email: sy-wang14@mails.tsinghua.edu.cn

2 SAPNet Method

2.1 SAP Layer

SAP consists of the two consecutive steps: area selection and pooling operation. As the first step, a pooling area is generated using random affine transformation. Second, regular pooling operations such as max or average pooling are conducted on such randomly transformed areas, thus giving rise to output of SAP layer.

Instead of regular pooling on fixed square, the random pooling area of SAP is transformed by an affine mapping as shown in Figure 1(a).Suppose that θ denotes angle of rotation, $[c_x, c_y]$ coordinates of a center point, $[o_x, o_y]$ a shift vector, and $[s_x, s_y]$ scaling factors. Consequently, we form a 7-tuple of parameters $\{\theta, c_x, c_y, o_x, o_y, s_x, s_y\}$ so as to express such an affine transform, i.e.,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x \cos\theta & s_y \sin\theta & t_x \\ -s_y \sin\theta & s_x \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

$$t_x = c_x(1 - s_x \cos\theta) - c_y s_y \sin\theta + o_x \tag{2}$$

$$t_y = c_x(1 + s_y \sin\theta) - c_y s_x \cos\theta + o_y \tag{3}$$

Actually, pooling area in SAP is produced by assigning a random distribution to each parameter. The Gaussian distribution that each parameter must satisfy is given below,

$$\theta \sim N(0, \sigma_\theta), \tag{4}$$

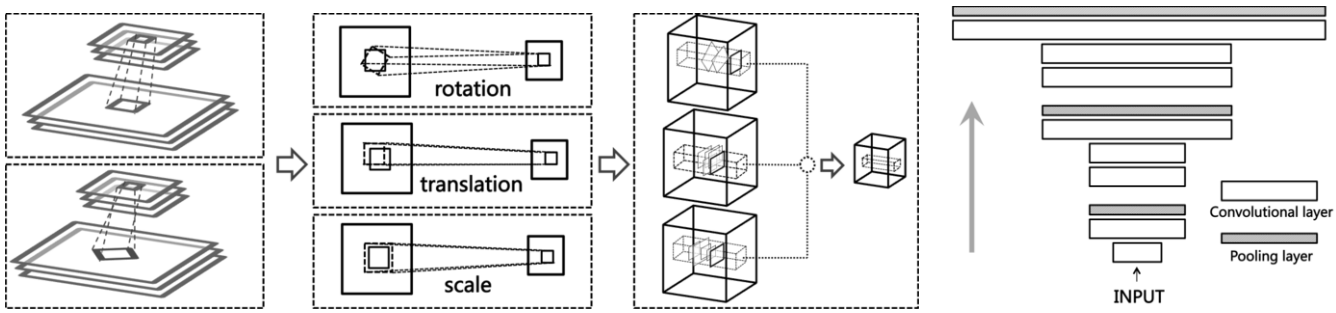
$$c_x, c_y \sim N(0.5d, \sigma_c) \tag{5}$$

$$s_x, s_y \sim N(1, \sigma_s) \tag{6}$$

where standard deviations σ_θ, σ_c and σ_s are called hyper-parameters, which should be pre-specified before training, and d indicates the height or width of incoming feature map.

Note that pixels within random pooling areas are usually transformed onto non-integral boundaries. Thus bilinear interpolation is required before pooling operation.

During every forward pass, a set of different affine transformation parameters are randomly produced via Equations (4), (5), and (6).



(a) Pooling areas of SAP randomly transformed (b) Three stacked T-shaped cubes

Figure 1. Schematic diagram of stochastic area pooling for generic CNN architecture with three stacked T-shaped cubes.

After that, these transformation parameters or the resulting pooling areas in the SAP layer remain unchanged and are not optimized over during either forward pass or back-propagation. Sequentially, max pooling is then done on such areas. Apparently, our SAP layer has the same forward and backward pass as that for regular CNNs with max pooling except for transformed pooling area.

In the test phase, we set the standard deviations of parameters $\sigma_\theta = 0$, $\sigma_c = 0$ and $\sigma_s = 0$ for the SAP layers, which means that we employ the same fixed upright pooling squares as that of regular pooling method.

2.2 Generic Stacked T-shaped CNN Architecture

In addition to pooling strategies, design of CNN architecture is also crucial to improvements in capabilities of hierarchical feature representation. Basically, this used to be an empirical trick. In this paper, we deliberately design a generic CNN architecture that structurally looks like three stacked T-shaped cubes (in Figure 1(b)), together with two SAP layers as separator. Note that the number of kernels in the convolutional layer preceding each pooling layer is always doubly expanded. We call such a general model as SAPNet. Specifically, every SAPNets comprises three ensembles, each of ensembles containing three convolutional layers, and one softmax layer. A total of 9 convolutional layers are separated by two SAP layers and one global average pooling layer.

In SAPNet, all convolutional layers have small receptive field of 3×3 with padding of 1 so as to preserve spatial resolution. In two SAP layers, we take 2×2 initial pooling areas with stride of 2. Following the last convolutional layer, one global average pooling is adopted. Similarly, we exploit a softmax layer as final output. Additionally, all learnable weight layers are embedded with regularization of batch normalization (BN), ReLU nonlinearity, and dropout with a small ratio of 0.1.

3 Experimental Results

We evaluated our approach on CIFAR-10, CIFAR-100, MINST, and SVHN datasets. The experimental results (Table 1) show that our SAPNet requires fewer parameters than regular CNN models. Our SAPNet yields a state of the art test error of 5.57% on CIFAR-10 with data augmentation. On CIFAR-100, the SAPNet-64 achieves a test error of 27.59%. On MNIST, the SAPNet with 0.76 million parameters receives the best result of 0.29%. Our SAPNet obtains a test error of 1.71% on SVHN benchmark, which is ranked second in all the existing machine learning models.

Table 1. Comparison with existing models on CIFAR-10, CIFAR-10 with data augmentation, CIFAR-100, MNIST, and SVHN.

Model	CIFAR-10	CIFAR-10+	CIFAR-100	MNIST	SVHN
Maxout [2]	11.68	9.38	38.57	0.45	2.47
NIN [7]	10.41	8.81	35.68	0.47	2.35
DSN [5]	9.69	7.97	34.57	0.39	1.92
RCNN [6]	8.69	7.09	31.75	0.31	1.77
Tree+Max-Avg [4]	7.62	6.05	32.37	0.31	**1.69**
ELU-Network [1]	6.55	-	**24.28**	-	-
SAPNet	**6.36**	**5.57**	27.59	**0.29**	1.71

ACKNOWLEDGEMENTS

The authors would like to be grateful to the anonymous reviewers for their valuable comments that considerably contributed to improve this paper. This work was supported in part by the National Science Foundation of China (NSFC) under Grant Nos. 91420106, 90820305, and 60775040, and by the National High-Tech R&D Program of China under Grant No. 2012AA041402.

REFERENCES

[1] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, 'Fast and accurate deep network learning by exponential linear units (elus)', *arXiv preprint arXiv:1511.07289*, (2015).
[2] Ian J Goodfellow, David Warde-farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio, 'Maxout networks', in *ICML*, pp. 1319–1327, (2013).
[3] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al., 'Spatial transformer networks', in *Advances in Neural Information Processing Systems*, pp. 2008–2016, (2015).
[4] Chen-Yu Lee, Patrick W Gallagher, and Zhuowen Tu, 'Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree', *arXiv preprint arXiv:1509.08985*, (2015).
[5] ChenYu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu, 'Deeply-supervised nets', in *AISTATS*, pp. 562–570, (2015).
[6] Ming Liang and Xiaolin Hu, 'Recurrent convolutional neural network for object recognition', in *CVPR*, pp. 3367–3375, (2015).
[7] Min Lin, Qiang Chen, and Shuicheng Yan, 'Network in network', in *ICLR*, (2014).
[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna, 'Rethinking the inception architecture for computer vision', *arXiv preprint arXiv:1512.00567*, (2015).

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

doi:10.3233/978-1-61499-672-9-1762

Hole in One: Using Qualitative Reasoning for Solving Hard Physical Puzzle Problems

Xiaoyu Ge[1] and **Jae Hee Lee**[2] and **Jochen Renz**[1] and **Peng Zhang**[1]

Abstract. The capability of determining the right sequence of physical actions to achieve a given task is essential for AI that interacts with the physical world. The great difficulty in developing this capability has two main causes: (1) the world is continuous and therefore the action space is infinite, (2) due to noisy perception, we do not know the exact physical properties of our environment and therefore cannot precisely simulate the consequences of a physical action.

In this paper we define a realistic physical action selection problem that has many features common to these kind of problems, the minigolf hole-in-one problem: given a two-dimensional minigolf-like obstacle course, a ball and a hole, determine a single shot that hits the ball into the hole. We assume gravity as well as noisy perception of the environment. We present a method that solves this problem similar to how humans are approaching these problems, by using qualitative reasoning and mental simulation, combined with sampling of actions in the real environment and adjusting the internal knowledge based on observing the actual outcome of sampled actions. We evaluate our method using difficult minigolf levels that require the ball to bounce at several objects in order to hit the hole and compare with existing methods.

1 INTRODUCTION

One of the grand visions of Artificial Intelligence is to build robots with similar everyday capabilities as humans, who can live among us and assist us with many of our daily tasks. To progress towards more capable and more human-like robots, we need to develop methods and technology that allow robots to successfully interact with their environment. When we humans are faced with a "*physical action selection problem*", i.e., a problem that requires selecting a physical action that achieves the desired goal, we are very good at coming up with a qualitative solution and with a qualitative prediction of the consequences of an action. We have selected one particular physical action selection problem that is an actual real-world problem and that covers many common aspects of physical action selection problems. We call our problem the "*Hole-in-One*" problem in reference to the problem in mini golf of identifying and executing a shot that sinks the ball with this single shot. Variants of the hole-in-one problem occur frequently, not just in mini golf, in Pachinko, in pool billiard, curling or in a multitude of physics-based video games such as Angry Birds, but also in many everyday situations.

What these problems have in common is that the selected action can be one of infinitely many possible force vectors. Once a force vector is given and the physical properties of all objects and the environment are exactly known, it is possible to compute the exact trajectory of the ball and to see if that force vector solves the problem. However, we have to identify a force vector out of infinitely many possibilities that solves the problem. While a geometrical or analytical solution of these problems is typically not possible if the obstacle course is non-trivial, humans are very successful in solving these kind of problems. These problems become even harder to solve when we do not know the exact physical setting. We often only know what we can see and our perception is thus the limiting factor in what we know about the physical setting. Because of the uncertainty about the physical environment, potential solutions to the problem need to be executed in the actual environment before we can be sure that it is a solution. If it is no solution, we need to find ways of adjusting it so that it will eventually lead to a solution.

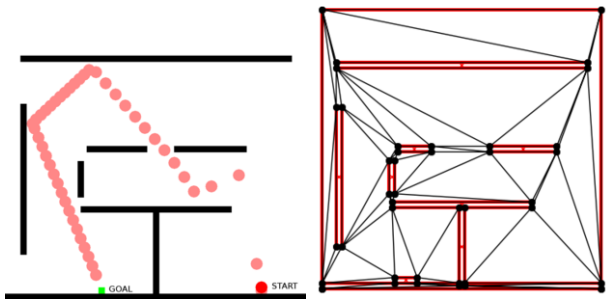
There are two key research streams in reasoning about physical systems, namely qualitative physical reasoning [2] and simulation-based reasoning [1]. The main advantage of qualitative reasoning is that it can rapidly draw useful conclusions from incomplete information [3]. [5] proposed a framework for reasoning about the motion of a 2D ball by qualitative simulation. While most of the previous work focuses on representing physical systems and describing (or predicting) consequences of actions, our method is solving a considerably harder problem as it has to find applicable actions from the infinite action space. In robotics, there has been extensive research on motion planning [8] or manipulation planning [4]. [9] developed a framework aiming at combining learning and planing and employing qualitative reasoning and linear temporal logic. There has been some work [7] on teaching robots to play mini golf.

In this paper we propose a solution to this problem: by a combination of qualitative reasoning and mental simulation as well as through a repeated process of limited sampling in the actual environment, observation of the consequences and adjusting our mental simulation. Using our proposed method, we are able to solve even very complicated instances of the hole-in-one problem. An extended version of this paper is available [6].

2 MODELING AND SOLVING HOLE-IN-ONE

We choose the following idealisation of the physical environment, which is often used in physics puzzle games: 1. The environment is a restricted 2D plane. 2. Objects are 2D rigid bodies with polygonal or circular shape. 3. There is a uniform downward gravitational force. 4. Object movements and interactions obey Newtonian physics. 5. Physics parameters of objects and the environment remain constant. We call this environment PHYS2D. An instance of PHYS2D is a tuple $\langle E, \mathbf{O}, \mathbf{P}, \mathbf{T} \rangle$, where E is the restricted plane where the objects are located, $\mathbf{O}$ a finite set of static rigid objects O, each of which has a shape, a location, an angle and a type, $\mathbf{P}$ is a set of physics

[1] ANU, Australia, email:{xiaoyu.ge,jochen.renz,p.zhang}@anu.edu.au
[2] QCIS, FEIT, UTS, Australia, email:jaehee.lee@uts.edu.au



(a) (b)

Figure 1: (a) Illustration of the problem domain in this paper. The green region is the target location H and the solid red circle is the ball B. An identified solution is shown. (b) Triangulation of the scenario

parameters that hold in the environment, such as gravity, and $\mathbf{T}$ is a set of object types and their respective physics properties such as mass and friction, or whether the object can move after being hit or remains static. We assume that all objects are initially static and stable under gravity. A physical action can be applied to an object O by exerting an impulse at a certain position p of the exterior boundary of O. An *impulse* is defined as a pair (θ, μ) where $\theta \in [0, 2\pi)$ is the direction and $\mu \in [0, \mu_{\max}]$ the magnitude of the impact. $\mu_{\max}$ is the maximal magnitude allowed in the environment, both θ and μ are continuous. In this paper we assume there is only *one* start object and *one* goal region. We call this physical action selection problem the Hole-in-One problem.

Definition 1. (Hole-in-One) **Instance:** An instance of Hole-in-One (see Fig. 1a for an example) is a tuple $\langle E, \mathbf{O}, \mathbf{P}, \mathbf{T}, B, H \rangle$, where we use a scenario of PHYS2D and determine a ball $B \in \mathbf{O}$ as the start object and H as the target hole, a polygon in E with a given location. **Solution:** A *solution* is a physical action applied to an object B such that B is delivered to the hole H as a consequence of the putt. To simplify the problem, we fix p to be the centroid of B.

The Hole-in-One problem distinguishes itself from common AI planning problems in that its search space is *infinite* and in particular the action space is *continuous*: Infinitely many different actions can allow an object to take infinitely many different paths. We propose the following method to solve this problem:

- As input scenario, we take the information about the physical environment that we obtained through potentially noisy perception.
- We first partition the free space of the given scenario into finitely many triangular zones (Fig. 1b).
- We defined qualitative rules that describe the physics that govern the transition of moving objects between the triangular zones. We use these rules to generate sequences of qualitative transitions between zones that coincide with potential real paths a moving object can take to achieve the goal. We call such a sequence a *qualitative path*.
- Once all qualitative paths are determined, we rank them by their likelihood of being realized, before we try to realize them.
- We now use a physics simulator that approximates the environment based on our input scenario to search for physical actions that realize the qualitative paths in their ranked order, i.e., actions that allow objects to follow the qualitative paths.
- The solutions we obtain here are not necessarily solutions in the real environment. Therefore, whenever we obtain a solution in our simulator, we immediately apply the solution to the real environment and see if it works. If it does not lead to a real solution, we adjust the object information in our simulator before we continue with the previous step. we will not adjust the triangulation or the qualitative paths when we adjust objects in our simulator.

3 EVALUATION

We evaluated our method in a virtual environment simulated by the Box2D (www.box2d.org) physics engine. The method also uses Box2D for its internal simulation with an incomplete knowledge of the environment. We perturb the visual input to the internal simulation by rotating each object at an angle sampled from a zero mean Gaussian. We created 72 mini-golf scenarios. The scenario designs are inspired by the game levels of a popular video game of mini-golf [3]. We compare our method with a solver (S_G) which uses a goal-oriented sampling strategy. The sampling strategy of S_G is similar to the one used in [9] that adjusts actions according to the distance between the final position of the ball and the target destination. Our method outperforms S_G in all the scenarios. S_G is less efficient because there could be many local optima in a problem. By contrast, our method can detect more different types of solutions (if there are any), which helps to avoid these local optima. Qualitative reasoning and triangulation can be achieved efficiently; it takes on average 4 seconds to generate qualitative paths based on a triangulation with around 60 zones. As the noise level increases, our method can still detect and realize qualitative paths that lead to the goal. Such qualitative paths have similar bounce sequences as their counterparts derived from perfect triangulation. Further details can be found in [6].

4 CONCLUSION

We studied a realistic problem that contains some of the essential components AI needs to successfully interact with the real world: being able to predict the consequences of physical actions and to select a physical action out of an infinite number of actions that achieves a specified goal. The proposed method involves a combination of qualitative reasoning and internal simulation together with testing proposed actions in the real environment and, if necessary, adjusting our internal knowledge based on the new observations.

REFERENCES

[1] P.W. Battaglia, J.B. Hamrick, and J.B. Tenenbaum, 'Simulation as an engine of physical scene understanding', *Proceedings of the National Academy of Sciences*, **110**(45), 18327–18332, (2013).
[2] E. Davis, 'Physical reasoning', *Handbook of Knowledge Representation*, 597–620, (2008).
[3] E. Davis, G. Marcus, and A. Chen, 'Reasoning from radically incomplete information: The case of containers', in *ACS*, volume 273, p. 288, (2013).
[4] M.R. Dogar, *Physics-Based Manipulation Planning in Cluttered Human Environments*, Ph.D. dissertation, MIT, 2013.
[5] K.D. Forbus, 'A study of qualitative and geometric knowledge in reasoning about motion', Technical report, Cambridge, USA, (1981).
[6] X. Ge, J.H. Lee, J. Renz, and P. Zhang, 'Hole in one: Using qualitative reasoning for solving hard physical puzzle problems', in *Proc. 29th International Workshop on Qualitative Reasoning*, (2016).
[7] S.M. Khansari-Zadeh, K. Kronander, and A. Billard, 'Learning to play minigolf: A dynamical system-based approach', *Advanced Robotics*, **26**(17), 1967–1993, (2012).
[8] S. Kumar and S. Chakravorty, 'Adaptive sampling for generalized probabilistic roadmaps', *J. Control Theory Appl*, **10**(1), 1–10, (2012).
[9] D. Wolter and A. Kirsch, 'Leveraging qualitative reasoning to learning manipulation tasks', *Robotics*, **4**(3), 253–283, (2015).

[3] http://www.eivaagames.com/games/mini-golf-game-3d/

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

doi:10.3233/978-1-61499-672-9-1764

Relational Grounded Language Learning

Leonor Becerra-Bonache[1] and **Hendrik Blockeel**[2] and
María Galván[1] and **François Jacquenet**[1]

Abstract. In the past, research on learning language models mainly used syntactic information during the learning process but in recent years, researchers began to also use semantic information. This paper presents such an approach where the input of our learning algorithm is a dataset of pairs made up of sentences and the contexts in which they are produced. The system we present is based on inductive logic programming techniques that aim to learn a mapping between n-grams and a semantic representation of their associated meaning. Experiments have shown that we can learn such a mapping that made it possible later to generate relevant descriptions of images or learn the meaning of words without any linguistic resource.

1 INTRODUCTION

Learning language models has been a research challenge for many years now. Grammatical Inference [7] has focused on that topic for more than 50 years. One of the main objectives of that research domain is to discover a grammar (or an associated automaton) that is a model of a set of positive (and sometimes negative) examples of sequences of words over an alphabet. Nevertheless, the learning process mainly aims to learn a syntactic structure without the help of any additional semantic information.

Since the early 2000s, methods for grounded language learning and semantic parser construction have been proposed [3, 4, 10, 11, 12]. These learn from pairs (S,M), where S is a sentence and M a meaning, a function that maps (parts of) S onto (parts of) M. In this setting, the training set must explicitly contain in M the meaning of (each part of) S; the learning cannot construct meanings by combining elements of M.

To overcome this problem some work has tried to directly learn from pairs (S,C) where S is a sentence and C is a representation of the context in which S is produced. [1] has been a first attempt to implement such an approach. Then, more recently, we presented in [2] a more efficient approach where a context is represented in first order logic and Inductive Logic Programming techniques are used to learn a mapping between n-grams (sequences of words) and their associated meanings. The main improvement with respect to previous approaches in grounded language learning is that the meanings are not provided in the training set, our learning algorithm is able to discover it by itself.

The system we present in this paper aims to improve [2] that was a proof of concept and we now want to study how such a model can deal with more realistic contexts, in noisy environments, and to observe various linguistic knowledge that can be discovered by such an approach. To make our system more realistic we decided to provide contexts in the form of images. That makes the construction of the dataset easier as the users do not have to manually provide the contexts which would otherwise be a laborious task. Thus our system learns from pairs (S,I) where S is a sentence that talks about a part of an image I.

In that way our system can be related to some deep learning approaches to the image captioning task as presented for example in [5, 6, 8, 9, 15] where their training sets are similar to ours as they are made up of pairs (sentence,image). Nevertheless, these approaches aims to learn a function that can map basic image features to ordered sequences of words. At the moment these approaches do not learn any meaning representation from the training set and it is not possible to use the learned model to do any kind of inference or reasoning. In our approach we use information about the objects of the image and build a first order logic representation of the meaning of n-grams. Thus we are able later to do some inference on that representation and it would be even possible to add a reasoning component.

We may also notice the work from Mateos Ortiz et al. [13] that uses some Machine Translation techniques to generate image description. In fact their model differs from ours in the sense that they need to build a parallel corpus (sentence,image) where each sentence has to be pre-processed by a Part-Of-Speech tagger. The main concern with such an approach is that linguistic resources are needed and we want to design an approach that does not need any linguistic resource, we want the learner to discover by itself the resources it needs.

2 OUR SYSTEM

As explained in section 1, the input of our system is a dataset made up of pairs (S,I) where S is a sentence related to a particular image I. As we use, in our experiments, the Abstract Scene Dataset provided by Zitnick et al. [16], we do not have to face computer vision problems such as object detection, semantic labelling, etc. which are known to be very hard problems. This dataset provides for each image, the set of all of its objects with some associated features. Thus, after a basic pre-preocessing step, each image I is transformed into a scene Sc. For representing scenes, we use a first-order logic based representation. Thus scenes are made up of a set of ground atoms. These atoms describe properties of, and relationships between, the objects of I. Thus, the input of the learner is a dataset made up of pairs (S,Sc) where S is a sentence related to a particular scene Sc.

The core algorithm of our system mainly aims to compute the meanings associated with all the n-grams of all the sentences of the whole dataset. We consider the meaning of an n-gram is whatever is

[1] Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France, email: {leonor.becerra,maria.galvan,francois.jacquenet}@univ-st-etienne.fr
[2] Department of Computer Science, KU Leuven, Belgium, email: hendrik.blockeel@cs.kuleuven.be

in common among all the situations in which it can be used. Thus, the meaning of an n-gram is a *context* that may be present or absent in a given scene. In our model, the learner constructs some generalized context represented by a set of atoms that may contain variables.

To compute the most specific generalization of a set of contexts $\{C_1, \ldots, C_n\}$ associated with a particular n-gram, we use the well known Plotkin's "least general generalization" (lgg) operator [14] usually used on first order clauses. Applied to contexts, the lgg has a simple mathematical description: given two contexts (set of atoms), it returns the *least general* context that *subsumes* both contexts. A context C_1 subsumes a context C_2 if and only if there exists a variable substitution that turns C_1 into a subset of C_2; C_1 is then also said to be more general than C_2. In fact, the lgg of two contexts returns all the properties that they have in common.

Nevertheless we can face a problem when, in a pair (S,Sc), there is an object that is referred in S but whose corresponding atom is not present in Sc. For example if the word "red" means red, then using it in a sentence associated with a context without red objects will cause the system to conclude that the color red is *not* common to all the contexts where "red" occurs, and therefore "red" cannot mean the color red. This leads to overgeneralization of meanings. To overcome this important problem **the first improvement** with respect to [2] has been the use of some heuristics. Instead of stating that the meaning of an n-gram G has to be common to *all* the scenes where this n-gram is used, without any exception, the algorithm repeatedly generalizes the meaning of G by computing its lgg with another randomly chosen scene until at least a certain percentage of the scenes are subsumed by that meaning.

A **second improvement** with respect to [2] is the way our system can learn the meaning of words without any linguistic resource. Indeed, in [2], a word was chosen to refer to a constant if and only if that constant was the only one remaining in the word's meaning, which was a basic strategy. Now, when the meaning of a word (a 1-gram) is updated, our program looks at the constants that occur in this meaning. Among these, it finds the constant whose occurrence "correlates" best with that of the word. If the "correlation" is high enough, this constant is then chosen to be the constant the word refers to.

Finally, **the third improvement** with respect to [2] concerns the way our system can generate all the relevant sentences associated with a given scene. To avoid generating trivial sentences that are true for most or all scenes and therefore not interesting, we defined a scoring function for sentences that takes into account the specificity of the n-grams that are chained together to form those sentences.

We made a series of experiments based on the *Abstract Scenes Dataset*, proposed by Zitnick et al. [16] that contains images of children playing outdoors and sentences that describe these images. The objective was to study the ability of our model to: (i) generate relevant sentences for a given scene and (ii) learn the meaning of words. In these experiments, we used a model learned from a corpus of 10.000 examples (in English). We first asked the system to generate all the sentences that were relevant for a given set of scenes. The sentences were obtained by chaining 5-grams. Choosing appropriate parameters to tune the core algorithm, we observed that, on average, 80% of the sentences generated were syntactically and semantically correct. Furthermore, as mention in the previous section, the sentences generated by our system try to be as specific as possible, that is, they do not state things that are true for almost all the scenes (and consequently, not interesting, such as "the sky is blue" or "the grass is green").

Concerning the meaning of words, by fine-tuning the correlation parameters of the system, we observed some improvements in terms of precision from 40% to 85% and in terms of recall from 8% to 25% with respect to [2].

3 CONCLUSION

In this paper we presented a grounded language learning system based on ILP techniques that can learn from datasets made up of pairs (S,C) where C is a set of ground atoms that represent the (partial) state of the world that is (partially) described by the sentence S. Our algorithm learns mappings between n-grams and most specific generalizations of the contexts common to the given n-grams in the dataset. We have shown that our system is able to learn the meaning of words without any linguistic resource. Having learned a language model from such a dataset, our system is able to use this model to generate relevant sentences that can describe some new scenes.

REFERENCES

[1] D. Angluin and L. Becerra-Bonache, 'Effects of meaning-preserving corrections on language learning', in *Proc. of the 15th Int. Conf. on Computational Natural Language Learning*, pp. 97–105, (2011).

[2] L. Becerra-Bonache, H. Blockeel, M. Galván, and F. Jacquenet, 'A first-order-logic based model for grounded language learning', in *Proc. of the 14th Int. Symposium on Advances in Intelligent Data Analysis*, pp. 49–60. LNCS 9385, Springer, (2015).

[3] D. L. Chen, J. Kim, and R. J. Mooney, 'Training a multilingual sportscaster: Using perceptual context to learn language', *Journal of Artificial Intelligence Research*, **37**, 397–435, (2010).

[4] D. L. Chen and R. J. Mooney, 'Learning to sportscast: a test of grounded language acquisition', in *Proc. of the 25th Int. Conf. on Machine Learning*, pp. 128–135. ACM, (2008).

[5] X. Chen and C. L. Zitnick, 'Mind's eye: A recurrent visual representation for image caption generation', in *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition*, pp. 2422–2431. IEEE, (2015).

[6] K. Cho, A. C. Courville, and Y. Bengio, 'Describing multimedia content using attention-based encoder-decoder networks', *IEEE Transactions on Multimedia*, **17**(11), 1875–1886, (2015).

[7] C. de la Higuera, *Grammatical Inference, Learning Automata and Grammars*, Cambridge University Press, 2010.

[8] H. Fang, S. Gupta, F. N. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig, 'From captions to visual concepts and back', in *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition*, pp. 1473–1482, (2015).

[9] A. Karpathy, A. Joulin, and F-F. Li, 'Deep fragment embeddings for bidirectional image sentence mapping', in *Proc. of the Int. Conf. on Neural Information Processing Systems*, pp. 1889–1897, (2014).

[10] R. J. Kate and R. J. Mooney, 'Learning language semantics from ambiguous supervision', in *Proc. of the 22nd AAAI Conf. on Artificial Intelligence*, pp. 895–900. AAAI Press, (2007).

[11] J. Kim and R. J. Mooney, 'Generative alignment and semantic parsing for learning from ambiguous supervision', in *Proc. of the 23rd Int. Conf. on Computational Linguistics*, pp. 543–551, (2010).

[12] T. Kwiatkowski, S. Goldwater, L. S. Zettlemoyer, and M. Steedman, 'A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings', in *Proc. of the 13th Conf. of the European Chapter of the ACL*, pp. 234–244, (2012).

[13] L. G. Mateos Ortiz, C. Wolff, and M. Lapata, 'Learning to interpret and describe abstract scenes', in *Proc. of the Conf. of the North American Chapter of the ACL*, pp. 1505–1515, (2015).

[14] G. D. Plotkin, *Machine Intelligence 5*, chapter A note on inductive generalization, 153–163, Edinburgh University Press, 1970.

[15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, 'Show and tell: A neural image caption generator', in *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition*, pp. 3156–3164. IEEE, (2015).

[16] C. L. Zitnick, R. Vedantam, and D. Parikh, 'Adopting abstract images for semantic scene understanding', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(4), 627–638, (2016).

This page intentionally left blank

PAIS Papers

This page intentionally left blank

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

doi:10.3233/978-1-61499-672-9-1769

Learning the Repair Urgency for a Decision Support System for Tunnel Maintenance

Y. Gatsoulis[1] and **M. O. Mehmood**[1] and **V. G. Dimitrova**[1] and **D. R. Magee**[1] and **B. Sage-Vallier**[2] and **P. Thiaudiere**[2] and **J. Valdes**[2] and **A. G. Cohn**[1]

Abstract. The transport network in many countries relies on extended portions which run underground in tunnels. As tunnels age, repairs are required to prevent dangerous collapses. However repairs are expensive and will affect the operational efficiency of the tunnel. We present a decision support system (DSS) based on supervised machine learning methods that learns to predict the risk factor and the resulting repair urgency in the tunnel maintenance planning of a European national rail operator. The data on which the prototype has been built consists of 47 tunnels of varying lengths. For each tunnel, periodic survey inspection data is available for multiple years, as well as other data such as the method of construction of the tunnel. Expert annotations are also available for each 10m tunnel segment for each survey as to the degree of repair urgency which are used for both training and model evaluation. We show that good predictive power can be obtained and discuss the relative merits of a number of learning methods.

1 INTRODUCTION

Complex decision making in domains with high impact, such as infrastructure management, is a challenging task that requires the consideration of a large number of parameters and their dependencies. For example, in the case of tunnel management, accurate pathology diagnosis and early risk assessment are critical for making cost effective maintenance plans. The common practice is that such decisions are made by a small number of domain experts, who follow their intuitions and apply tacit knowledge gained over many years of experience. This results in unsustainable subjective decision models, where it is common for experts to have no clear consensus on how the factors influence the outcomes and knowledge can be lost when experts leave.

We tackle these issues in the context of tunnel management within the EU project NETTUN[3]. Initial work [13] was focussed on following ontology engineering methodologies, where we engaged domain experts with extensive experience in tunnel management in a knowledge elicitation process to identify the concepts they consider and the rules they apply when diagnosing the pathologies of a tunnel based on its characteristics and inspection data.

Diagnosis of the pathologies is not sufficient, the experts are also required to assess the urgency of repairs that sections of tunnels require: the available maintenance budget must be prioritised, in particular taking account considerations relating to public safety. Compared to diagnosing pathologies the decision processes for prioritisation are more complex and difficult for the experts to articulate, and the ontology based methods for pathology diagnosis are not so appropriate since deciding priorities are not so much conceptual distinctions but a process of ranking. Furthermore, ontological models can have some limitations. Firstly, they may not be able to capture the true complexity of the decision process. Secondly, the process of validating these models is important, but laborious and slow. It is hard to identify missing or inaccurate rules, and some rules are "more reliable" than others, but experts typically cannot articulate this information. Furthermore, there are aspects of the decision process, such as risk assessment and potential for further degradation of tunnel portions, which take into account a number of parameters which experts find hard to specify declaratively.

To address these challenges, we adopt supervised machine learning models, taking advantage of the existence of provenance data with past observations and expert decisions. To the best of our knowledge this is the first time that machine learning methods are used in this domain for this task. A similar domain where such issues have been investigated is that of diagnosing the condition of water pipes, where most recent work focusses on Bayesian approaches [15, 8]. In our study we preferred to investigate other state of the art machine learning models (Section 3) that require little input from the experts since this is problematic as noted above.The methods we employ can also learn from the data with minimal pre-processing. Another closely related case study employed a Gaussian process model to classify surrounding rocks in tunnels, as this knowledge is important for their design and construction [16]. Although Gaussian processes are also able to provide probabilistic estimates, their performance tends to degrade in high dimensional problems when the number of features is a few dozens or more, such as the case here.

There appears to be very little work on DSS for tunnel maintenance. We have already mentioned [13] above which is concerned with pathology diagnosis, which is also the topic of [11] which also focusses on the uses of sensors to obtain a score per segment (somewhat similarly to the *cotation* score described below). There are a number of DSS to support other aspects of tunnel management, in particular, construction (e.g. [9]). There are also a variety of investigations into DSS for other kinds of transport infrastructure, e.g. highway maintenance [10], bridges [5], pavement maintenance [4], overpasses [17].

The rest of the paper is organised as follows. Section 2 describes the tunnel data and the pre-processing steps undertaken to make them suitable for the machine learning methods used in this project, which

[1] School of Computing, University of Leeds, UK, emails: {y.gatsoulis, o.m.mehmood, v.g.dimitrova, d.r.magee, a.g.cohn}@leeds.ac.uk
[2] Société Nationale des Chemins de Fers Français (SNCF), France, emails: {bastien.sage-vallier, patrick.thiaudiere, joaquin.valdes}@reseau.sncf.fr
[3] EU FP7 "New Technologies for Tunnelling and Underground Works", http://www.nettun.org

are described in Section 3. Section 4 evaluates the chosen methods, while Section 5 discusses further work and concludes the paper.

2 DATA

We have data for 47 tunnels from a national rail company. For each tunnel, survey data was collected during periodical inspections (typically every four years), and in the dataset there are multiple inspections for most of the tunnels (between one and four for each tunnel) resulting in a total of 137 inspection surveys in the dataset. For the purposes of recording surveys, each tunnel is decomposed into 10m segments. This results in a total of 8283 segments or data points. The tunnels vary in length and the average number of segments is $\mu_l = 62$ with standard deviation of $\sigma_l = 70$. There are also characteristics describing the properties of the tunnel, which can be regarded as static (i.e. set at the time of tunnel construction, and not varying over time subsequently). These variables along with the urgency repair scale are explained in more detail in the following section.

2.1 Description

The static characteristics data of each tunnel segment are called *influencing factors*, of which there are 32. Examples of influencing factors are the climate of the area, the ground type the tunnel was built on, the lining type (see Figure 1a), etc. The influencing factors are nominal data and the possible values for each of them are different. For example, the climate influencing factor can be 'favourable conditions' or 'medium conditions' or 'hard conditions'; the ground type influencing factor can be one of the following values, 'altered rock', 'compact rock', 'soil' or 'mix ground'.

The tunnels are periodically inspected for potential problems, which are called *disorders*. In this dataset 24 disorders are present. Examples of disorders are moisture (Figure 1b), displacement of a lining element (Figure 1c), etc. The disorders are binary variables, representing their presence or absence, and unlike influence factors, they can change over time.

The experts in rail company have developed a model that aggregates the observed disorders in each 10m segment into a single numeric value, called the *cotation* value (0-100) which provides a summary of the degree of disorders in that segment (see Figure 2). The higher the value the worse the condition of the tunnel is. But it does not take account of influencing factors; so a lower cotation value may be more urgent to repair if its influencing factors are particularly egregious.

Lastly, we collected the experts' recommendations about the urgency of repairs for these 47 tunnels, and these values are treated as the ground truth, and as the target variable that a model learns. This was given in a scale from 1 to 5, which denote the following recommendations: 1: "the segment is good and no repairs are needed", 2: "pay attention to this segment but no repairs need to be planned at the moment", 3: "repairs need to be planned within u_3 time", 4: "repairs need to be planned within u_2 time", and, 5: "repairs need to be planned within u_1 time"; with $u_1 < u_2 < u_3$, i.e. 5 denotes that a tunnel segment requires the most urgent repairs (see example annotation in Figure 2).

2.2 Preparation

The dataset is biased towards the first two scales denoting 'no repair': category 1 accounts for 84% of the data, while category 2 accounts for a further 5% of it. Combined, they yield a ratio of 8:1 of 'no repair' versus 'repair'. Moreover, 11 of the tunnels and 44 of the periodic inspections are classified along their full length with the category good (1), hence, offering little further information about this category that is not covered from the rest of the data. These 11 "good tunnels" are filtered out before further processing, resulting in a remaining total of 37 tunnels, 93 inspections and 6211 segments/data points with a bias ratio between the $\{1, 2\}$ versus $\{3, 4, 5\}$ categories, i.e. 'no repair' versus 'repair', of approximately 6:1 (85% of the data), which is a small improvement from before.

As the most critical decision for the tunnel owner is whether a tunnel segment requires repairs or not we have collapsed these five categories into two by merging 1-2 together representing 'no repair', and 3-5 into 'repair'. Essentially, this makes the independent variable a binary one denoting whether a tunnel segment requires repairs or not. A further stage of model building can be used to distinguish between the urgency of repair in the former case (3,4,5).

In summary, the dataset used for model building consists of 37 tunnels comprising 93 period inspections and 6211 segments. The target variable is binary: 'repair' (3,4,5) vs 'no repair' (1,2) with a 1:6 ratio. Each instance is represented by an attribute vector consisting of 24 disorders, 32 influencing factors, and a cotation value.

3 METHODS

We conducted an initial investigation to verify the non-linearity nature of the problem using a logistic regression model for different sensitivity threshold values. The results of Figure 3 confirmed this hypothesis, as it can be seen that it performs inadequately regardless of the value of sensitivity. Factor analysis, dimensionality reduction and appropriate transformation could possibly help to improve its performance, but on the other hand there are other state of the art methods that are better suited in such cases and able to learn the non-linear relations of such complex data.

As a result, we decided to investigate the effectiveness of three popular state of the art models of machine learning: decision trees, random forests and support vector machines. The reasons for choosing these and a brief explanation of the models is given in this section.

3.1 Decision Trees

One of the desired requirements was for the tunnel diagnosis experts to be able to understand the reasons for the classification produced by the machine learning method; the ultimate decision as to the urgency of repair remains with a human, and the DSS is aimed to support their decision and recommendation. For this reason one of the models we investigated was a decision tree, as it offers excellent and fast explanations on the underlying reasoning while still performing sufficiently in most cases: the expert is able to inspect the decision tree and see why a particular categorisation has been made.

Decision trees are non-parametric machine learning methods that partition the state space using decision rules, so that training instances x_i of the same category y_i are grouped together. They are most commonly represented as a decision tree structure. The key question during training is which dependent variable to use for the split. In general, this is achieved by computing the impurity (H) of the dataset before and after the split for a given variable. So at each node m denote its data as Q. For each candidate split $\theta = (j, t_m)$ consisting of a feature j and threshold t_m, partition the data into $Q_{left}(\theta)$ and $Q_{right}(\theta)$ subsets:

$$Q_{left}(\theta) = \{(x, y)|x_j \leq t_m\} \quad (1)$$

$$Q_{right}(\theta) = \{(x, y)|x_j > t_m\} = Q \setminus Q_{left}(\theta) \quad (2)$$

(a) Example of shotcrete lining type influence factor

(b) Example of moisture disorder

(c) Example of displacement of lining element disorder

Figure 1: Pictorial examples of particular values of some influencing factors and disorders of the tunnels

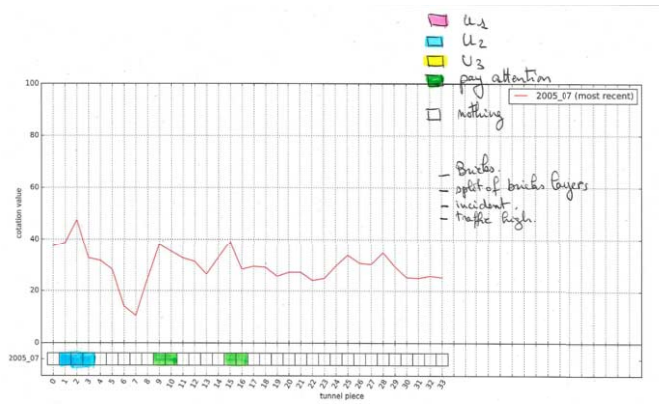


Figure 2: A graph of cotation values for an example tunnel. Also shown are the expert annotations for the ground truth for the urgency of repair: here "nothing" corresponds to repair urgency 1; "pay attention" to 2; U3 to 3; U2 to 4; and U1 to 5.

The impurity at m is computed using an impurity function $H()$:

$$G(Q,\theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta)) \quad (3)$$

and in this paper we used the Gini impurity:

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk}) \quad (4)$$

The final step is to select the variable for this node and the parameters that minimize the impurity, i.e. in this case the variable with the highest Gini gain:

$$\theta^* = \text{argmin}_\theta \, G(Q,\theta) \quad (5)$$

3.2 Random Forests

Another machine learning model that we investigated was random forests [1, 2], because they typically offer good classification performance at the trade-off of being harder to explain the underlying reasoning processes.

Random forests fall in the category of ensemble methods, which build estimators based on multiple weak classifiers, in this case short length decision trees, and specifically in this study these were CART models. Each tree is build on a random sample of the training dataset, and this process is known as bootstrap aggregating or bagging for short. Furthermore, each tree selects randomly a subset of the features which is used to train the decision tree on the random training subsample. The advantage of data and feature bagging is that the resulted meta-classifier has reduced variance and overfitting.

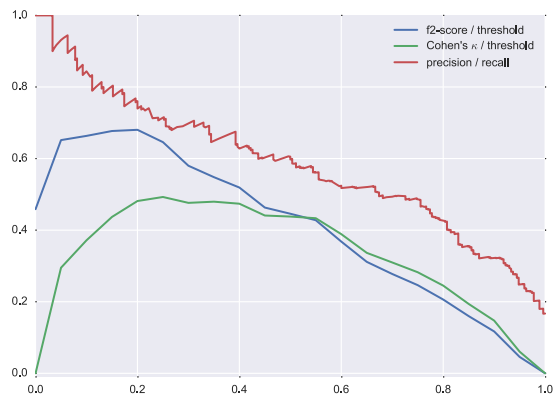


Figure 3: Logit model fitted on the dataset (best visualized in colour)

3.3 Support Vector Machines

Lastly we chose to investigate support vector machines since they are typically effective in high dimensional spaces, and from our past experience we have achieved better than state of the art results in similar problems [12].

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. The separating lines can be linear functions as well as non-linear ones. In this investigation a radial-basis function is used as the kernel for the separation.

4 RESULTS & DISCUSSION

Using the formatted data described in Section 2 we performed a cross-validation study with the methods presented in Section 3. The details of the experiments and the performance results are presented and discussed in this section.

4.1 Performance metrics

The performance metrics report are precision (p), recall (r) and f2-score ($f2$) for class 'repair', as well as Cohen's kappa coefficient (κ)

which is a more robust measurement of accuracy, particularly when there is class imbalance. When averages are reported these are the micro-averages, unless stated otherwise. We chose $f2$ rather than $f1$ since we wish to emphasise recall over precision – a false negative is potentially much more serious than a false positive.

4.2 Cross-validation

In order to test the performance of the models we conducted a leave-one-tunnel-out cross-validation (the whole tunnel, with all of its surveys). This type of cross-validation is more suitable in this case, as the typical procedure of performing k-fold cross-validation, i.e. by randomly populating the folds from the data seems to overestimate the performance of the classifiers, as shown by the results in Table 1 and Table 3 (presented in Section 4). Since adjacent tunnel segments are not truly independent instances (there are likely to be similar disorders in adjacent segments), by having the possibility one segment in the training data and its neighbour in the test data does not represent truly random sampling method. This is analogous to research in activity recognition from video data in which a preferred methodology is to leave-one-person-out [12].

Table 1: Micro performance metric averages showing that typical cross-validation (results shown for 5-folds) overestimates the true performance of the classifiers in comparison to the results shown in Table 3 where a leave-one-tunnel-out cross-validation methodology was used.

	precision (p)	recall (r)	f2-score ($f2$)	kappa (κ)
DT	.86	.87	.89	.87
RF	.91	.89	.89	.89
SVM	.92	.85	.86	.88

Leaving one tunnel out is more representative of the use case when a new tunnel comes in.

The tunnels vary in number of segments with the smallest having 3 segments (30m) and the longest 377 segments (3.7Km), with an average number of segments of $\mu_l = 62$, and standard deviation $\sigma_l = 70$. Very small tunnels, like the ones with 3 segments, can potentially bias the results in a positive or negative way. For this reason we decided to define a minimum number of segments that a tunnel must have in order to be considered as a cross-validation test case. The question then is whether choosing a minimum length of the tunnels to be cross-validated can also have any biased effect on the performance metrics of the classifiers. As such, we conducted an analysis by cross-validating for various lengths of the tunnels for a range of 10 minimum segments up to 100 minimum segments with a step of 5. Figure 4 shows the results a random forest classifier using 500 CART estimators.

It can be seen from the plot that choosing any particular minimum length does not influence significantly the results in some biased manner. Similar graphs were obtained for the rest of the models. As such, it is safe to choose a minimum length of 10 segments to have the most exhaustive cross-validation analysis. Only one tunnel had a length of less than 10 segments, resulting in a 36-fold cross validation (the 37th tunnel was still included in the training data in each fold).

4.3 Class imbalance

As described earlier the dataset even after the initial filtering still remains biased towards the 'no repair' class with a ratio of about

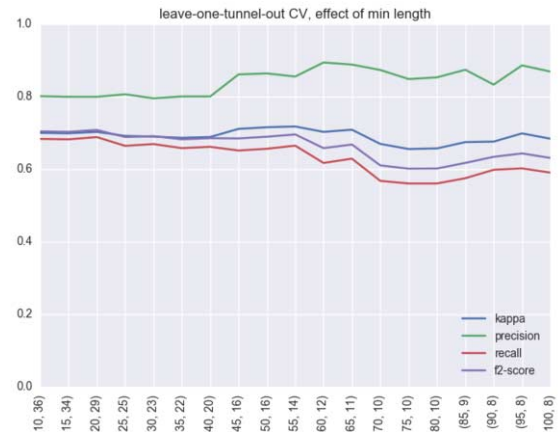


Figure 4: Investigation of the effect of number of folds in leave-one-tunnel-out cross-validation using a random forest classifier, the x-axis has i, j values where i is the minimum number of segments for a tunnel to be considered in the fold, and j is the number of tunnels with more segments than the minimum length (best visualized in colour)

6:1. Learning from imbalanced data is a common issue and a number of solutions have been proposed [7], mainly based on under-sampling the most popular class(es) or over-sampling the smaller one(s). We initially performed a k-fold cross-validation with a number of under/over-sampling methods and found that the most balanced results in terms of precision and recall were achieved with Tomek links under-sampling [14] and synthetic minority over-sampling technique (SMOTE [3]), which are both well-established and widely used methods that deal with the issue of class imbalance. We used both of these in the performance analysis of the classifiers in full leave-one-tunnel-out cross-validation, as well as tested the models with no sampling leaving the original training data unaltered.

4.4 Baseline

Table 2 presents the results from a weighted random guess according to the classes ratio in the training set on a leave-one-tunnel-out cross-validation.

Table 2: Class-portion-weighted random guess baselines

	precision (p)	recall (r)	f2-score ($f2$)	kappa (κ)
No sampling	.14	.14	.14	.00
Tomek links	.13	.13	.13	.00
SMOTE	.87	.53	.33	.00

4.5 Classifiers results

As described in Section 3 the classifier models we investigate are a CART tree, a random forest (RF) and a support vector machine (SVM). Table 3 shows the performance of the three models using no sampling, Tomek links under-sampling and SMOTE over-sampling (Section 4.3) in a leave-one-tunnel-out cross-validation (Section 4.2).

As expected, decision trees tend to perform overall worse than the other two methods in all sampling cases. An interesting result is that with Tomek links under-sampling they have a higher recall value ($r = 0.73$) than the others, i.e. they are able to recognise more of the 'repair' class than the other two methods, but their precision is much

Table 3: Performance metrics of the investigated models in a leave-one-tunnel-out cross-validation.

	precision (p)	recall (r)	f2-score ($f2$)	kappa (κ)
no sampling				
DT	.61	.65	.64	.57
RF	.79	.68	.70	.70
SVM	.82	.69	.71	.71
Tomek links under-sampling				
DT	.58	.73	.70	.59
RF	.80	.68	.70	.70
SVM	.81	.69	.71	.71
SMOTE over-sampling				
DT	.60	.68	.66	.57
RF	.76	.70	.71	.69
SVM	.59	.82	.76	.62

lower ($p = 0.58$), which means that they might become "annoying" to the experts if many segments that require no repairs are highlighted as needing so. It seems that under-sampling of the data allows the decision tree to better generalize by over-fitting even less on the 'no repair' class, since many of its training instances, which might be carrying "noisy" information has been removed.

A similar outcome appears with the SVM when SMOTE over-sampling is used. Its recall value is the highest among all models and all sampling methods ($r = 0.82$), given that SMOTE over-sampling possibly has amplified the 'repair' class, resulting in more and stronger support vectors. However, partly due to the trade-off between precision and recall, its precision value is one of the lowest ($p = 0.59$).

The best balance between precision and recall is given by RF and SVM under Tomek links under-sampling with similar performance metrics. RF also performed similarly with SMOTE over-sampling given its tolerance to over-fitting, however for a marginally worse recall, but about 4-5% better precision, the RF and SVM models with Tomek links seem better suited.

Lastly, all models perform significantly better than the baseline. However, the best results using leave-one-tunnel-out cross-validation were in the range of 0.7-0.8, which are worse than when using standard cross-validation, which were around 0.9. Given the number of variables and the number of tunnels, it is likely that more data and further investigation and discussion with the experts is needed.

4.6 Finding the most important factors

In the results presented in the previous section the full sets of disorders and influencing factors were used, which together with the cotation value account for 57 independent variables in total. We investigated whether all factors are equally important and if the models, despite their tolerance and the measurements we had taken, have still over-fitted. Figure 5 shows the cross-tabulations of the disorders from their contingency tables[4].

The graph shows that some disorders are more discriminative than others for separating the two classes. For example, it seems that disorders 7, 8, 17, 19, etc. occur more frequently when a tunnel needs repairs, while for example 1, 3, 6, etc. offer very little variation between the two classes. Due to the large number of possible values that the influencing factors can take, a similar cross-tabulations plot for them is not visually informative.

Ideally, it would be beneficial to perform some form of dimensionality reduction by factor analysis, such as principal component analysis or linear discriminant analysis. Factor analysis methods work

[4] Due to intellectual property reasons we do not display the names of all of the disorders; instead, they are numerically denoted.

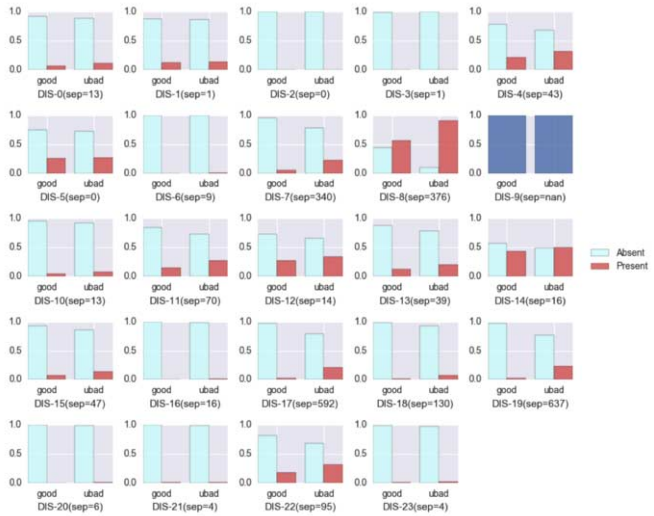


Figure 5: Cross tabulations of the disorders

on the assumption of continuous variables and thus not apply due to presence of categorical variables in our data. Instead, the models of the decision tree and of the random forest are able to provide an estimate on the importance of the features. For decision trees the importance of a feature is the Gini importance which is computed as the (normalized) total reduction of the criterion brought by that feature [2]; while for the random forest model the importance of the features is given by averaging the feature importance of each tree. The importance of each feature is a numerical value between 0-1, with higher values signifying more importance. It can be thought as the amount of variability that a feature explains in the dataset.

Table 4: Five most important disorders and influencing factors according to the decision tree and random forest models

disorder	importance	influence factor	importance
rock faults	.07	climate	.04
rock deterioration	.05	lining type	.04
falling/missing of lining element	.05	discontinuities density	.03
leakage	.05	drainage system	.03
moisture	.05	water flow/load	.03

The variable with the highest importance value is that of cotation, with an average importance between the decision tree and the random forest of 0.66. This is expected as the experts use soft thresholds that directly give a classification, in many cases regardless of the other factors when this value is high enough. To better understand the importance of the disorders and the influence factors, we excluded the cotation value and tested the models with only the disorders and the influence factors.

Table 4 shows the five most important factors for each of the disorders and influencing factors. The 15 most important disorders and 8 most important influencing factors are able to explain 80% of the variability of the dataset. Further, Table 5 shows the association between influencing factors and disorders by listing the five most correlated ones according to Cramer's V [6] and with $\phi_c > 0.37$. Some of these relations are explained by the tunnel experts. For example, the presence of a waterproofing system, which protects from extrados hydraulic pressure, can influence diagonal cracks occurring in situations like landslides, lateral thrust, or differential settlement. For a tunnel with an unlined longitudinal profile, the joint rock patterns

and cracks may create polyhedron which are potentially unstable – an example of relationship between tunnel shape and rock elements. Further discussions with the experts has also revealed that our system is identifying patterns which they would over-fit to a set of tunnels, yet this would not benefit a holistic approach. For example, hydraulic overpressure may not be a common occurrence but must be factored in, if it occurs.

Table 5: Five most correlated disorders and influencing factors according to Cramer's V co-efficient

disorder	influencing factor	Cramer's V (ϕ_c)
diagonal cracks	waterproofing system	0.53
missing rock elements	tunnel shape	0.53
diagonal cracks	strain anisotropy	0.52
diagonal cracks	tunnel age	0.47
rock deterioration	tunnel shape	0.43

4.7 Classifier results when using the most important factors

Following the analysis from the previous section for the most important factors, we used the set of the 23 disorders and influence factors together with the cotation value to retest the models. The results are shown in Table 6.

Table 6: Performance metrics of the investigated models using the most discriminant disorders and influence factors in a leave-one-tunnel-out cross-validation.

	precision (p)	recall (r)	f2-score ($f2$)	kappa (κ)
no sampling				
DT	.62	.71	.69	.60
RF	.80	.69	.71	.71
SVM	.82	.68	.70	.70
Tomek links under-sampling				
DT	.61	.72	.69	.60
RF	.79	.70	.72	.71
SVM	.81	.69	.71	.71
SMOTE over-sampling				
DT	.52	.68	.65	.52
RF	.78	.70	.72	.70
SVM	.58	.81	.75	.61

It can be seen that the results are fairly similar to the ones before when using the complete set of variables. This means that 24 factors out of the total 57 are sufficient for classifying the data equally well, i.e. a dimensionality reduction of nearly 50%.

5 CONCLUSIONS

In this paper we have presented the results of a decision support system based on state of the art machine learning methods for the domain of tunnel maintenance by a European national rail operator. This is a critical application domain, as many businesses rely on reliable and safe transport infrastructure, while the high cost of the repairs (and disruption to journeys during their implementation) dictate careful financial and operation planning.

A specialised cross-validation procedure was employed to avoid misleading overestimated results compared to standard methods. The performance metrics have demonstrated a good level of effectiveness of the algorithms. To deal with the class imbalance, since as expected health portions are many more than unhealthy ones, we utilized popular and well-tested methods of under- and over-sampling. However, the results showed that there were no significant differences, which implied that the class bias in this case is not detrimental to the effectiveness of the algorithms. Also, it was shown that similar results can be obtained with half the features from the original set, which are able to explain the majority variability of the dataset and has the potential to reduce the possibility of overfitting to the training set.

Current future work is focusing on integration of the system to the end-users' site, as well as further discussions with the experts and extensive testing with further data. We also plan to hierarchically refine the 1-5 classification (in particular to split the "repair" case (3-5) into those which are most urgent and those which are not).

Acknowledgements

This work is part of the NeTTUN project (http://nettun.org), funded by the EC 7th Framework under Grant Agreement 280712.

REFERENCES

[1] L. Breiman, 'Random forests', *Machine Learning*, **45**(1), 5–32, (2001).
[2] L. Breiman and A. Cutler, *Random Forests*, Online (http://www.stat.berkeley.edu/~breiman/RandomForests), 2004.
[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, 'SMOTE: Synthetic Minority Over-sampling Technique', *Journal of Artificial Intelligence Research*, **16**, 321–357, (2002).
[4] Jui-Sheng Chou, 'Web-based CBR system applied to early cost budgeting for pavement maintenance project', *Expert Systems with Applications*, **36**(2, Part 2), 2947–2960, (March 2009).
[5] Michael P. Enright and Dan M. Frangopol, 'Maintenance planning for deteriorating concrete bridges', *Journal of Structural Engineering*, **125**(12), 1407–1414, (1999).
[6] Cramr Harald, *Mathematical Methods of Statistics*, p282, Princeton University Press, 1946.
[7] H. He and E.A. Garcia, 'Learning from Imbalanced Data', *IEEE Transactions on Knowledge and Data Engineering*, **21**(9), 1263–1284, (2009).
[8] Z. Li, B. Zhang, Y. Wang, F. Chen, R. Taib, V. Whiffin, and Yi Wang, 'Water pipe condition assessment: a hierarchical beta process approach for sparse incident data', *Machine Learning*, **95**(1), 11–26, (jun 2014).
[9] V. Likhitruangslip and P. G. Ioannou, 'RISK-SENSITIVE DECISION SUPPORT SYSTEM FOR TUNNEL CONSTRUCTION', in *ASCE Geotechnical Special Publication*, (2004).
[10] Bhoj Raj Pantha, Ryuichi Yatabe, and Netra Prakash Bhandary, 'Gis-based highway maintenance prioritization model: an integrated approach for highway maintenance in nepal mountains', *Journal of Transport Geography*, **18**(3), 426 – 433, (2010). Tourism and climate change.
[11] Noureddine Rhayma, Aurlie Talon, Pierre Breul, and Patrick Goirand, 'Mechanical investigation of tunnels: risk analysis and notation system', *Structure and Infrastructure Engineering*, **12**(3), 381–393, (2016).
[12] J. Tayyub, A. Tavanai, Y. Gatsoulis, A. G. Cohn, and D. C. Hogg, 'Qualitative and Quantitative Spatio-temporal Relations in Daily Living Activity Recognition', in *Proc. of 12th Asian Conference on Computer Vision*, pp. 115–130, Singapore, (2014).
[13] D. Thakker, V. Dimitrova, A.G. Cohn, and J. Valdes, 'PADTUN - Using Semantic Technologies in Tunnel Diagnosis and Maintenance Domain', in *ESCW2015*, (2015).
[14] I. Tomek, 'Two Modifications of CNN', *IEEE Transactions on Systems Man and Communications*, **6**, 769–772, (1976).
[15] C. Wang, Z. Niu, H. Jia, and H. Zhang, 'An assessment model of water pipe condition using Bayesian inference', *Journal of Zhejiang University SCIENCE A*, **11**(7), 495–504, (July 2010).
[16] Yan Zhang, Guoshao Su, and Liubin Yan, 'Classification of surrounding rocks in tunnel based on Gaussian process machine learning', in *2011 International Conference on Electric Technology and Civil Engineering (ICETCE)*, pp. 3971–3974. IEEE, (April 2011).
[17] Jana elih, Anej Kne, Aleksander Srdi, and Marjan ura, 'Multiplecriteria decision support system in highway infrastructure management', *Transport*, **23**(4), 299–305, (2008).

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

doi:10.3233/978-1-61499-672-9-1775

ONE - A Personalized Wellness System

Ajay Chander and **Ramya Srinivasan** [1]

Abstract. As the world becomes increasingly digitally readable through a variety of sensors, digital services will play a key role in advising and supporting people towards a variety of goals. In this paper, we present a personalized wellness system that leverages techniques from cognitive science and machine learning to improve a user's well-being by suggesting daily micro-goals (e.g., "*bring a healthy snack to work*"), and by enabling social sharing of individual achievements. Specifically, we propose a method for estimating a user's likelihood of successfully completing a given micro-goal ("ONE") and study the correlation between ONEs and users' actions to improve their chances of reaching their wellness objectives.

1 INTRODUCTION

Health-tech, the use of technology to provide personalized healthcare and wellness, is in the midst of a furious renaissance [3]. Given that lifestyle diseases lead to 75% of long-term healthcare costs, a particular area of focus for healthtech is disease prevention and support of positive wellness behaviors [3]. Many sensors to track and visualize our wellness behaviors have and are being built [6]. However, these systems assume that awareness of activity patterns and data will lead to behavior change and goal achievement, an assumption that is not necessarily true. Our humanity brings with it biases, and as social animals, our behaviors tend to be more influenced by social information [4], [2]. While many health companies (e.g., Healthways) have taken advantage of the popularity of social media sites such as Facebook and Twitter to create communities that promote participation in exercise and diet programs, they are not designed to offer benefits at an individual level [1].

In this work, we propose a personalized wellness system that improves the chance of a user successfully achieving a wellness objective, wherein objectives could be focus areas such as nutrition, body, mind, etc. (Fig. 1a). The app, called ONE, is motivated by the cognitive science concepts of *novelty* and *social proof.* ONEs are purposefully chosen simple micro-goals. A ONE comprises of an unique image and a textual message (Fig. 1b). A new ONE every 24 hours leverages novelty. Additionally, the number of people who have completed a given ONE is also available, serving as a social proof to influence action (Fig. 1c). A user can like (heart) or dislike a ONE and also post a picture as a proof of having completed a ONE. A preliminary version of the app is available at https://goo.gl/hLSFsi

We build machine learning models that can provide an estimate of a user's likelihood of hearting/completing a new ONE. Specifically, we learn features that influence a user's action— to understand what is appealing/compelling for a user in terms of hearting/completing a ONE. We leverage this information to suggest those ONEs that are more likely to be completed by a user and hence propel them towards the achievement of their wellness objectives.

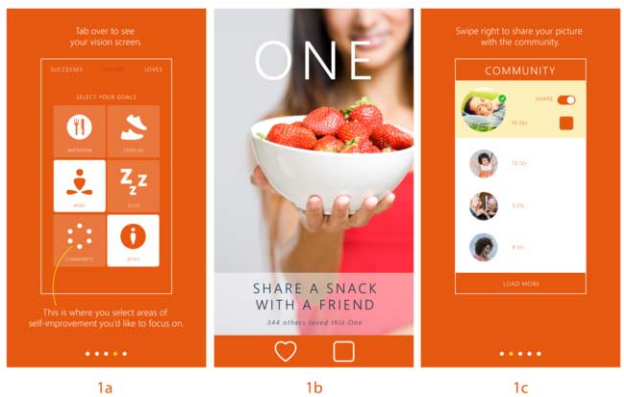


Figure 1. Illustration of the ONE app. Left: Wellness objectives which the users can choose to work on, Center: A sample ONE, Right: Community page showing others who have completed a ONE

2 METHODOLOGY

The first step in the design of a model that can determine a user's likelihood of completing a new ONE (micro-goal) is in selecting features that are good predictors of the same. Towards this, we investigate multi-modal features, namely, the textual features and the image features associated with a ONE message, and user-specific features such as their wellness objectives and their likes history.

2.1 Feature Extraction and Validation

By choosing those features that exhibit high correlations with likes, we can ensure good prediction of a user liking a new ONE. Consider, for e.g., the correlation between wellness objectives and likes. Let $P(u, v)$ represent the probability with which a user u likes ONEs belonging to a wellness objective v. Thus, $P(u, v) = \frac{likes(u,v)}{likes(u,V)}$, where V is the set of all wellness objectives. Averaging $P(u, v)$ across all users will provide the probability distribution for the event of an average user liking from a wellness objective v.

In order to understand the choice of distribution that best fits the data under consideration, a Cullen and Frey graph is plotted. Results indicated that the observation can be modeled by a Beta distribution. In order to rule out a uniform fit to the data (which would be the case if the users liked randomly across all wellness objectives), Kolmogorov-Smirnov (K-S) goodness of fit is performed with the null hypothesis being that a normal distribution fits the data. The result of the test indicates that the null hypothesis is to be rejected.

[1] The authors are with Fujitsu Laboratories of America, USA and have equally contributed to the paper.

Thus, an average user does not like randomly, instead there are some favorite wellness objectives, and this information can be leveraged to estimate their likelihood of liking a new ONE. In a similar manner, other predictors are established, the details of which is described next.

2.2 Feature Descriptors

Table 1 provides a summary of the feature descriptors used to predict user's likelihood of liking a new ONE. Each user is represented by a 7-D feature vector that captures textual (rows 1-3 of Table 1), image (rows 4-5) and user-specific information (rows 6-7).

Table 1. Summary of feature descriptors.

Feature	Description
Positive words count	Words associated with a positive sentiment in the ONE text
Local heart count	Number of people who liked a ONE as is seen by the user under consideration
ONE ID	Unique number associated with the theme of the ONE text
Face ID	Binary value indicating presence/absence of a face in the ONE image
Emotion ID	Unique number associated with the emotion of the ONE image
Likes count	Total number of likes by the user normalized by the number of likes across all users
Objective ID	Unique number associated with the wellness objectives preferences of the user

Positive words in the ONE message are counted based on their co-occurrence with the positive valence words of the AFINN-111 [5], a dataset consisting of 2477 words rated for valence between -5 and 5. Based on the relevance of the contents of a ONE text with certain themes (e.g., body, community, growth, mind, nature related), it is given a unique ONE ID (numbers 1-5). In a similar manner, based on the emotion associated with the ONE image—anger, calmness, disgust, joy, affection and a neutral emotion, an emotion ID is attached with the image (1-6). Each user is allowed to choose a set of wellness objectives when they install the ONE app. The unique combination of objectives a user chooses, is identified by the objective ID feature. The description of the rest of the features is clear from Table 1.

2.3 ONE Achievement Prediction

A user's progress towards achievement of wellness objectives can be predicted in terms of (1) user liking a new ONE, and (2) user completing a ONE/posting a picture of it. We explain the model with respect to the user liking a new ONE, but the method is generalizable to picture posting prediction, except that likes count feature (Table 1) would now be the number of pictures posted in an objective category.

We build a Logistic regression (LR) model for each wellness objective. For training, each user u is represented by the 7-D feature vector (Sec 2.2) and the label is the probability that the user likes a ONE from that wellness objective $P(u, v)$ (Sec 2.1). L2 regularization is used in training the model. We use roughly $\frac{2}{3}$ of the data for training and the rest for testing. The testing instances are ranked in the order of the obtained probabilities of liking. This, in turn, can be used to suggest ONEs that are more likely to be liked by the user. The model is validated by computing the normalized discounted cumulative gain ($NDCG$) using the predicted ranks (by LR model) and the actual ranks (Sec. 2.1) for a user.

3 Experiments

User data is being continuously collected anonymously and stored using Amazon Web Services as the mobile app's backend. We maintain multiple data stores, including:
1. User goal-behavior: All aspects of user actions (e.g., hearting, post pictures, un-hearting) towards completing the ONE are recorded into this database. Each entry of the database has the timestamp of the corresponding action, the user's ID, the associated ONE, and the local heart count.
2. User app-interactions: This database records all interactions of the users with the app (e.g., entered the vision page, entered community page, etc.). Each entry contains user's ID, time stamp of interaction, corresponding ONE, and actual interaction.

We report initial results on a dataset comprising of over 50 users spanning 3 months. We first analyzed the correlation between various features and user's likes using Chi-squared correlation coefficient. Specifically, we set the null hypothesis that there does not exist a correlation between the chosen feature and users' likes, and computed Chi-square correlation coefficient at 95% significance level. Results indicated statistically significant correlation between emotion ID and likes with most users liking ONEs that conveyed a feeling of "joy" and "affection". Many users were interested in ONEs related to community activities followed by those ONEs concerning mind. While there was a high correlation between ObjectiveID, Tag ID and Likes count with users' likes, the Face ID had the least correlation with likes. In future, we would like to have a feature weighting strategy and also explore sequence of users' interactions for better prediction.

The ONE app has been designed through a human-centered design process, which included testing various versions of the app and its interface and interactions with a variety of users in our lab. In order to evaluate the empirical prediction performance of the LR model, we computed the $NDCG_5$ defined by $\frac{DCG_5}{IDCG_5}$ wherein DCG_5 is estimated from the LR model using the predicted probabilities of users liking ONEs from a wellness objective, and $IDCG_5$ is computed from the actual number of likes by user across various objectives. Results gave a decent value of 0.8388. A Naive Bayes classifier was also tested, but LR model was the better one.

4 Conclusions

We presented ONE, a personalized wellness system, that aids users in achieving their goals by suggesting new micro-goals every day and by providing social information for behavioral support. By learning the probability with which a user likes or completes a ONE, it is possible to suggest those ONEs that are more likely to be completed by the user, thus enabling goal achievement. Initial experiments demonstrated promising performance of the system.

REFERENCES

[1] D. Centola, 'Social media and the science of health behavior', *Circulation*, 2135–2144, (2013).
[2] N. Christakis and J. Fowler, 'Social contagion theory: Examining dynamic social networks and human behavior', *Statistics in Medicine*, (2011).
[3] Rock Health, 'Digital health consumer adoption', (2015).
[4] R. Nickerson, 'Confirmation bias: A ubiquitous phenomenon in many guises', *Review of General Psychology*, 175–220, (1998).
[5] F. Nielsen, 'A new anew: Evaluation of a word list for sentiment analysis in microblogs.', *ESQ Workshop on Making Sense of Microposts*, (2011).
[6] Z. Yumak and P. Pearl, 'Survey of sensor-based personal management systems', *BioNanoScience*, 254–269, (2013).

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

doi:10.3233/978-1-61499-672-9-1777

Planning Search and Rescue Missions for UAV Teams

Chris A. B. Baker and **Sarvapali Ramchurn** and **W. T. Luke Teacy**[1] and **Nicholas R. Jennings**[2]

Abstract. The coordination of multiple Unmanned Aerial Vehicles (UAVs) to carry out aerial surveys is a major challenge for emergency responders. In particular, UAVs have to fly over kilometre-scale areas while trying to discover casualties as quickly as possible. To aid in this process, it is desirable to exploit the increasing availability of data about a disaster from sources such as crowd reports, satellite remote sensing, or manned reconnaissance. In particular, such information can be a valuable resource to drive the planning of UAV flight paths over a space in order to discover people who are in danger. However challenges of computational tractability remain when planning over the very large action spaces that result. To overcome these, we introduce the survivor discovery problem and present as our solution, the first example of a continuous factored coordinated Monte Carlo tree search algorithm. Our evaluation against state of the art benchmarks show that our algorithm, Co-CMCTS, is able to localise more casualties faster than standard approaches by 7% or more on simulations with real-world data.

1 Introduction

The increased prevalence of low-cost, robust, commercially available Unmanned Aerial Vehicles (UAVs) has led to concerted efforts to utilise these platforms in order to aid first responders with collecting sensory data without putting human lives at risk [1]. In particular, work has focused on developing autonomous systems to minimise the involvement of overstretched first responder personnel, and to ensure UAVs can take action quickly [22, 24]. Key to this work, is the idea of enabling coordinated UAVs to explore a disaster space to discover the spatial location of casualties: a difficult task given the spatial extent to explore and the continuous action-space represented by a UAV's axes of motion.[3]

To enable this exploration, advances in data collection have created new sources of information about disaster scenarios that contribute to increased awareness of the situation on the ground during a disaster. In particular, data gathered by crowd-sourcing is becoming more readily available because of the speed with which it can be generated, and its ability to directly reflect the experiences of the people on-scene; people who can often give a very accurate report on the hazards in their vicinity and the number of potential casualties [21]. However, despite these advancements, at present there is little work that seeks to use information on *danger* and the spatial locations of *people* to inform the paths of UAVs through a disaster space, in order to maximise the number of observations made of possible casualties.

Currently, the state of the art for UAV path planning algorithms focus on areas such as target tracking for surveillance [5, 17] which, although related to the exploration of a disaster space, are designed to find a known number of targets that are in motion, rather than an unknown number of survivors distributed over an area: a very different problem since algorithms must be able to make predictive estimations of people that *might* be present in as-yet unvisited locations. Other developments in path planning focus on trying to reach a set goal location [9, 19], or working with single autonomous vehicles [8, 18]; neither of which fulfil the need for algorithms that coordinate multiple vehicles in an *explorative* traversal of the disaster space, rather than aiming for a particular final location. Moreover, specific challenges exist in coordinating multiple vehicles. For example, there is often no benefit to multiple UAVs providing sensor data of the same location: there must be coordination between the vehicles to allow them to find survivors in a disaster, without all attending the same locations. Additionally, UAVs must be able to coordinate over actions temporally: visiting the same location at the same time might be straightforward to avoid, but planning a UAVs current action to account for the future action of other nearby UAVs is a non-trivial problem, particularly as the number of UAVs in an environment increases.

Against this background, some work has recently been performed on planning problems that involve the coordination of multiple vehicles, including in disaster scenarios [2, 4]. However, as yet both these approaches require some degree of simplification before planning can commence by discretising the environment into a number of *cells* to be examined. Since locations in the real-world are not discretised in this way, this requires some additional processing of incoming data before UAVs can begin their exploration. Furthermore, by simplifying data in this way it is inevitable that some information is lost when a UAV can only be considered as visiting single cells, rather than being able to plan according to a *continuous* range of motion.

In this paper we seek to address these shortcomings in planning UAV searches of disaster areas, specifically with a view to future applications and field-tests as part of the MOSAIC collaboration[4] and ongoing work with Rescue Global.[5] In particular we make the following contributions to the state of the art:

1. We introduce a novel formulation of the *survivor discovery problem*; specifically modelled on a likely real-world scenario with the goal of locating an unknown number of people, over a wide area, by detection of mobile phone signals and where the diminishing survivability of the people with time is incorporated into the reward function.
2. We develop a novel decentralised algorithm that allows multiple UAVs to coordinate to explore a large continuous disaster space

[1] Agents Interactions and Complexity Group, University of Southampton, UK, email {cabb1g08, sdr, wtlt}@ecs.soton.ac.uk
[2] Department of Computing, Imperial College London, UK, email n.jennings@imperial.ac.uk

[3] Although this work is framed in terms of disaster response, the same coordination algorithms could be applied to other UAV applications: such as geophysical surveying, security operations, or ecological monitoring.
[4] An EPSRC funded collaboration between the University of Southampton and the University of Sheffield: `http://www.mosaicproject.info`
[5] Information available at: `http://www.rescueglobal.org`

under spatial and temporal constraints. Our approach utilises a *belief map*—a term referred to the mapping of spatial locations onto some function that represents numerical data—to represent the presence of people and danger in the environment, and to form the basis of the rewards calculated in the planning algorithm.

3. Furthermore, in order to demonstrate the applicability of our scenario to potential future disasters, we test and evaluate our approach on real-world data (gathered from the 2010 Haiti earthquake) simulating a very large action space, showing consistent gains in survivor discovery of at least 7% compared to benchmarks, with higher gains of around 20% for scenarios with additional UAVs.

In the following sections we first outline the current state of the art areas of research in autonomous path-planning and exploration. Next, we introduce the specific formulation of the environment model and UAV behaviour considered in our simulations, before introducing the continuous form of our coordinated Monte Carlo tree-search algorithm. Following this, we present empirical evidence to substantiate the benefits of this approach, before concluding and discussing future work and applications we will explore.

2 Background

In order to best use UAVs to aid responders in disasters they must be able to plan paths autonomously, as a group, in a decentralised manner. This is particularly important given the implications of a UAV failing in the field: any centralised system that relies on a single UAV (or other central coordinator) will fail entirely if that central point fails; whereas a decentralised system can—in principle—continue to function as UAVs are removed. Furthermore, as we have already indicated, it is beneficial to use prior information about the area to inform the flight paths of UAVs in order to maximise the likelihood of discovering survivors. Currently, work on path planning in robotics focusses primarily on reaching goal locations and frequently formulates path planning as a control problem [14]. Conversely, in a disaster scenario there need not be any final end-point to a UAV's path planning; rather the length of the exploration may be constrained by—for example—battery life, and the number of people to be discovered must be maximised over the length of the path. Alternatively, much work has also been done to enable the use of vision algorithms and belief data to track mobile targets or map an area [20]. However, this area of research often focusses on locating a known number of targets, or covering a bounded space for the purposes of mapping. These foci are not relevant in a scenario where an unknown number of people are distributed over an already-mapped (but very large) area, as is frequently the case with displaced populations during and after disasters.

Against this background, we find closer similarities with work on solving Markov Decision Processes (MDPs); specifically where locality of UAVs can be used to reduce calculation overheads. In particular, the generality of MDP formulations suits our construction of a simulation environment where we are provided with numerical data used for a belief map; and MDPs in general have a number of well-explored algorithmic solutions available. Specifically, work by [2] utilises factored tree-searches for partially observable MDP solutions; exploiting problem structure to allow factorisation in a way reflective of local state spaces and interactions. In a similar way, we use factored trees in this paper to represent the available actions of UAVs in a disaster environment, factoring the value of locating people between UAVs within spatial proximity of each other. However, our work deals with a continuous state-space (in this case representing the continuous range of actions available to a UAV at any given time); as well as a detailed model of the scenario specifically geared towards UAV applications. This requires significant changes to the existing approach in [2] and [4], primarily in applying sampling of the continuous action-space while still allowing UAVs to coordinate with one another. Crucially, discrete approaches to MDP solutions require some form of recognition that a particular set of actions has been entirely explored: clearly where a continuum of actions exists this cannot be said to be true, since the number of individual actions available for selection is infinite. As such we have had to adapt different approaches usually applied to planners dealing with continuous spaces [11, 25] into the regime of coordinated exploration.

3 Scenario Model

The overarching aim of disaster response work is to minimise the loss to human life in the disaster area. Currently, we perceive there to be a lack of suitable environment models that fully characterise this problem and express it in terms of sensing technology and UAV behaviour that already exists. To this end, we introduce a novel formulation for the exploration of a disaster situation where we require that UAVs focus on areas of perceived danger, but also where these regions intersect with likely occupation by people. The rationale here, is that data about a region containing known hazards (for example, high levels of radiation or presence of fire) is only useful in preserving human life when it is known or believed that there are likely to be persons in the vicinity of a hazard; or will be at some future time. We give the example of radiation as a possible manifestation of danger in a disaster scenario. In principle, we can extend this to any phenomenon that is present over an extended area and represents some risk to human life. This general approach could then represent several types of hazard in a disaster area, such as flooding, chemical spills, or risk from earthquake-damaged buildings. At this point, we consider the location of such hazards as static. Such an assumption can be justified in scenarios with slow-changing conditions (relative to the time taken by the UAVs to explore).

We will now outline the formulations describing the state of the environment, and the actions available to the UAVs in our simulations.

3.1 Environment Model

In considering a model for the distribution in space of a number of civilian casualties, we exploit the ubiquity of mobile phone ownership and assume the use of mobile phone signals as proxies for the presence of a person. As well as having precedent in previous use in disaster scenarios [15, 26], this has the specific advantage of allowing identification of individual sources using unique identifiers associated with each handset. While a priori knowledge of the number of victims in a given area might be unavailable, first responders can maintain a belief distribution over unobserved victims while also attempting to isolate signals that have been observed, in order to reduce the uncertainty in the location of victims.

As a result, we explicitly envisage a scenario where UAVs are equipped with some form of detector capable of providing a (noisy) estimate of the range of individual unique phone signals. Specifically, we seek to localise the expected position of victims in order to reduce the time taken by search and rescue teams to find (and subsequently rescue) them. In more detail, we associate the uncertainty of a person's location with time taken to search the area for that person. By moving the detectors around the space, the expected location

value can be determined with higher precision; effectively reducing the area to be covered (and thus the time taken) by rescue services from a large initial area to a much smaller location.

We consider a search area containing a number of signals $s \in \mathcal{S}$ indicating the presence of people in some *danger* (mapped spatially by a two dimensional scalar function $\mathcal{D} : \mathbb{R}^2 \mapsto [0, 1]$), corresponding to their expected likelihood of dying within the next time-step t. The reward we gain R is related to the number of people we hope to observe, their likelihood of survival, and a discovery time t indicating how long it would take to rescue any victim:

$$R = \sum_{p_i, d_i \forall s_i \in \mathcal{S}} p_i \cdot (1 - d_i)^t$$

where $d_i \in \mathcal{D}$ and $p_i \in \mathcal{P}$ represents the expected number of people for a given signal $s_i \in \mathcal{S}$. Each signal s_i is mutually distinguishable from other signals, and the magnitude of each can be sensed by UAVs within a set radius. In the first instance we assume a relatively flat prior belief of victim position, implying a long time to locate an individual. However, with a set of observations (O) of—for example—the strength of a mobile phone signal some estimate can be made of the location of a person; effectively reducing the time to locate them. We denote this using a *time to find* parameter t_f that decreases linearly with the estimated area of the location of an individual phone signal.

As such, at any given time our reward is then:

$$\sum_{p_i, d_i \forall s_i \in \mathcal{S}} p_i \cdot (1 - d_i)^{t_f(O)}$$

whereas projecting reward to any arbitrary time in the future we have for a series of observations at time t:

$$R_{known} = \sum_{p_i, d_i \forall s_i \in \mathcal{S}} p_i \cdot (1 - d_i)^{t_f(O_t)+t}$$

By collecting information on signal sources we can use a population Monte-Carlo (PMC) [7] to model the likely locations of a person, which increases in precision as more measurements are taken, explicitly reducing t_f. Thus reward is fundamentally a function related to the danger at a given location believed to contain a victim, and the precision with which the location of that person is known. Planning can be performed by simulating the result of measurements on the probability distribution for each person and extrapolating the effect on t_f in each instance.

To account for signals we have yet to observe, we include a term for the *expected* number of victims outside of the range of observations. In principle the search area for such victims would be the entirety of the area over which observations have yet to be recorded, since the positions of the victims are known with no localisation whatsoever. Thus for a continuous distribution of expected people (p_e):

$$R_{unknown} = \int p_e(x, y) \cdot (1 - d(x, y))^{t_f} \, dxdy$$

The global reward function at time t then simply becomes the sum of the two components:

$$R_{total} = R_{known} + R_{unknown} \quad (1)$$

It is worth noting that since the formulation of reward from the population Monte-Carlo simulation is essentially separate from the tree-search function outlined below, the two are (in general cases) separately applicable.

3.2 UAV Behaviour Formulation

We consider simple UAV flight dynamics—including minimal constraints on performance—since the focus of our work is on planning rather than constraint optimisation, and because restrictions on UAV behaviour can be included in subsequent iterations of the model as constraints on the reward function. Thus the set of UAVs $U = \{u_1, \ldots, u_m\}$ traverse the space in iterations of a fixed distance δ per time-step t (i.e. at fixed speeds and altitudes); with a continuous domain of available angles available to determine the direction of the next action. The action vectors enabling UAV u_k to move at the next time step is defined as $a_k = (a_{k\alpha}, a_{k\beta}, a_{k\gamma}, \ldots)$ where each Greek index can be interpreted as an angle between 0^o and 360^o. In theory the cardinality of a_k is infinite, but as detailed below we use continuous space tree-search methods to restrict our search to finite subsets. Each UAV selects a sequence of actions to produce its trajectory $T_k = [a_k(t = 1), a_k(2), \ldots, a_k(t_{end})]$ (for a trajectory that ends at time t_{end}); which together form the set of all trajectories $\mathbf{T} = \{T_1, \ldots, T_m\}$. Thus the collective goal of the UAVs is to plan a set of trajectories to satisfy: $\mathbf{T}^* = \arg\max_R R(\mathbf{T})$.

4 The Coordinated Continuous Monte Carlo Tree-Search Algorithm

In choosing Monte-Carlo tree search as the basis for our solution, we note its ability to sample very quickly from large state spaces (traditionally used in solving games), and the flexibility with which it can be applied to general problems [6]. To do this we exploit locality between UAVs to factor the search space into local joint-action trees. Furthermore, we allow trees to coordinate over shared factors (that is, UAVs in multiple trees) using the max-sum algorithm [23] by exchanging messages to express the local reward gained by UAVs taking particular actions at future times. In other words, when selecting which node in a tree to expand the individual trees are coordinated over their shared UAVs to select mutually nonconflicting actions that are maximally beneficial to both trees.

At this stage, we design the algorithm to plan on-line and recalculate the optimum action at each time-step. In this way we have built-in robustness to temporal changes in the map (as well as not requiring a priori knowledge of future coordination requirements). We currently run simulations in a centralised fashion—insofar as they are performed on a single computer—but with allowances for multiple parallel threads representing the different individual calculations for each portion of the factored utility. In addition, we note that the nature of the max-sum coordination is such that UAVs are not required to have perfect information: it is sufficient that they know their local utility and are able to share this with local neighbours.

Specifically, we introduce an additional step to the standard MCTS process of tree growth. This growth is typically summarised: node *selection*, *expansion*, *rollout* or simulation, and *backpropagation* [6, 16]. Most significantly, we modify the selection process to determine which node to expand by coordinating in parallel between trees via max-sum. We detail our approach in the following subsections.

4.1 Tree Construction

At each timestep in the simulation, the coordinated MCTS (Co-MCTS) algorithm begins by calculating which UAVs require coordination with their neighbours, leading to the form of the UAV-based factor graph constructed in the joint-action creation function $\mathcal{J}$ (Line 3), detailed fully in Algorithm 1. This is performed to establish whether coordination is needed in a given UAV's locality: where

a UAV is spatially isolated from neighbouring UAVs, a local tree is grown. The resulting groups of UAVs will form the basis of the factor graph used in the max-sum calculation (Line 13 in Algorithm 1). The result of $\mathcal{J}$ is represented formally by a set $N = \{n_1, \ldots, n_f\}$ that represents the domain of the factor nodes to be coordinated. Specifically, each member of N contains a set of actions corresponding to a group of UAVs that require coordination. In this case, the corresponding element in N—say n_i—would be the set of actions available to these UAVs: $n_i = \{a_1, a_2, \ldots, a_k\}$. Trees are grown for each n_i in N, each of which in turn represents the factors in the max-sum graph connected to the variables representing the available actions of the UAVs. In more detail, trees are grown for each joint-utility between interacting UAVs, and coordination between trees is performed when trees share access to a given UAV. Individual nodes in the tree n_i will be indicated as $n_i^{(k)}$, or from any arbitrary tree by $n^{(k)}$.

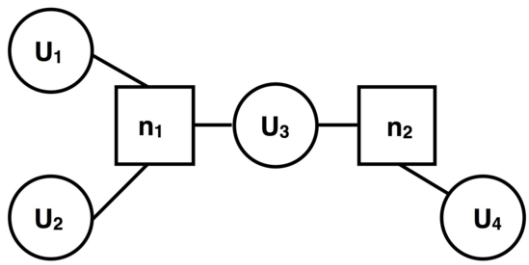


Figure 1. An example of four UAVs (u_{1-4}) interacting via a max-sum factor graph. Trees are grown for n_1 and n_2.

An example max-sum factor graph is represented in Figure 1. Here, four UAVs may interact at the next time-step in the following sub-sets: $\{u_1, u_2, u_3\}$ and $\{u_3, u_4\}$. As such, the algorithm maintains two joint trees for these sets, represented as the utility nodes n_1 and n_2, which must coordinate over the action of the UAV common to both: u_3. Framing this in terms of the action selected as a result of the tree-search, the coordination serves to ensure the two trees select an action for a_3 that is both mutually beneficial to both factored reward functions, and also the same; since a_3 can only take a single action at the next time step, the two trees must "agree" on what this action is.

4.2 Node Selection and Expansion

Algorithm 1 begins with the creation of the root nodes representative of each factor seen in Line 6, which are recorded in the set N_r (Line 4). Following this, the creation and growth of branches is performed Δ times inside the loop beginning at Line 8. This begins by exploring down each tree, starting from the root node, to determine which node to branch on next. Node selection is performed in accordance with standard progressive widening [11, 25], where we sample randomly from the action set of a node up to a limit of K actions per node, with K defined as in [10] to be a parameter of constants $C > 0$ and $\alpha \in (0, 1)$ and the time of simulation t: $K = Ct^{\alpha}$.

Line 9 introduces the current set of nodes (across all trees) to be expanded next, N_{next}, and Line 10 creates the set of previously expanded nodes N_{prev}. At Line 13 the max-sum algorithm is used to maximise the value of rewards (as per Equation 1) the actions over each n_i, returning a vector of favourable actions $a^* = (a_1^*, a_2^*, \ldots, a_m^* \mid a_k^* \in a_k)$. Since each n_i depends on a subset of actions, the function select$(n^{(k)}, a^*)$ serves to return only the actions corresponding to a given $n^{(k)}$. This is then used as the argument to create the new expansion to a node in N_{next} in Line 16.

Algorithm 1 Coordinated MCTS

```
CoMCTS (U, D, t = 0)
1.  for t in [1, ..., t_end]
2.    //Creation of factor graphs given UAV locations//
3.    N ← J (U)
4.    N_r ← ∅
5.    for n_i in N
6.      append(N_r) ← n_i^(0)
7.    endfor
8.    for eachstep in [1, ..., Δ]
9.      N_next ← N_r
10.     N_prev ← ∅
11.     while N_next ≠ ∅
12.       //Max-sum coordination returns best actions for shared factors//
13.       a* ← maxsum (N, N_next)
14.       for n^(k) in N_next
15.         //Actions relevant to n^(k) selected//
16.         n_new^(k) ← expand (n^(k), select (n^(k), a*))
17.         if expansions (n^(k)) ≥ K
18.           remove (n^(k), N_next)
19.           append (n^(k), N_prev)
20.         endif
21.       endfor
22.     endwhile
23.     for n^(k) in N_prev
24.       //Simulation (rollout) and backpropagation of results//
25.       rollout(n_new^(k))
26.       backpropagate(n_new^(k))
27.     endfor
28.   endfor
29.   for n_i in N
30.     a*_i = (bestactions (n_i))
31.   endfor
32.   t ← t + 1
33. endfor
```

Although not explicitly tested, we note that this approach ensures communications between UAVs need not be excessive. We note existing literature has shown at-length that max-sum in particular is robust to low bandwidth and irregular message-passing [12, 13, 23]. In practice we envisage that where UAVs share a tree a single UAV will handle the growth and planning of the joint actions.

4.3 Rollout and Backpropagation

The rollout portion of the MCTS (line 25 in Algorithm 1) is traditionally a coarse estimate of the affect of future actions as the result of exploring a particular node in the action space, although some more recent work has focussed on principled simulations using existing MCTS techniques [3]. In this example, we base the rollout on a random-walk through the action space starting at the node just expanded, biased in the direction of the last action taken. This method has the benefit of showing not just the contribution of any random series of actions, but of taking more actions similar to the one represented by the frontier node (for each UAV). Intuitively, a purely random rollout from one node in a joint action tree will be insignificantly different from a rollout from any similar node. Conversely, our rollout policy contributes to the exploration value of a node by indicating possible future reward through continued tree expansion with a preference for repetitions of the action itself.

Finally the rewards calculated at the leaf nodes are backpropagated up the tree towards the root by iteratively updating cumulative average rewards for each upstream node (Line 26). This is unchanged from classic MCTS.

5 Empirical Evaluation

To verify the performance of our algorithm on data relevant to real-world disaster scenarios, we used data from the Ushahidi project [21] produced from crowd-sourced information during the 2010 Haiti earthquake to generate $\mathcal{D}$ and $\mathcal{S}$.[6] This dataset was selected as it represents one of the largest available sources of information about spatial distributions of people and damage to buildings from any recent natural disaster. Furthermore, with increased interest in simple systems that allow crowds to provide data using their phones very quickly after a disaster takes place, the prevalence of such datasets will invariably increase in future; further underscoring the importance of testing our algorithm on this type of data. Specifically, we extracted the level of damage and coordinates of buildings in a 2km square centred on the capital, Port-au-Prince. Damage was rated based on crowd reports on a scale from 1 to 5, with 5 being most severe.

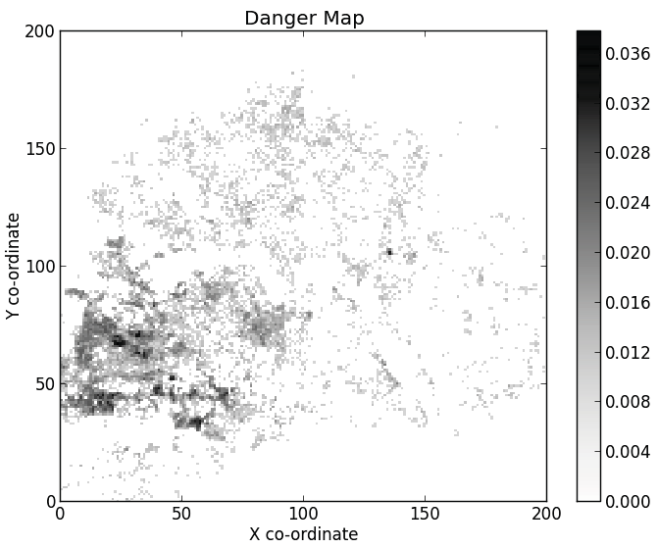


Figure 2. Danger $\mathcal{D}$ as a function of position, created from Ushahidi dataset centred over Port-au-Prince. Dimensions of $2km$ along each side.

We then constructed a decomposed grid world of size 200×200 of $10m \times 10m$ cells, to form the basis of the danger function $\mathcal{D}$. Since damaged buildings represent an estimate of the damage in an area and thus, the danger to the victims on the ground, we formed a belief map of danger to the populace by summing the total number of buildings above a threshold level (set to a crowd report of damage 3 and above) in each cell, before multiplying by a common factor to convert the data into a map representative of expected fatalities (noting the constraint imposed by the domain of $\mathcal{D}$). The environment is displayed in Figure 2 with a scale showing the value of d in each location. In calculating values for use in the reward functions, the value of danger is based on the mean expected position of the signal based on the data collected. Where the spatial location is not yet clearly established we have found that empirically the change in spatial danger was smooth enough that nearby values tended to be close to the final estimated value of d_i in most cases.

Assuming a UAV speed—typical of quad rotor vehicles—of $10ms^{-1}$ amounts to the traversal of one action of moving $\delta = 10m$ in one timestep of one second. We typically simulate UAV searches over time horizons of $t_{end} = 1000$.

The performance metric used is the percentage reduction in the time for total cumulative discovery time t_f (averaged over the number of UAVs) since it best reflects the ability of the UAV search to pinpoint victims for rescue. We benchmark against a similarly coordinated—but discrete—MCTS implementation, where the action space of the UAVs is restricted to moving between the cells forming the danger-function environment. This scenario poses similar challenges of coordination in large action spaces but benefits from existing work that deals with factored finite-space tree-search [2], and has already shown its efficacy in planning over the Ushahidi dataset [4].

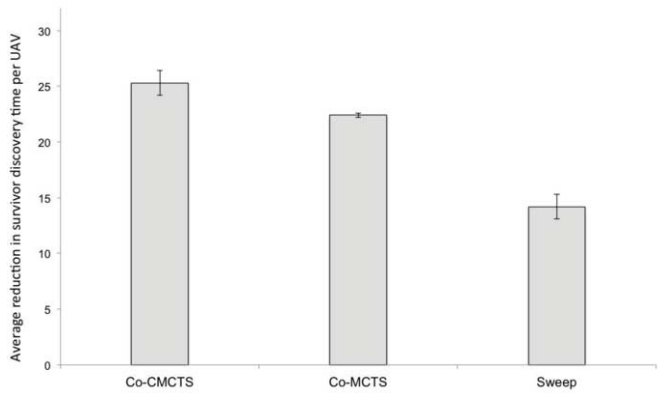


Figure 3. Result of randomised starting position tests for each of the continuous coordinated MCTS, discretised (cellular) coordinated MCTS and a simple lawnmower sweep-search; performed 10^6 times. Results indicate reduction in t_f averaged over the four UAVs in the scenario.

An initial simulation with four UAVs in randomised start locations on the map shown in Figure 2 is shown performing against a discretised coordinated MCTS implementation (as described in [4]) and a simple lawnmower-style sweep search over the area for comparison. This shows a gain of around ~7% over a discretised search space (Figure 3). We note that computation time of tree growth on the order of hundreds of nodes typically took less than half a second, demonstrating that computational complexity is not excessive.

Furthermore, we are able to demonstrate the consistency of our approach on addition of further UAVs to the simulation. Intuitively the reward gained by each UAV in a well-coordinated algorithm should suffer fewer diminishing returns when adding more to the scenario. In detail, this is because any additional UAVs should still localise close-to the same number of people as other UAVs in the environment, if they coordinate the exploration task effectively as a group. If they do not, one would expect additional UAVs would explore the same regions of the disaster space as those already present: which, as discussed previously, offers negligible improvements to the global reward function when compared to localising previously un-seen casualties. We demonstrate in Figure 4 that additional UAVs results in a slower decrease in observed reward than in the discretised action space. Most notably at 5 UAVs the difference in performance per-vehicle is approximately 18% in favour of the continuous algorithm. This benefit is due to the continuum of actions available being less restrictive than in a cellular decomposition of the search area; allowing more effective coordination.

[6] Available from `http://www.ushahidi.com`

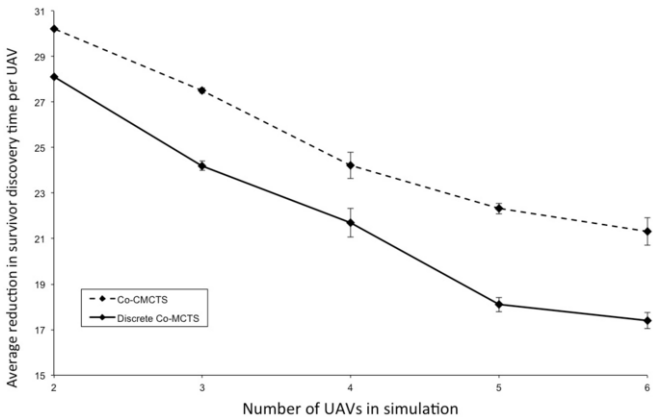


Figure 4. Comparison of continuous and discrete coordinated MCTS in a $t_{end} = 1000$ simulation of varying numbers of UAVs. The continuous space approach is not only better than the discretised approximation, it is more consistent in its reward per-UAV added to the scenario. Results here are averaged per-UAV in the simulation.

6 Conclusions

Motivated by increased availability of belief-data about disaster environments, we have introduced an implementation of a decentralised, factored, coordinated Monte Carlo tree search algorithm for the purpose of discovering survivors in a simulated UAV path planning scenario. Tests were carried out on real-world data from the 2010 Haiti earthquake via the Ushahidi platform; an environment with a continuous action space over a large area. We demonstrated the capability of our Co-CMCTS algorithm in sampling this space and planning paths, and demonstrated consistent performance gains over a discretised algorithm in the number of survivors discovered of up to 18%. Future work will seek to extend these solutions to different densities of survivors, time-varying belief maps, and—as part of ongoing collaborative efforts—will attempt field-trials of the algorithms proposed above on real-world platforms to further demonstrate the efficacy and real-world applicability of our contributions.

References

[1] S. M. Adams and C. J. Friedland, 'A Survey of Unmanned Aerial Vehicle (UAV) Usage for Imagery Collection in Disaster Research and Management', in *Proceedings of the Ninth International Workshop on Remote Sensing for Disaster Response*, volume 9, Stanford, MA, (2012).

[2] C. Amato and F. A. Oliehoek, 'Scalable Planning and Learning for Multiagent POMDPs', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 1995–2002, Austin, Texas, (2015). AAAI.

[3] H. Baier and M. H. M. Winands, 'Nested Monte-Carlo Tree Search for Online Planning in Large MDPs', in *European Conference on Artificial Intelligence (ECAI)*, volume 242, pp. 109–114. IOS, (2012).

[4] C. A. B. Baker, S. D. Ramchurn, W. L. Teacy, and N. R. Jennings, 'Planning Search and Rescue Missions for Unmanned Aerial Vehicle Teams', in *ICAPS Proceedings of the 4th Workshop on Distributed and Multi-Agent Planning (DMAP-2016)*, London, (2016). AAAI.

[5] S. Bernardini, M. Fox, and D. Long, 'Planning the Behaviour of Low-Cost Quadcopters for Surveillance Missions', in *Proc. of 24th Int. Conference on Automated Planning and Scheduling*, pp. 445–453, Portsmouth, NH, (2014). AAAI.

[6] C. B. Browne, E. J. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, 'A Survey of Monte Carlo Tree Search Methods', *Transactions on Computational Intelligence and AI in Games*, **4**(1), 1–43, (2012).

[7] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert, 'Population Monte Carlo', *Journal of Computational and Graphical Statistics*, **13**(4), 907–929, (2004).

[8] M. Cashmore, M. Fox, T. Larkworthy, D. Long, and D. Magazzeni, 'AUV Mission Control via Temporal Planning', in *2014 IEEE International Conference on Robotics and Automation*, pp. 6535–6541, Hong Kong, China, (2014). IEEE.

[9] Y.-b. Chen, G.-c. Luo, Y.-s. Mei, J.-q. Yu, and X.-l. Su, 'UAV path planning using artificial potential field method updated by optimal control theory', *International Journal of Systems Science*, (October), 1–14, (jun 2014).

[10] A. Couëtoux, J.-B. Hoock, N. Sokolovska, O. Teytaud, and N. Bonnard, 'Continuous Upper Confidence Trees', in *Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, ed., C. A. Coello-Coello, number 5, pp. 433–445, Rome, Italy, (2011). Springer.

[11] R. Coulom, 'Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search', in *5th International Conference on Computer and Games*, eds., P. Ciancarni and H. J. V. D. Herik, volume 4630, pp. 72–83, Turin, Italy, (2006).

[12] F. M. Delle Fave, A. Farinelli, A. Rogers, and N. R. Jennings, 'A Methodology for Deploying the Max-Sum Algorithm and a Case Study on Unmanned Aerial Vehicles', in *The Twenty-Fourth Innovative Applications of Artificial Intelligence Conference*, pp. 2275–2280, Toronto, (2012). AAAI.

[13] A. Farinelli, A. Rogers, and N. R. Jennings, 'Agent-based decentralised coordination for sensor networks using the max-sum algorithm', in *Autonomous Agents and Multi-Agent Systems*, volume 28, pp. 337–380, (2014).

[14] C. Goerzen, Z. Kong, and B. Mettler, 'A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance', *Journal of Intelligent and Robotic Systems*, **57**(1-4), 65–100, (nov 2009).

[15] A. Goetz, S. Zorn, R. Rose, G. Fischer, and R. Weigel, 'A Time Difference of Arrival System Architecture for GSM Mobile Phone Localization in Search and Rescue Scenarios', in *Positioning Navigation and Communication (WPNC), 2011 8th Workshop on.*, pp. 1–4. IEEE, (2011).

[16] L. Kocsis and C. Szepesvari, 'Bandit based Monte-Carlo Planning', *Machine Learning: ECML 2006*, **4212**, 282–293, (2006).

[17] A. Kolling and A. Kleiner, 'Multi-UAV Motion Planning for Guaranteed Search', in *Autonomous Agents and Multiagent Systems*, pp. 79–86, (2013).

[18] M. Kothari and I. Postlethwaite, 'A Probabilistically Robust Path Planning Algorithm for UAVs Using Rapidly-Exploring Random Trees', *Journal of Intelligent and Robotic Systems*, **71**(2), 231–253, (sep 2012).

[19] M. Kothari, I. Postlethwaite, and D.-W. Gu, 'Multi-UAV Path Planning in Obstacle Rich Environments Using Rapidly-exploring Random Trees', in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 3069–3074, Shanghai, (dec 2009). IEEE.

[20] Y. C. Liu and Q. H. Dai, 'A Survey of Computer Vision Applied in Aerial Robotic Vehicles Yu-chi', in *OPEE 2010 - 2010 International Conference on Optics, Photonics and Energy Engineering*, number 201, pp. 277–280, Wuhan, China, (2010). IEEE.

[21] N. Morrow, N. Mock, A. Papendieck, and N. Kocmich, 'Independent Evaluation of the Ushahidi Haiti Project', Technical report, Ushahidi, (2011).

[22] R. R. Murphy, 'A Decade of Rescue Robots', in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5448–5449, Vilamoura, Portugal, (2012). IEEE.

[23] A. Rogers, A. Farinelli, R. Stranders, and N. R. Jennings, 'Bounded approximate decentralised coordination via the max-sum algorithm', *Artificial Intelligence*, **175**(2), 730–759, (2011).

[24] United Nations Foundation, 'Disaster relief 2.0', Technical report, United Nations, (2011).

[25] Y. Wang, J.-Y. Audibert, and R. Munos, 'Algorithms for Infinitely Many-Armed Bandits', in *Advances in Neural Information Processing Systems*, eds., D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, 1729–1736, NIPS, 21 edn., (2008).

[26] S. Zorn, R. Rose, A. Goetz, and R. Weigel, 'A Novel Technique for Mobile Phone Localization for Search and Rescue Applications', in *2010 International Conference on Indoor Positioning and Indoor Navigation*, number September, pp. 15–17, Zurich, Switzerland, (2010). IEEE.

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

doi:10.3233/978-1-61499-672-9-1783

Integrating ARIMA and Spatiotemporal Bayesian Networks for High Resolution Malaria Prediction

A. H. M. Imrul Hasan [1] and **Peter Haddawy**[2]

Abstract. Since malaria is prevalent in less developed and more remote areas in which public health resources are often scarce, targeted intervention is essential in allocating resources for effective malaria control. To effectively support targeted intervention, predictive models must be not only accurate but they must also have high temporal and spatial resolution to help determine when and where to intervene. In this paper we take the first essential step towards a system to support targeted intervention in Thailand by developing a high resolution prediction model through the combination of Bayes nets and ARIMA. Bayes nets and ARIMA have complementary strengths, with the Bayes nets better able to represent the effect of environmental variables and ARIMA better able to capture the characteristics of the time series of malaria cases. Leveraging these complementary strengths, we develop an ensemble predictor from the two that has significantly better accuracy that either predictor alone. We build and test the models with data from Tha Song Yang district in northern Thailand, creating village-level models with weekly temporal resolution.

1 INTRODUCTION

Malaria remains a global public health problem with an estimated 214 million cases of malaria globally in 2015 and 438,000 malaria deaths [23]. In Thailand, 31,121 and 15,446 confirmed cases were reported in 2014 and 2015, respectively [18]. Since malaria is prevalent in less developed and more remote areas in which public health resources are often scarce, targeted intervention is essential in allocating resources for effective malaria control. Since 2009 Thailand has implemented an E-Malaria Information System (EMIS) [13] to systematically gather case data and data on relevant covariates in order to support control policy decisions as well as to track their effectiveness. With the rise of resistant strains of malaria as well as the greatly increased incidence of dengue (another mosquito vector borne disease) in Thailand and neighbouring Malaysia, Thailand's Center of Excellence for Biomedical and Public Health Informatics, which houses EMIS, has expressed interest in exploring use of this and related data sources to support targeted intervention by producing appropriate predictive models. To effectively support targeted intervention, predictive models must be not only accurate but they must also have high temporal and spatial resolution to help determine when and where to intervene [16]. While much work has been done on malaria prediction models, high resolution prediction remains a challenge [21].

Modeling of malaria is challenging because disease transmission can exhibit spatial and temporal heterogeneity, spatial autocorrelation, and seasonal variation. In addition, some covariates such as temperature affect incidence rates in a nonlinear fashion. Among the numerous techniques that have been used to create predictive models [30], ARIMA is the most popular because of its ability to accurately model characteristics of the time series as well as capture some dependence on covariates. Despite a variety of modeling approaches (ARIMA, regression, neural nets, SIR models) having been explored, no work has yet explored the potential of Bayes nets as a modeling framework for malaria. Bayesian networks [19] provide a number of advantages for modeling of malaria, including the ability to explicitly represent uncertainty, handle missing data, and represent nonlinear relations. In addition, the model structure, which typically reflects the problem structure, can be used to generate explanations of the predictions.

In this paper we take the first essential step towards effective targeted intervention by developing a high resolution prediction model through the combination of Bayes nets and ARIMA. Bayes nets and ARIMA have complementary strengths, with the Bayes nets better able to represent the effect of environmental variables and ARIMA better able to capture the characteristics of the time series of malaria cases. We find that for one week prediction the Bayes net model performs best for high and mid-level incidence while ARIMA performs better for low-level incidence. For two week prediction, ARIMA performs best for all incidence levels. Leveraging these complementary strengths, we develop an ensemble predictor from the two that has significantly better accuracy that either predictor alone at every incidence level, using model trees to select the features and the weights to put on each model. We build and test the models with data from Tha Song Yang district in northern Thailand, creating village-level models with weekly temporal resolution. This is the first work to use Bayesian networks to model malaria and the first to create an ensemble forecasting model using Bayes nets and ARIMA.

2 RELATED WORK

A number of researchers have explored the combination of neural networks and ARIMA. One approach has been to use the neural network to classify the residuals [6, 14, 28] from the ARIMA model. A residual is the difference between an actual value and it's prediction. The logic behind this is that the residuals will contain non-linearity since ARIMA cannot capture the non-linear structure of time series. Adhikari [1] proposed an approach to combine neural nets with a number of forecasting models (Box-Jenkins ARIMA, FANN, EANN and SVM) in two steps. First a set of sample weights is obtained from the inverse relation between absolute forecast and error forecast of the respective model. Second a neural net model is made to predict the combining weights by going through the sample weights.

[1] Mahidol University, Thailand, email: ahmimrul.has@student.mahidol.ac.th
[2] Mahidol University, Thailand, email: peter.had@mahidol.ac.th

One common combining method for ensemble techniques is to assign weights to component forecasts where each weight is inversely proportional to the prediction error of the corresponding component model, which ensures that the model with higher error gets the lower vote for the combined contribution. A nonlinear framework [2] is also proposed by researchers where correlations between pairs of component forecasts are considered along with the optimal weights determined from pairs of train and test sets. Wichard [24] proposed hybrid ensembles that combine multiple models (Nearest Trajectory, Neural Network, Difference, Trend cycle and AR model) by using a weight which is proportional to symmetric mean absolute percent error (SMAPE) computed over a left-out part of the time series.

Relevant work on using Bayes nets for disease modeling includes that of Cooper et al. [5] on modeling spatiotemporal patterns for non-contagious diseases that can cause outbreaks in a population such as may occur in bioterrorist attacks. Spatiotemporal Bayes nets have been applied to a number of environmental modeling problems. Most Bayes net environmental models to date have either focused on spatial aspects [12, 7] or temporal aspects [11], with only the recent work of Wilkinson et al. [25] addressing the combined dimensions of spatial heterogeneity, spatial influence, and temporal evolution.

In this paper, we combine Bayesian network and ARIMA forecasting models by a simple linear function of weighted component models and a few selected features. The combining weights and features are chosen by a model tree algorithm.

3 GEOGRAPHIC REGION AND DATA

We demonstrate our approach with the problem of weekly village level malaria prediction in Tha Song Yan district of Tak province of Thailand. Tha Song Yang is a hilly area with 66 villages in which malaria is endemic. It is located along the border with Myanmar and this proximity to the border results in imported cases. Policy makers were interested in having a predictive model that can assist in timely targeted intervention, particularly given the remoteness of some villages, as well as in understanding the factors that most influence the malaria incidence.

The case data for our model consists of weekly clinically confirmed malaria cases obtained from Thailand's national E-Malaria Information System (EMIS) [13]. The data covers each of the 66 villages for the years 2012 and 2013, providing a total of 6,579 records with 12,800 total cases (plasmodium falciparum, plasmodium vivax). The numbers of cases per village per week ranged from 0 to 82 with a mean of 2.1.

In addition to the case data, our model makes use of a number of environmental factors associated with malaria. Predictive models often make use of environmental factors such as rainfall, temperature, and vegetation as determinants of mosquito vector density and infectivity, as well as malaria incidence in the preceding time period (typically week or month) as an estimator of the human reservoir of the parasite and the population susceptibility [8]. Since seasons affect the environmental factors, models also often incorporate some representation of time or seasonality. The factors included in our model and the source for each are

- Normalized Difference Vegetation Index (NDVI): monthly satellite data from MOD11A3,
- Land Surface Temperature (LST): monthly satellite data at 5 km resolution from MOD11C3,
- Rainfall: daily satellite data at 10 km resolution from JAXA Global Rainfall Watch,
- Slope: Average in 1 km buffer around each village, computed from elevation data,
- Distance to nearest stream: Euclidean distance from village center to closest point on the stream,
- Stream density: total stream length in 4 km buffer around each village,
- Distance to border: Euclidean distance from village center to the closest point on the border with Myanmar,
- Month: month of the year.

NDVI, LST, Rainfall, and Month are temporal variables whose values are indexed by week, while Slope, Stream density, Distance to nearest stream, and Distance to border are non-temporal variables whose values are constant over time. The variables NDVI, Distance to nearest stream, and Stream density are thought to positively impact malaria incidence. LST has a nonlinear effect on malaria with malaria incidence low for low temperatures, increasing over some region, and then dropping off for high temperatures. Rainfall is known to have a positive effect on malaria incidence except for very heavy rainfall which can wash away the larvae. Slope is included because it interacts with rainfall with rain draining off more quickly the higher the slope. Distance to border is a proxy for the number of imported cases and is thought to have a positive effect on incidence. Some values for the variables obtained from satellite data were missing due to cloud cover during some time periods. Missing values were filled in using temporal and spatial interpolation as appropriate.

4 BAYESIAN NETWORK PREDICTION MODEL

Malaria may be modeled using one Dynamic Bayes net (DBN) per village. Figure 1 shows the structure of the DBN prediction model for two time slices: week 0 and week 1. The model includes temporal nodes such as NDVI at week zero (NDVI_w0), and non-temporal nodes for random variables whose states do not change with time, such as Border_Distance.

Time lags in the model include a one week lag in the effect of Rainfall on NDVI and a three week lag in the effect of Rainfall on Mosquito Population Density. Our malaria model includes three latent variables: Rainfall_Effect_w1, which represents the interaction of rainfall and slope; Stream_Effect, which summarizes the effect of stream distance and stream density; and Mosquito_pop_density_w1, which represents the effect of various environmental factors on the vector density. Inclusion of these variables increases the explanatory power of the network and, importantly, reduces the size of some of the conditional probability tables. For example, inclusion of Mosquito_pop_density_w1 reduces the size of the CPT for the node Incidence_w1 which would otherwise be too large to learn from the available data. Because of the inclusion of latent variables, the conditional probability tables (CPTs) for the Bayes net were learned using the expectation maximization (EM) algorithm.

The model is used for prediction by entering known values for the variables at week zero (w0), rainfall at week minus 2 (Rainfall_wm2), and Month for weeks zero and one, and computing the posterior probability of incidence at week 1 (Incidence_w1). To predict incidence for week two, an additional time slice is included with similar repeated structure. Each node of malaria incidence is divided into 14 ranges. The predicted incidence is then computed as the expected value of the incidence random variable:

$$E(incidence\_w_1) = \sum_{i=1}^{14} \{P(range_i) * mean(range_i)\} \quad (1)$$

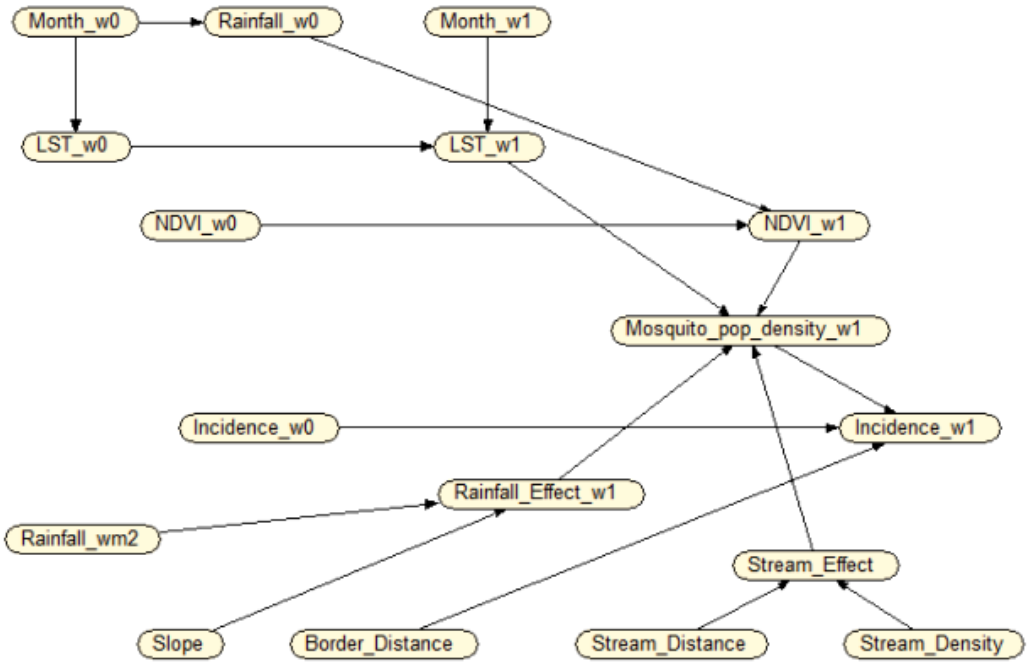


Figure 1. Bayesian network model showing two time slices.w0 = week 0, w1 = week 1, wm2 = week minus 2

where $i = 1, 2, ..., 14$ are the ranges, $P(range_i)$ is the probability of ith range and $mean(range_i)$ is the mean of the distribution of the data over the ith range. The prediction accuracy was evaluated by mean absolute error (MAE):

$$MAE = \frac{\sum_{case=1}^{N} Abs(Predicted_{case} - Actual_{case})}{N} \quad (2)$$

where N is the number of cases. The MAE of the Bayes net model for one week prediction over all 66 villages is 1.098 and the MAE for two week prediction is 1.417.

5 ARIMA PREDICTION MODEL

Auto Regressive Integrated Moving-Average (ARIMA), also known as the Box-Jenkins approach [4], is the most popular stochastic time series forecasting model of the past few decades. It is a modeling approach that can be used to calculate the probability of a future value lying between specific limits. It has three parameters: autoregression (AR), integration (I), and moving average (MA). The ARIMA$(p, 0, 0)$ or autoregressive model is represented as

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + e_t \quad (3)$$

where θ_0 is the intercept, p is the number of auto regressive terms, Y_t is predicted result, Y_{t-p} is the observation of time $t - p$, $\phi_1, \phi_2, ..., \phi_p$ is a set of parameters that are calculated by linear regression, and e_t is the regression error. The ARIMA$(0, 0, q)$ or moving average only depends on q past random terms and the current random term e_t and is expressed as

$$Y_t = \mu - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ..... - \theta_q e_{t-q} + e_t \quad (4)$$

where q is the number of the moving averages, $\theta_1, \theta_2, ..., \theta_q$ is a set of parameters, and μ is the mean of the series. The ARIMA$(p, 0, q)$ model is

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \mu - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} + e_t \quad (5)$$

The ease of use of the Box-Jenkins methodology [4] for optimal model fitting, as well as the flexibility of the representation have made ARIMA a highly popular modeling approach. A variant of ARIMA capable of modeling seasonal data called SARIMA [9] includes additional terms in the ARIMA model and is written as $(p, d, q)(P, D, Q)_m$. The upper-case notation is for seasonal parts of the model. The term m is the number of units per season, P is the number of seasonal autoregressive (SAR) terms, D is the number of seasonal differences and Q is the number of seasonal moving average (SMA) terms. ARIMA models are used frequently by researchers for disease surveillance and prediction (malaria [8, 22, 29], dengue [20], HFMD [17]). The version of ARIMA that includes external predictors is known as ARIMAX, which is denoted by ARIMA(p, q, d)X (where X is the external independent variables). The ARIMA model with extra variables often performs better [15, 3] than simple univariate ARIMA when the dependent variable is explainable by other external factors.

In this study, the ARIMA models were developed in the R Software package v3.2.3. A best fit was obtained from the combination of all 66 time series (66 villages) by applying the auto.arima() function available in the R package called "forecast" [10]. We took 70% incidence data for training from each village and concatenated them by inserting Null values in between to keep the seasonality intact. The remaining 30% of the data was used for model testing. We trained

a seasonal ARIMA((1,1,0)(1,0,0)) model for this study. For multivariate ARIMA, we chose explanatory variables from the environmental factors (Month_w0, Rainfall_w0, LST_w0, NDVI_w0, Rainfall_wm2) that we used for the Bayes net (section 4). An iterative greedy method was used to select the external variables that provided significant improvement in prediction accuracy based on MAE. The only variable so selected as external covariate was Month_w0. We computed two-period-ahead forecasts by following the rolling window method. Over all 66 villages, the MAE of ARIMA for one week prediction is 1.102 and for two week prediction is 1.217, while the MAE of ARIMAX for one week prediction is 1.074 for two week prediction is 1.251.

6 COMBINING BAYESIAN NETWORK AND ARIMA PREDICTION MODELS

We analyzed the prediction accuracy of the ARIMA and Bayes net models by testing them on three different subsets of villages divided according to average incidence rate: (1) 13 villages with high incidence{Min: 0, Max: 82, Avg.: 7.43}, (2) 13 villages with medium incidences{Min: 0, Max: 16, Avg.: 1.91} (3) 14 villages with low incidence{Min: 0, Max: 3, Avg.: 0.099} and all 66 villages containing the entire spectrum of incidence. The results are shown in table 1 (columns: BN, ARIMA, ARIMAX). It can be seen from the table that the Bayes net and the ARIMA models have complementary strengths. For one week prediction, the Bayes net model has the best performance for high- and mid-level incidence villages and the ARIMA models have the best performance for low-incidence villages. For two week prediction, the ARIMA models perform best for all classes of villages.

We combined the models using stacked generalization [27], where outputs are collected from $level_0$ models (trained on $level_0$ data) and treated as data for another learning problem at $level_1$. The model applied in this step is referred to as the $level_1$ model. For the $level_1$ generalization we used model tree induction [26] on the Bayes net, ARIMA, and ARIMAX predictions along with a number of attributes that characterize the incidence rate in the current and previous weeks as well as overall:

- *incidence_W_0:* Incidence of the current week (W_0)
- *incidence_rate:* Sum of weekly incidence of a village divided by the maximum sum among all villages
- *incidence_WM1:* Incidence of previous week
- *incidence_Avg:* The average incidence of a village
- *BN_Prediction:* Predicted incidence by Bayesian Network
- *ARIMA_Prediction:* Predicted incidence by the ARIMA model
- *ARIMAX_Prediction:* Predicted incidence by the ARIMAX model
- *Last_Two_Avg:* Average of last two weeks incidence

We used the M5P [26] model tree algorithm from the WEKA (v3.6.10) data mining tool. The test set of $level_0$ was further divided into 70% and 30% then used as the train and test sets of $level_1$. For the combination of the Bayes net and ARIMA, the algorithm generated a tree with a single leaf node for both one- and two-week prediction and selected only incidence rate and current week incidence in addition to the BN and ARIMA predictions:

$$\begin{aligned} incidence\_W_1 = {} & (0.3424 \times incidence\_W_0) + \\ & (0.253 \times ARIMA\_Prediction\_W_1) + \\ & (3.8015 \times incidence\_rate) + \\ & (0.2451 \times BN\_Prediction\_W1) + 0.0128 \end{aligned} \quad (6)$$

$$\begin{aligned} incidence\_W_2 = {} & (0.3224 \times incidence\_W_0) + \\ & (0.2409 \times ARIMA\_Prediction\_W_2) + \\ & (6.6964 \times incidence\_rate) + \\ & (0.0824 \times BN\_Prediction\_W2) + 0.0448 \end{aligned} \quad (7)$$

For the combination of the Bayes net and ARIMAX the algorithm also generated a tree with only one node for one- and two-week prediction. The prediction models include the variables incidence rate, current incidence, and incidence average. For two-week prediction the combining function substitutes the value of last two week average for the BN prediction value.

$$\begin{aligned} incidence\_W_1 = {} & (0.3582 \times incidence\_W_0) + \\ & (0.2254 \times ARIMAX\_Prediction\_W_1) - \\ & (20.226 \times incidence\_rate) + \\ & (0.9385 \times incidence\_avg) + \\ & (0.2112 \times BN\_Prediction\_W1) - 0.0571 \end{aligned} \quad (8)$$

$$\begin{aligned} incidence\_W_2 = {} & (0.6355 \times incidence\_W_0) + \\ & (0.5065 \times ARIMAX\_Prediction\_W_2) - \\ & (26.1348 \times incidence\_rate) + \\ & (1.2706 \times incidence\_avg) - \\ & (0.5763 \times Last\_Two\_Avg) - 0.01 \end{aligned} \quad (9)$$

For one-week prediction, equations 6 and 8 each assign roughly the same weights to the BN and ARIMA/ARIMAX predictions. For two-week prediction, equation 7 assigns significantly lower weight to the BN prediction and equation 9 leaves it out altogether. In the four combination formulas, the relatively large magnitude coefficient on incidence rate is due to its relatively small range of values. The prediction accuracy of the (BN+ARIMA) and (BN+ARIMAX) models is shown in Table 1.

7 RESULTS AND DISCUSSION

We compared the accuracy for one- and two-week predictions of the two ensemble models with the BN and ARIMA models using data from three consecutive weeks from the months of July, September, and November.

Figure 2 shows one-week predictions of BN, ARIMA, ARIMAX, (BN+ARIMA) and (BN+ARIMAX) models versus actual incidence for the three three-week periods for a high incidence village. Figure 2(a) shows that the BN underestimates the peak in week 1 of the first month but does well with the remaining weeks of all three months. ARIMA overestimates weeks 1 and 3 of the first month while ARIMAX overestimates weeks 1 and 2 of the first month and week 1 of second month but both do well on the remaining low incidence weeks. This is consistent with the results in Table 1. The combinations (BN+ARIMA) and (BN+ARIMAX) correct for the major inaccuracies in the other models and perform equally well for this village with only slight differences in their predictions.

Figure 3 shows two-week predictions of BN, ARIMA, ARIMAX, (BN+ARIMA) and (BN+ARIMAX) models versus actual incidence for the same time periods for a second high incidence village. Figure 3(a) shows that the BN fits the three consecutive weeks of first two

Table 1. Prediction Accuracy of BN, ARIMA, ARIMAX, (BN+ARIMA) and (BN+ARIMAX) for one- and two-week prediction.

	Mean Absolute Error									
Models	BN		ARIMA		ARIMAX		(BN+ARIMA)		(BN+ARIMAX)	
Set of villages	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2
13-high	2.310	3.033	2.504	2.883	2.461	2.967	2.114	2.562	2.007	2.527
13-med	1.421	1.951	1.504	1.730	1.483	1.720	1.259	1.581	1.228	1.485
14-low	0.323	0.461	0.160	0.211	0.130	0.163	0.189	0.232	0.122	0.157
66-all	1.098	1.417	1.102	1.217	1.074	1.251	0.963	1.121	0.911	1.068

W_1 = First Week and W_2 = Second Week

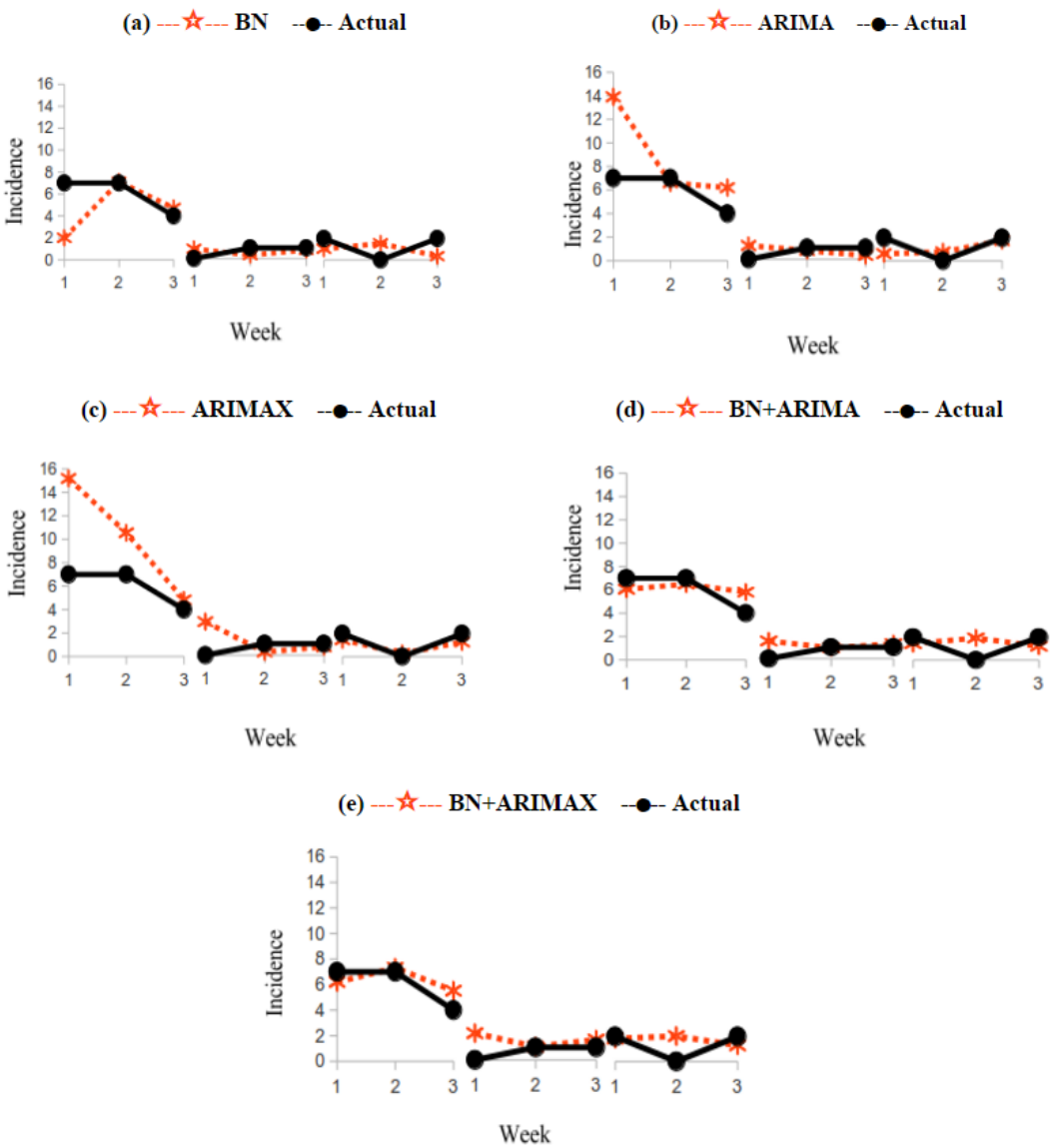


Figure 2. One week ahead malaria prediction versus actual for BN(a), ARIMA(b), ARIMAX(c), BN+ARIMA(d) and BN+ARIMAX(e) over a period of 3 weeks of 3 different months of a year for Village-1.

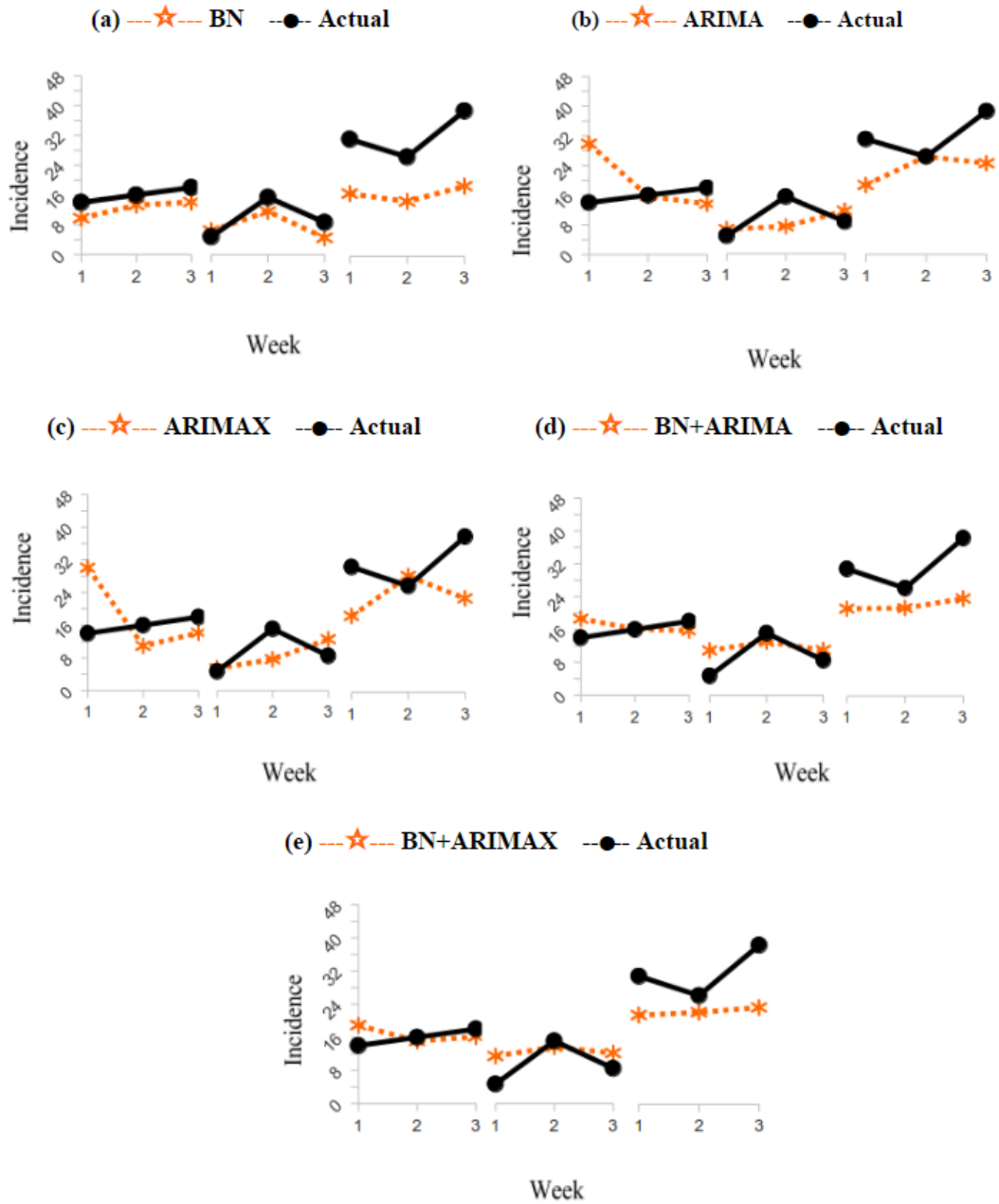


Figure 3. Two week ahead malaria prediction versus actual for BN(a), ARIMA(b), ARIMAX(c), BN+ARIMA(d) and BN+ARIMAX(e) over a period of 3 weeks of 3 different months of a year for Village-2.

months well but underestimates all three weeks of the last month. The ARIMA and ARIMAX models (Figure 3(b,c)) both overestimate the first week of the first month and underestimate weeks 1 and 3 of the last month. Again BN+ARIMA and BN+ARIMAX do better than all three individual models by correcting for the largest errors in the other models.

Table 2 shows the percentage improvement of performance of the combined BN+ARIMA model compared to BN, ARIMA, and ARIMAX for high, medium, and low incidence villages, as well as all 66 villages overall. Statistical significance was evaluated using a 2-talied t-test. All values are statistically significant ($p < 0.05$) except where indicated. The combined model outperforms the BN model with the difference statistically significant except for one week prediction for high incidence villages. The model performs worse than ARIMA and ARIMAX alone for one and two-week prediction for the low incidence villages. For all 66 villages BN+ARIMA outperforms the single models in all cases with the differences statistically significant except for ARIMAX W_2.

Table 3 shows the percentage improvement of performance of the combined BN+ARIMAX model compared to BN, ARIMA, and ARIMAX. This ensemble model now significantly outperforms the other three models in all cases except for three where the difference is not statistically significant. In particular, the improvement over ARIMAX for one and two week prediction is not statistically signif-

icant. Comparing the ensemble with the BN and ARIMAX models we see the largest improvement over the BN model for low incidence villages and the largest improvement over the ARIMAX model for the high incidence villages, which is reflective of the complementary strengths of the two models. For all entries in the tables, the BN+ARIMAX model outperforms the BN+ARIMA model.

Table 2. Performance Improvement(%) of BN+ARIMA over BN, ARIMA and ARIMAX.

Models	Percentage Improvement BN		ARIMA		ARIMAX	
Set of Villages	W_1	W_2	W_1	W_2	W_1	W_2
13-high	8.48*	15.53	15.58	11.13*	14.10*	13.65
13-med	11.40	18.96	16.29	8.61*	15.10	8.08*
14-low	41.49	49.67	-18.13	-9.95	-45.38	-42.33
66-all	12.30	20.89	12.61	7.89	10.34	10.39*

W_1 = First Week, W_2 = Second Week
All values are statistically significant ($p < 0.05$) except where indicated by *.

Table 3. Performance Improvement (%) of BN+ARIMAX over BN, ARIMA and ARIMAX.

Models	Percentage Improvement BN		ARIMA		ARIMAX	
Subset of villages	W_1	W_2	W_1	W_2	W_1	W_2
13-high	13.14	16.70	19.87	12.36*	18.47	14.84
13-med	13.58	23.87	18.35	14.14	17.19	13.65
14-low	62.14	65.97	23.56	25.64	5.92*	3.74*
66-all	17.08	24.63	17.38	12.24	15.22	14.63

W_1 = First Week, W_2 = Second Week
All values are statistically significant ($p < 0.05$) except where indicated by *.

8 CONCLUSION

In this paper we have taken the first essential step towards a system to support targeted malaria intervention using the data in Thailands E-Malaria Information system by developing a high resolution prediction model. We developed a Bayesian network model that represents the effect of environmental variables and captures nonlinear effects. Comparison with traditional ARIMA models showed that the two types of models have complementary strengths. Leveraging these complementary strengths, we developed an ensemble predictor that has significantly better accuracy than either predictor alone. Our results were obtained for one district in northern Thailand. A next step will be to test the generality of the model by applying it to districts with varying endemicity and environmental characteristics.

The structure of our Bayes net model can be used to provide causal explanations but by creating an ensemble in the way we did, we lose some of this explanatory power. It would thus be of potential benefit to seek to integrate the ARIMA model directly into the Bayes net. This might be done by including a node for the ARIMA result and a node that computes the weighted average of the models.

We intend to use this model as part of a decision support tool for targeted intervention of malaria. We are currently working to integrate it with a GIS to facilitate interaction and more intelligibly display results. An additional long-range goal is to apply our techniques the modeling of dengue.

ACKNOWLEDGEMENTS

This paper is based upon work supported by the U.S. Army International Technology Center Pacific (ITC-PAC) under Contract No. FA5209-15-P-0183. This work was also partially supported through a fellowship from the Hanse-Wissenschaftskolleg Institute for Advanced Study, Delmenhorst, Germany to Haddawy for collaborative work with the University of Bremen. We thank Saranath Lawpoolsri for providing the malaria data, Patiwat Sa-angchai for providing the environmental data, and Ngamphol Soonthornworasiri for advice on the ARIMA model.

REFERENCES

[1] R. Adhikari, 'A neural network based linear ensemble framework for time series forecasting', *Journal of Neurocomputing*, **157**, 231–242, (2015).
[2] R. Adhikari and R.K. Agrawal, 'A novel weighted ensemble technique for time series forecasting', *Lecture Notes in Computer Science*, **7301**, 38–49, (2012).
[3] W. Anggraeni, R.A. Vinarti, and Y.D. Kurniawati, 'Performance comparisons between arima and arimax method in moslem kids clothes demand forecasting: Case study', *Procedia Computer Science*, **72**, 630–637, (2015).
[4] G.E.P Box and G.M. Jenkins, *Time series Analysis: Forecasting and Control*, Holden-Day, California, 3rd edn., 1970.
[5] G.F. Cooper, D.H. Dash, J.D. Levander, W. Wong, W.R. Hogan, and M.M. Wagner, 'Bayesian biosurveillance of disease outbreaks', *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence (UAI '04)*, 94–103, (2004).
[6] L.A. Diaz-Robles, J.C. Ortega, J.S. Fu, G.D. Reed, J.C. Chow, J.G. Watson, and J. Moncada-Herrera, 'A hybrid arima and artificial neural network model to forecast particulate matter in urban areas: The case of temuco,chile', *Atmospheric Environment*, **42**, 8331–8340, (2008).
[7] W.M. Dlamini, 'A bayesian belief network analysis of factors influencing wildfire occurrence in swaziland, environmental modelling & software', *Environmental Modelling & Software*, **25**, 199–208, (2010).
[8] A. Gomez-Elipe, A. Otero, M. van Herp, and A. Aguirre-Jamie, 'Forecasting malaria incidence based on monthly case reports and environmental factors in karuzi, burundi,1977-2003', *Malaria Journal*, **6**, (2007).
[9] K.W. Hipel and A.I. McLeod, *Time Series Modelling of Water Resources and Environmental Systems*, ELSEVIER, Amsterdam, 1994.
[10] R.J. Hyndman. Forecasting functions for time series and linear models. `http://CRAN.R-project.org/package=forecast`, 2014.
[11] S. Johnson, F. Fielding, G. Hamilton, and K. Mengersen, 'An integrated bayesian network approach to lyngbya majuscula bloom initiation', *Marine Environmental Research*, **69**, 27–37, (2010).
[12] S. Johnson, K. Mengersen, A. de Waal, K. Marnewick, D. Cillers, A.M. Houser, and L Boast, 'Modelling cheetah relocation success in southern africa using an iterative bayesian network development cycle', *Ecological Modeling*, **221**, 641–651, (2010).
[13] A. Khamsiriwatchara, P. Sudathip, Sawang, S. Vijakadge, Potithavoranan T, A. Sangvichean, W. Satimai, C. Delacollette, P. Singhasivanon, S. Lawpoolsri, and J. Kaewkungwal, 'Artemisinin resistance containment project in thailand. (i): Implementation of electronic-based malaria information system for early case detection and individual case management in provinces along the thai-cambodian border', *Malaria Journal*, **11**, (2012).
[14] M. Khashei, M. Bijari, and G.A.R. Ardali, 'Hybridization of autoregressive integrated moving average(arima) with probabilistic neural networks(pnns)', *Journal of Computer & Industrial*, **63**, 37–45, (2012).
[15] C. Kongcharoen and T. Kruangpradit, 'autoregressive integrated moving average with explanatory variable(arimax) model for thailand export', *Internation Journal of forecasting*, (2013).

[16] M.A. Kulkarni, R.E. Desrochers, and J.T. Kerr, 'High resolution niche models of malaria vectors in northern tanzania: A new capacity to predict malaria risk?', *PLoS One*, **5**, (2010).
[17] L. Liu, R.S. Luan, F. Yin, X.P. Zhu, and Q. Lu, 'Predicting the incidence of hand, foot and mouth disease in sichuan province, china using the arima model', *Epidemiology and Infection*, **144**, 144–151, (2015).
[18] Thailand Ministry of Public Health. Report on weekly malaria situation. http://www.thaivbd.org/n/home, 2015.
[19] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan-Kaufmann, San Francisco, 1st edn., 1988.
[20] S. Promprou, M. Jaroensutasinee, and K. Jaroensutasinee, 'Forecasting dengue haemorrhagic fever cases in southern thailand using arima models', *Dengue Bulletin*, **30**, 99–106, (2006).
[21] A.M. Tompkins and V. Ermert, 'A regional-scale, high resolution dynamical malaria model that accounts for population density, climate and surface hydrology', *Malaria Journal*, **12**, (2013).
[22] K. Wangdi, P. Singhasivanon, T. Silawan, S. Lawpoolsri, N.J. White, and J. Kaewkungwal, 'Development of temporal modelling for forecasting and prediction of malaria infections using time-series and arimax analyses: a case study in endemic districts of bhutan', *Malaria Journal*, **9**, (2010).
[23] World Health Organization (WHO). World malaria report 2015. http://apps.who.int/iris/bitstream/10665/200018/1/9789241565158_eng.pdf?ua=1, 2015.
[24] J.D. Wichard, 'Forecasting the nn5 time series with hybrid models', *International Journal of Forecasting*, **27**, 700–707, (2011).
[25] L. Wilkinson, Y.E. Chee, A.E. Nicholson, and P. Quintana-Ascencio, 'An object-oriented spatial and temporal bayesian network for managing willows in an american heritage river catchment', *UAI Workshop on Models for Spatial, Temporal, and Networked data*, (2013).
[26] I. H. Witten and E. Frank, *Data Mining, Practical Machine Learning Tools and Techniques*, Morgan-Kaufmann, San Francisco, 2nd edn., 2005.
[27] D.H. Wolpert, 'Stacked generalization', *Neural Networks*, **5**, 241–259, (1992).
[28] G.P. Zhang, 'Time series forecasting using a hybrid arima and neural network model', *Journal of Neurocomputing*, **50**, 159–175, (2003).
[29] Y. Zhang, P. Bi, and J. Hiller, 'Meteorological variables and malaria in a chinese temperate city: A twenty-year time-series data analysis', *Environment International*, **36**, (2010).
[30] K. Zinszer, A.D. Verma, K. Charland, T.F. Brewer, J.S. Brownstein, Z. Sun, and Buckeridge, 'A scoping review of malaria forecasting: past works and future directions', *BMJ Open*, **2**, (2012).

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

doi:10.3233/978-1-61499-672-9-1791

Rapid Adaptation of Air Combat Behaviour

Armon Toubman[1], Jan Joris Roessingh[1], Pieter Spronck[2], Aske Plaat[3], and Jaap van den Herik[4]

Abstract. Adaptive behaviour for computer generated forces enriches training simulations with appropriate challenge levels. For adequate insight into the range of possible behaviour, the adaptation has to take place in a rapid fashion. Ideally, each new behaviour model should remain readable by (and thereby under the control of) human experts. Although various attempts have been made at creating adaptive behaviour, current solutions require large numbers of simulations. Moreover, usability by end users has been of subordinate interest, as is compliance with doctrine and ethics. In this work, we present a machine learning method that enables fast behaviour adaptation, while keeping the behaviour models in a human-readable format. We demonstrate the effectiveness of the proposed method in beyond-visual-range air combat simulations.

1 INTRODUCTION

The use of training simulations for defence applications is growing [1]. Commercial off-the-shelf simulation packages, such as STAGE [2] and Virtual Battlespace (VBS) [3], allow experts to quickly develop and operate scenarios for simulations. To make the scenarios more realistic, they are often inhabited by computer generated forces (CGFs).

Traditionally, CGF behaviour is scripted using if-then rules which map observations to actions. However, writing good scripts requires domain expertise, which is a costly resource. Poorly written scripts have low training value, as no skills learned by the trainee are transferable to the real world. Furthermore, trainees can learn to purposefully exploit bad CGF behaviour. This is usually counterproductive and should be discouraged [4, 5]. Dedicated CGF behaviour authoring tools, such as Smart Bandits [6], have been developed to mitigate this issue, often by introducing enhanced user interfaces and ready-to-use behaviour modules. However, it is still up to the experts to design the CGF behaviour and adapt this behaviour to reach specific training goals.

Nowadays, advances in the field of machine learning offer the prospect of automatically generating behaviour models and adapting these models online (i.e., during operation in training simulations). Automatic generation of behaviour models has the potential to greatly decrease the workload of CGF developers, while online adaptation can increase the training value of CGFs by continuously challenging the trainees. Over the years, various machine learning approaches have been tried, yet adaptive capabilities in CGF behaviour authoring tools are still rare [5, 7]. Adoption of these approaches is tamed by the large amount of time needed for quality control: proposed machine learning methods for CGF behaviour generation require substantial processing times, and produce behaviour models that are hard for human end users to understand. The latter is a critical feature, as end users need to be able to verify that generated behaviour complies with doctrine, and by extension, ethics. This consideration is also important in related fields, e.g., the development of behaviour for autonomous unmanned vehicles, or decision support systems for human pilots.

In this work, we present a machine learning method that is specifically focused on rapid generation of understandable behaviour models. The method entails the adaptation of behaviour represented as finite-state machines (FSMs), through a reinforcement learning technique called dynamic scripting.

FSMs have been successfully used to represent CGF behaviour in the Smart Bandits behaviour authoring tool, which is currently in use by the Royal Netherlands Air Force (RNLAF) to control CGFs in beyond-visual-range air combat training simulations. By cutting up the FSMs into their constituent states and transitions, the dynamic scripting algorithm is able to efficiently recombine the FSMs and provide adaptive behaviour. Furthermore, in contrast to many other machine learning algorithms, dynamic scripting does not alter defined pieces of behaviour during the learning process, which is a great step towards keeping generated behaviour in line with military doctrine, and keeping behaviour models understandable by experts.

To the best of our knowledge, this is the first work developing adaptive capabilities for two cooperative CGFs in 2v2 beyond-visual-range air combat. In this work, we actively take into account (1) computational speed, (2) usability by end users, and (3) built-in ethical and doctrinal consideration.

The rest of this paper is structured as follows: Section 2 gives an overview of related work. Section 3 describes the integration of FSMs into dynamic scripting. Section 4 shows the experimental setup used to test the adaptive CGFs. The results of the experiments are presented in Section 5 and discussed in Section 6. Finally, section 7 concludes the paper.

2 RELATED WORK

Air combat is the fight between armed aircraft. It can be represented as a ‘game’ with a large, continuous state space, a variety of available actions, and limited resources (see [8] for a complete treatise). When generating air combat behaviour, creative solutions are required while being bound by tactical doctrine and rules of engagement (and training goals, for training simulations).

[1] Department of Training, Simulation & Operator Performance, Netherlands Aerospace Centre, Netherlands, email: {Armon.Toubman,Jan.Joris.Roessingh}@nlr.nl
[2] Tilburg center for Cognition and Communication, Tilburg University, Netherlands, email: p.spronck@gmail.com
[3] Leiden Institute of Advanced Computer Science, Leiden University, Netherlands, email: aske.plaat@gmail.com
[4] Leiden Centre of Data Science, Leiden University, Netherlands, email: jaapvandenherik@gmail.com

Air combat is usually divided into within-visual-range (WVR) combat (also known as air combat manoeuvring or dogfighting) and beyond-visual-range (BVR) combat, in which combating aircraft engage each other with long-range sensors and weapons. WVR and BVR combat require different approaches: WVR is often modelled as a pursuit-evasion problem, consisting of complex manoeuvring and rapid decision-making, whereas BVR requires planning and higher-level strategical thinking.

2.1 Machine learning for air combat behaviour

A wide range of machine learning techniques has been tried to efficiently generate effective WVR and BVR air combat behaviour. A non-exhaustive overview of these approaches is given below. The research in this area is quite fragmented, not only between WVR and BVR combat, but also between simulation environments and experimental methods. While this means that no absolute comparisons can be made among reported results, the reported parameters may serve as an indication of the computational complexity of the methods.

Neural networks have been applied in various ways with varying success. Early work with neural networks includes the use of a three-layer back-propagation network by Rodin and Amin [9] for predicting and countering WVR tactical manoeuvres. Rodin and Amin report "successfully training" their network after 60,000 iterations. More recently, Teng et al. [10] applied self-organizing neural networks with a Q-learning component for online learning of WVR behaviour. The resulting behaviour models were evaluated in small-scale human-in-the-loop experiments. The learning network was able to reach a 93% mean win ratio after 120 episodes against a statically controlled CGF. Furthermore, the network peaked at a 40% win ratio against pilots in training, and below 10% against experienced pilots. Teng et al. report using available air combat doctrine for building the state- and action-space for the Q-learning component [11], by encoding expert knowledge as if-then rules.

Evolutionary algorithms have also been used in various forms. Mulgund et al. [12] applied a genetic algorithm to optimize tactical parameters for many-versus-many BVR engagements. Starting from a scenario with equal losses on both sides, their algorithm was able to develop tactics by which all enemy CGFs were defeated, without any casualties on the friendly side. However, only few parameters are reported. In a follow-up study, Zhang et al. [13] used 40 generations, with a population size of 80. Smith et al. applied a learning classifier system to develop novel one-versus-one WVR tactics for an experimental fighter jet [14, 15]. A population of 200 rules is reported, tested throughout 300 generations. Furthermore, air combat tactics have been described through grammars, which have been used as templates for genetic programming algorithms (see, e.g., [16] and [17], both BVR). Expressing tactics through grammars limits the search space, ensuring that only valid behaviour is generated. However, large numbers of simulations are seemingly needed to reach convergence using this method, with for example [16] reporting convergence near 50% fitness after 100,000 simulations.

While a large number of simulations may be acceptable for exploratory studies such as [15], or offline learning before human-in-the-loop trials, it poses a problem in the case of learning online during training simulations. A CGF, trying to adapt its behaviour to that of a human participant, only has limited time to do so between engagements. Furthermore, trainees can only participate in a limited number of simulations, which constrains the time available to adapt even further.

2.2 Transparency of behaviour

Apart from the time to adapt, the transparency of generated behaviour models is of great importance. Behaviour models generated for military applications should be usable (editable, readable, testable, etc.) by different end users, e.g., scenario developers and training instructors [7]. Techniques such as neural networks and evolutionary algorithms often produce behaviour models that are hard to decode, understand, and manually edit.

In earlier work, we have made attempts at generating BVR air combat behaviour using dynamic scripting [18, 19]. Dynamic scripting is a reinforcement learning method that takes a rule base, and recombines the rules from this rule base into scripts [20]. This method does not 'invent' new behaviour, and instead relies on the rule base being filled with rules based on expert knowledge. As a result, the generated behaviour can only be as good or bad as the knowledge contained in the rule base. Applying a pure dynamic scripting solution in the air combat domain has yielded encouraging results, however the technique remains to be validated in a production environment.

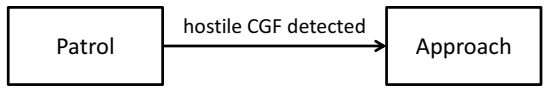
Rather than having experts write if-then rules, a more intuitive method of defining behaviour is the use of finite-state machines [21]. This is also the method used in Smart Bandits [6, 22, 23], the CGF behaviour authoring tool developed by the Netherlands Aerospace Centre, and currently in use by the Royal Netherlands Air Force. Each CGF controlled by Smart Bandits is in a certain state, and each state has associated actions. However, Smart Bandits provides no adaptive capabilities. As Smart Bandits provides both (1) a drag-and-drop interface for authoring CGF behaviour, usable by various end users, and (2) an established repository of well-tested CGF behaviour that is actively being used in training simulations, it is an ideal testing ground for introduction of adaptive behaviour.

3 ADAPTIVE FINITE-STATE MACHINES

In Smart Bandits, CGFs are controlled with FSMs. When FSMs are to control CGF behaviour, the states of the FSM are linked to pieces of related behaviour [21]. For example, a *Patrol* state may correspond to a CGF repeatedly moving between two points in the simulated world (see Figure 1a). A transition to another state then occurs when a certain change in the world state is perceived by the CGF. Continuing the example, if the CGF is in the *Patrol* state and detects a hostile CGF, it might transition to the *Approach* state in which the CGF will move towards the detected CGF. The example above can be expressed as a set of rules, as shown in Figure 1b.

The resulting rules can now be stored in a rule base, which serves as the input for the dynamic scripting technique. As mentioned in Section 2, dynamic scripting [20] is a rule-based reinforcement learning technique. When the dynamic scripting algorithm is initialized with a rule base, it assigns a weight value to each rule in the rule base. Before each episode (in our case, a simulated air combat encounter), a predefined number of rules are drawn from the rule base through roulette wheel selection, in which each rule is represented by its weight. Together, the rules that are drawn from the rule base form the script that governs the behaviour

(a)




```
if state == Patrol:
  if near(point_a): move_to(point_b)
  if near(point_b): move_to(point_a)
```

(b)

```
if state == Approach:
  move_to(detected_hostile_CGF)

if state == Patrol and hostile_CGF_detected():
    set_state(Approach)
```

Figure 1. Representing a behavior controller as a finite-state machine, (a) graphically and (b) as rules.

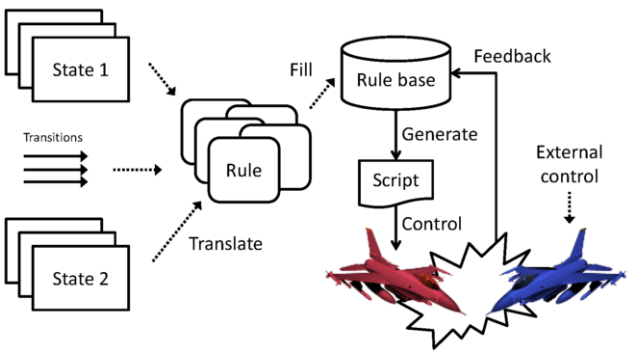


Figure 2. Adaptive finite-state machines through dynamic scripting.

of a CGF during an encounter with an opponent. At the end of the encounter (i.e., when one side is defeated and the simulation ends), a fitness value is calculated for the script, and this value is fed back to the rule base. The rule base updates the weights of the rules according to the fitness, in such a manner that rules that contributed to a high fitness value are rewarded with a weight increase, resulting in an increased probability of being selected the next time that a script is generated. Similarly, a low fitness results in a decrease of the weights of rules that contributed to this fitness value. The entire process of creating adaptive FSMs through dynamic scripting is illustrated in Figure 2. Through the use of behaviour rules, this process also enables the implementation of ethical decision-making. So far we have not concentrated on that topic, but we have set aside space in our technique for future implementations.

In the original description of dynamic scripting, rules are selected probabilistically, under the assumption that all rules are valid choices for inclusion in a script. However, for our goals these assumptions are invalid, as each state and each transition should be represented in a generated script. Not doing so could lead to scripts containing invalid FSMs. Two steps are required to resolve this issue. First, for a non-empty subset of states and transitions in the FSM, we create multiple interchangeable implementations, i.e., rules that trigger on the same conditions. These implementations express different but equally valid behaviours. In the case of states, each implementation provides behaviour that can be displayed in that state. In the case of transitions, each implementation provides a valid transition between states based on some conditions. Second, we alter the original dynamic scripting rule selection algorithm such that all states and transitions are represented in each script that

Algorithm 1. Script generation

Input: A rule base containing one or more implementations for each state and transition in a FSM.

Output: A script containing a rule for each state and transition in the FSM.

```
script = []
for element in fsm.get_elements():
  # fsm.get_elements() returns all states and
  # transitions in the FSM for which an
  # implementation needs to be included
  # in the script
  sum_of_weights = 0
  candidate_rules = []
  for rule in rule_base:
    # the rules in rule_base that are an
    # implementation of the current element are
    # added to a list of candidates for selection
    if rule.is_implementation_of(element):
      candidate_rules.append(rule)
      sum_of_weights += rule.weight
    end if
  end for
  if sum_of_weights == 0:
    # should the sum of the weights of the current
    # candidates be zero, we select a candidate at
    # random for inclusion in the script
    selected_rule = random.choice(candidate_rules)
    script.append(selected_rule)
  else:
    # we select a rule from candidate_rules through
    # roulette wheel selection based on the weights
    # of the candidate_rules
    selected_rule = roulette_wheel(candidate_rules)
    script.append(selected_rule)
  end if
end for
return script
```

is generated. This ensures that each generated script contains a completely valid FSM, and the proper set of rules concerning human values. This updated rule selection algorithm is shown in Algorithm 1.

As an example, consider the *Patrol* state from Figure 1a. One of the implementations of this state can be the rule definition as found in Figure 1b. An alternative implementation could be defined that directs the CGF aircraft to patrol in a triangular pattern a, b, and c rather than between points a and b. This implementation would be expressed by writing a new rule. Implementations of state transitions can be defined in a similar way, by using alternative preconditions for the rules governing the state transitions.

Air combat is a complex problem in a high-dimensional state space. Capturing air combat behaviour in rules greatly reduces the complexity of the generated behaviour models. The expert knowledge embedded in the rules enables the definition of behaviour for large parts of the state space, thereby quickly covering a large part of all possible situations. Furthermore, the dynamic scripting algorithm only recombines pieces of behaviour, and does not invent new pieces of behaviour. While this limits creativity, it also makes the behaviour generation system, and thereby the air combat task, easier to control and understand, compared to traditional machine learning methods. Finally, dynamic scripting is expected to converge quickly to satisfactory behaviour, as only a limited set of FSMs can be generated.

4 METHOD

To determine whether the method described in the previous section is capable of fast behaviour adaptation, we implemented the method in an air combat simulation using the STAGE [2] simulation environment. In this simulation, two cooperating CGFs, both controlled using an adaptive FSM, were tasked with the combat of two CGFs using static (non-adaptive) behaviour.

To be a suitable replacement for static CGFs, the adaptive CGFs should perform at least as well as the static CGFs. For this reason, we compare the performance of the adaptive CGFs to that of static CGFs using an FSM as currently found in Smart Bandits. Furthermore, to demonstrate the adaptive CGFs' adaptive capabilities, the adaptive CGFs will be placed in scenarios where they have to adapt from either an arbitrary initialization or after they have tuned their parameters to a previous opponent. These scenarios are analogous to generating good behaviour before any training by a human participant takes place, and adapting to changes in a human participant's behaviour during training.

The rest of this section describes the CGFs and the simulations in more detail.

4.1 CGFs

Both the static and adaptive teams consisted of two fighter jets (lead and wingman) equipped with radar, a radar warning receiver, and four semi-active long range missiles. The tactics used by the teams are described in Subsection 4.1.1 and 4.1.2.

4.1.1 Adaptive team

The FSMs used by the adaptive CGFs were based on an operational 2-versus-2 tactic. This tactic consists of two phases. The first phase is the opening sequence of the tactic, in which the CGFs detect the opposing CGFs, select an approach formation and assign targets between themselves. In the second phase, the CGFs engage and fire at their targets, after which they re-evaluate their tactical situation and either evade incoming missiles or select new targets.

For the adaptive CGFs, the tactic was subdivided into ten states in rule form. For this tactic, no meaningful new transitions could be identified, and as a result the original transitions were embedded in the rules created for the states. Next, new, additional implementations of selected states were designed and added as rules. Together with the original states and transitions, these rules formed the rule base for the adaptive CGFs. In total, 8 new states were added, resulting in a rule base with 18 rules. The adaptive lead and wingman were each assigned their own copy of the rule base, so that they could each optimize their own behaviour.

4.1.2 Static team

The scripts used by the static CGFs were based on one of two tactics. The first tactic (Tactic 1) was the same as the tactic used by the adaptive CGFs, resulting in a *mirror match*. By letting the adaptive CGFs fight against their own tactic, we will be able to show that they are able to improve on their own tactic using only a few extra variations of states. The second tactic (Tactic 2) was specifically designed to counter this tactic, to force the adaptive CGFs to come up with a creative solution.

Using these two tactics for the static team allows us to show different features of using the adaptive FSMs. By including the second tactic, we are able to show the adaptive capabilities of the adaptive CGFs, after they have already adapted to another tactic. This is in essence a form of transfer learning [19]. The ability to rapidly adapt to new tactics is important, as human trainees only spend a limited amount of time in a simulator, and ideally the adaptation of the adaptive CGFs is evident within that timeframe.

4.2 Learning parameters

We performed two types of simulations. First, the adaptive CGFs engaged the static CGFs using either Tactic 1 or Tactic 2 in fifty consecutive episodes, allowing the adaptive team to adapt to both tactics separately. In these cases, a baseline was set by engaging the static team with CGFs using the original (non-adaptive) Smart Bandits tactic. Second, the adaptive team, having already adapted to either Tactic 1 or Tactic 2, engaged the static team using the other tactic in fifty consecutive episodes. This demonstrates the "online" adaptivity of the adaptive CGFs. Each scenario was repeated ten times to obtain average performance data. For the baselines, each scenario was only repeated five times, as no learning took place.

Each trial ended when (1) a fighter jet on either side was hit with a missile[2], or (2) both sides had used all of their missiles, or (3) ten minutes of simulated time had passed. If an adaptive CGF had hit a static CGF, the adaptive team was declared the winner of the episode. In all other cases, the static team was declared the winner, even in a situation where no adaptive CGF was hit.

The dynamic scripting algorithm requires a fitness value as input, by which the proper weight adjustments are calculated. Earlier work determined the accumulated *probabilities-of-kill* of missiles fired to be effective fitness values for learning in the air combat domain [18]. However, we were unable to retrieve the necessary values to implement the probability-of-kill fitness from the STAGE API. Instead, a fitness of 1 was given to the winning team, and a fitness of 0 to the losing team. The weight adjustments were calculated as shown in Equation 1.

$$adjustment = max\ (\ -25,\ 50 * (\ (2 * fitness\ ) - 1\ )\ ) \qquad (1)$$

According to this Equation 1, the maximum possible reward is higher than the maximum possible punishment. This results in an algorithm that moves quicker into (local) optima than stepping back out of them.

5 RESULTS

We recorded which team successfully ended each episode, and calculated the win ratio as the number of wins divided by total number of repetitions of each episode. On average, each series of fifty episodes took 3.5 hours of real-time simulation.

Figure 3a shows the performance of the adaptive CGFs against the static CGFs using Tactic 1. The baseline CGFs fighting these static CGFs results in a mean win ratio of 0.46. The adaptive CGFs quickly converge to and hold a mean win ratio of 0.55, from episode 2 onwards. Optimal performance (0.80 mean win ratio) is first reached at episode 12, and again at episodes 37 and 41.

[2] Being outnumbered, the remaining team member is assumed to flee.

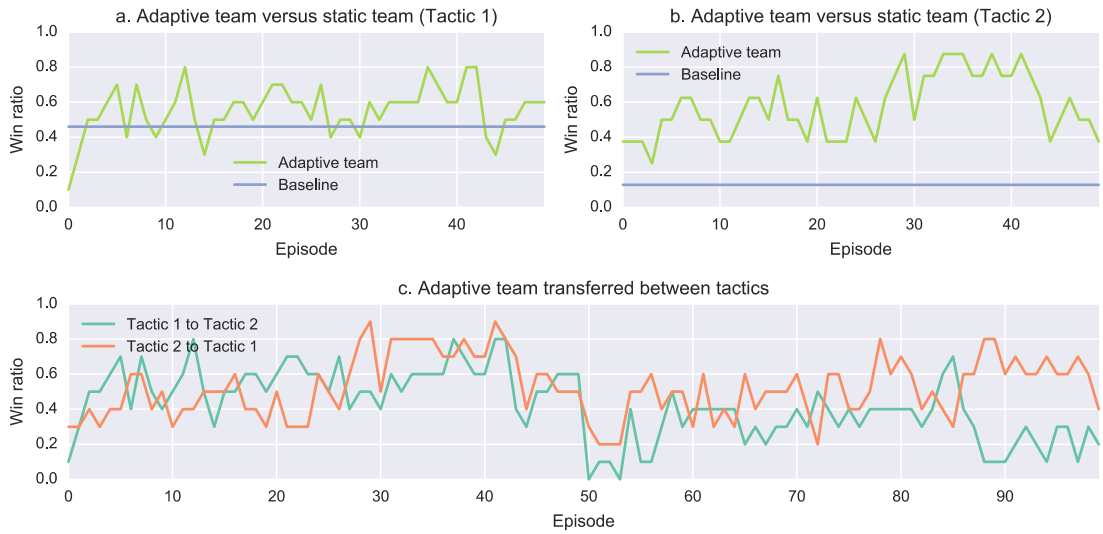


Figure 3. Performance of the adaptive CGFs against the static CGFs.

Figure 3b shows the same as Figure 3a, except for the static CGFs using Tactic 2. The baseline CGFs fighting these static CGFs results in a mean win ratio of 0.13. The adaptive CGFs' performance oscillates around 0.50 until episode 27. Between episodes 28 and 42, the performance spikes to a mean win ratio of 0.8, after which it drops again to the 0.50 level.

Figure 3c shows the performance of the adaptive CGFs when engaging the static CGFs, after the latter changed from Tactic 1 to Tactic 2 (green curve) and from Tactic 2 to Tactic 1 (orange curve). For the first 50 episodes, the same data is used that is also shown in Figures 3a and 3b. The remaining 50 episodes show the performance against the newly introduced tactics. The first peak reached in both cases are a mean win rate of 0.70 at episode 85 (Tactic 1 to Tactic 2), and 0.80 at episode 78 (Tactic 2 to Tactic 1).

6 DISCUSSION

The purpose of this study was to determine whether the method described in Section 3 is capable of fast adaptation of air combat behaviour. We tested this adaptive capability against static opponents that acted using two different tactics.

For the baselines, we relied on the performance of CGFs using the Smart Bandits tactic defined by experts. Figure Figure **3**a shows a win ratio near 0.50, which is expected as both sides repeatedly use the same tactics. However, random factors in the simulation environment (e.g., the hit rate of missiles) can still influence encounters.

Figures 3a and 3b show how well the adaptive CGFs are able to adapt to the two tactics employed by the static CGFs. Against Tactic 1, the adaptive CGFs are able to improve the baseline win ratio of 0.46, to a maximum of 0.80. This is a noteworthy result, as it shows that even with a limited amount of extra states, and given that the tactic taken from Smart Bandits was already optimized by experts, our algorithm was still able to further optimize the adaptive team's behaviour. During the design of scenarios, such a function may prove useful to aid the designers of opposing CGFs even before any training of human pilots takes place.

As mentioned in Subsection 4.1.2, the static CGFs' Tactic 2 was designed to defeat Tactic 1, which was employed in a non-adaptive manner in the baselines. The result is apparent in Figure 3b, with the baseline performance only reaching a 0.13 mean win ratio. The adaptive CGFs present a more positive picture. Although at first the performance stays around the 0.50 level, a new optimum is reached around episode 28. This optimum is maintained for about 15 episodes, after which the performance suddenly reverts to the old level. The high optimum indicates that the adaptive CGFs had good options (i.e., rules/states) to choose from, and the dynamic scripting algorithm was able to find the right combination quite efficiently.

The drop between episodes 40 and 45 signifies a certain brittleness of the system, as the adaptive CGFs are not able to hold their optimal solution. This is most likely caused by the random factors in the simulation environment, as mentioned earlier. A possible solution might be to introduce a memory of well-performing tactics, and to occasionally retry those tactics once the performance is dropping. Against static opponents, such memorized tactics could greatly mitigate the effect of random factors, and thereby increase the win ratio. Against other adaptive opponents (such as human trainees), retrying previously successful tactics may prove beneficial as well, especially if no other local optimum has been found for some time.

Of course, the adaptive CGFs had more options (i.e., pieces of behaviour) available to them than the baseline CGFs, meaning that the fact that they were able to defeat the static CGFs more often is not an impressive result by itself. However, what does matter is that the system can reach new performance levels, and maintain these levels for a significant amount of time. Furthermore, our system is able to let CGFs adapt their behaviour relatively fast, certainly when compared to systems employing creative methods such as neural networks and evolutionary algorithms.

An important use case for adaptive behaviour is online adaptation, i.e., adapting to the behaviour of human trainees during training. Figure 3c shows how well the adaptive CGFs can adapt to opponents using a new tactic, after having already adapted to earlier opponents with a different tactic. In both cases, a similar pattern is visible: the performance of the adaptive CGFs immediately dips when the new tactic is introduced, after which a moderate (0.40-0.60) performance level is held until a peak is reached around episode 80. With this kind of plasticity, the CGFs can quickly react to new tactics that human trainees may try out against them. Furthermore, with the low number of episodes needed to reach good behaviour (with e.g., a ≥0.5 win ratio), it

becomes feasible to run faster-than-real-time simulations between human-in-the-loop training sessions. This opens up the possibility of continuous adaptivity with a minimum amount of downtime, while keeping maximal control over the generated behaviour.

As mentioned in Section II, computational speed is crucial for machine learning in training simulations. Existing methods required large numbers of computational cycles to adapt CGF behaviour. Therefore, with regard to training simulations, the rapid adaptation as presented in this paper forms a substantial improvement over the existing methods. Field trials are currently underway with opponents controlled using adaptive FSMs flying against active RNLAF F-16 pilots. The results of these trials will be reported in the near future.

7 CONCLUSION

We have developed a machine learning method that is able to rapidly adapt the behaviour of CGFs to that of their opponents. The adaptive power of this method was shown in simulated air combat experiments. Compared to earlier work, the proposed method is computationally inexpensive and requires few iterations to generate good behaviour. Furthermore, the resulting behaviour models are in a format that is easily readable by human experts. This enables experts to effectively verify that the generated CGF behaviour complies with training goals and doctrine, including ethical decision-making. With adaptive CGFs as presented in this paper, military training simulations can be made more challenging and effective, leading to armed forces that are better prepared to defend shared values.

Future work includes evaluating the behaviour of the adaptive CGFs in human-in-the-loop trials, scaling up to engagements involving larger numbers of CGFs, and automatically setting up behaviour to realize predefined training goals in training simulations.

ACKNOWLEDGMENTS

The authors thank Remco Meiland, Joost van Oijen, Kees Krikke, Arnoud van Leeuwen, Nico Zink, and Aleid Duiker for their technical support and RNLAF Lt. Col. Roel Rijken for his continuous input.

REFERENCES

[1] J. D. Fletcher, "Education and training technology in the military," *Science (New York, N.Y.),* vol. 323, pp. 72-75, January 2009.

[2] Presagis. (2015). *STAGE*. Available: http://www.presagis.com/files/product_brochures/2015-04-DS-SIM-STAGE14_web.pdf

[3] B. I. Simulations. (2014). *Virtual Battlespace 3*. Available: https://bisimulations.com/sites/default/files/BISim_VBS3_V2B_VBSIG_only_preview_2014%20%282%29.pdf

[4] R. Lopes and R. Bidarra, "Adaptivity challenges in games and simulations: a survey," *Computational Intelligence and AI in Games, IEEE Transactions on,* vol. 3, pp. 85-99, 2011.

[5] N. Abdellaoui, A. Taylor, and G. Parkinson, "Comparative Analysis of Computer Generated Forces' Artificial Intelligence," in *RTO-MP-MSG-069 - Current uses of M&S Covering Support to Operations, Human Behaviour Representation, Irregular Warfare, Defence against Terrorism and Coalition Tactical Force Integration*, Brussels, Belgium, 2009.

[6] NLR. (2016). *Smart Bandits AIR*. Available: http://www.nlr.org/capabilities/smart-bandits-air/

[7] A. Toubman, G. Poppinga, J. J. Roessingh, M. Hou, L. Luotsinen, R. A. Løvlid*, et al.*, "Modeling CGF Behavior with Machine Learning Techniques: Requirements and Future Directions," in *Proceedings of the 2015 Interservice/Industry Training, Simulation, and Education Conference*, Orlando, Florida, 2015, pp. 2637-2647.

[8] R. L. Shaw, *Fighter Combat*: Naval Institute Press, 1985.

[9] E. Y. Rodin and S. M. Amin, "Maneuver prediction in air combat via artificial neural networks," *Computers & mathematics with applications,* vol. 24, pp. 95-112, 1992.

[10] T.-H. Teng, A.-H. Tan, and L.-N. Teow, "Adaptive computer-generated forces for simulator-based training," *Expert Systems with Applications,* vol. 40, pp. 7341-7353, 2013.

[11] T.-H. Teng, A.-H. Tan, Y.-S. Tan, and A. Yeo, "Self-organizing neural networks for learning air combat maneuvers," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, 2012, pp. 1-8.

[12] S. Mulgund, K. Harper, K. Krishnakumar, and G. Zacharias, "Air combat tactics optimization using stochastic genetic algorithms," in *Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics*, 1998, pp. 3136-3141.

[13] K. Zhang, W. Li, and Z. Wang, "Large-Scale Air-Combat Formation Optimization Using Simulated-Annealing GA (Genetic Algorithm)," in *Proceedings of the 24th Congress of International Council of the Aeronautical Sciences*, Yokohama, Japan, 2004.

[14] R. E. Smith, B. A. Dike, B. Ravichandran, A. El-Fallah, and R. K. Mehra, "The fighter aircraft LCS: A case of different LCS goals and techniques," in *Learning Classifier Systems*, ed: Springer, 2000, pp. 283-300.

[15] R. E. Smith, B. A. Dike, R. K. Mehra, B. Ravichandran, and A. El-Fallah, "Classifier systems in combat: two-sided learning of maneuvers for advanced fighter aircraft," *Computer Methods in Applied Mechanics and Engineering,* vol. 186, pp. 421 - 437, 2000.

[16] J. Yao, Q. Huang, and W. Wang, "Adaptive Human Behavior Modeling for Air Combat Simulation," in *2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 2015, pp. 100-103.

[17] R. Kop, A. Toubman, M. Hoogendoorn, and J. Roessingh, "Evolutionary Dynamic Scripting: Adaptation of Expert Rule Bases for Serious Games," in *Current Approaches in Applied Artificial Intelligence*. vol. 9101, M. Ali, Y. S. Kwon, C.-H. Lee, J. Kim, and Y. Kim, Eds., ed: Springer International Publishing, 2015, pp. 53-62.

[18] A. Toubman, J. J. Roessingh, P. Spronck, A. Plaat, and J. Van den Herik, "Rewarding Air Combat Behavior in Training Simulations," in *Proceedings of the 2015 IEEE International Conference on Systems, Man and Cybernetics*, Hong Kong, 2015, pp. 1379-1402.

[19] A. Toubman, J. J. Roessingh, P. Spronck, A. Plaat, and J. Van den Herik, "Transfer Learning of Air Combat Behavior," in *Proceedings of the 2015 IEEE International Conference on Machine Learning and Applications*, Miami, Florida, 2015.

[20] P. Spronck, M. Ponsen, I. Sprinkhuizen-Kuyper, and E. Postma, "Adaptive game AI with dynamic scripting," *Machine Learning,* vol. 63, pp. 217-248, 2006.

[21] D. Fu and R. Houlette, "The Ultimate Guide to FSMs in Games," in *AI Game Programming Wisdom*. vol. 2, S. Rabin, Ed., ed: Charles River Media, 2004, pp. 283-302.

[22] J. J. Roessingh, R.-J. Merk, P. Huibers, R. Meiland, and R. Rijken, "Smart Bandits in air-to-air combat training: Combining different behavioural models in a common architecture," in *21st Annual Conference on Behavior Representation in Modeling and Simulation*, Amelia Island, Florida, USA, 2012.

[23] A. Khatami, P. Huibers, and J. J. Roessingh, "Architecture for goal-driven behavior of virtual opponents in fighter pilot combat training," in *Proceedings of the 22nd Annual Conference on Behavior Representation in Modeling and Simulation*, Ottawa, Canada, 2013.

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

doi:10.3233/978-1-61499-672-9-1797

An Intelligent System for Personalized Conference Event Recommendation and Scheduling

Aldy GUNAWAN and **Hoong Chuin LAU** and **Pradeep VARAKANTHAM** and **Wenjie WANG** [1]

Abstract.

Many conference mobile apps today lack the intelligent feature to automatically generates optimal schedules based on delegates' preferences. This entails two major challenges: (a) identifying preferences of users; and (b) given the preferences, generating a schedule that optimizes his preferences. In this paper, we specifically focus on academic conferences, where users are prompted to input their preferred keywords. Our key contribution is an integrated conference scheduling agent that automatically recognizes user preferences based on keywords, provides a list of recommended talks and optimizes user schedule based on these preferences. To demonstrate the utility of our integrated conference scheduling agent, we first demonstrated the app in the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2015) and conducted a survey to collect some data, which are used to verify the results presented in this paper. It is able to provide well calibrated results with respect to precision, accuracy and recall. We also tested the app in the 2015 WI-IAT International Conference (Singapore). The android and web-based apps have been demonstrated and deployed in AAMAS 2016 (Singapore) with positive responses from the users.

1 Introduction

In a large conference setting where talks are presented in parallel sessions across multiple days, it is challenging for a conference attendee to generate a plan of talks to attend that optimize his/her preferences. Furthermore, this adds to the cognitive challenge if the conference venue is large, where one may need to consider time to travel between talks. To reduce this cognitive load, we aim to provide an integrated conference scheduling agent that not only identifies user preferences (based on keywords) but also generates a schedule of talks to attend at different times of the conferences while considering the user preferences. We are specifically interested in academic conferences where data associated with users is easily available.

Both the individual problems (understanding user preferences and optimizing schedule accounting for preferences) have received significant interest in existing work. The first thread of related research is with respect to learning user preferences given papers has been studied extensively in the machine learning community. Statistical topic modeling has become a popular method for analyzing large sets of text collections by representing high dimensional data in a low dimensional subspace [21]. The topic model is built using MALLET, which is introduced by Andrew McCallum and his team in 2002 [10]. MALLET is able to navigate large bodies of information by finding clusters of keywords that frequently appear together, called topics. The second thread of related work is with respect to optimizing preferences given constraints on scheduling talks. This problem is related to a single resource scheduling problem with the objective of maximizing the profitability of the resulting schedule under fixed processing times [20].

One of the best known systems in the area of academic conference event recommendation is Conference Navigator 3.0 [13]. In Conference Navigator system, users directly select preferred talks. It also collects the wisdom of the user community and makes it available through community-based recommendation interface to help users in making scheduling decisions.

Our key contribution is in providing an integrated solution for both these problems and demonstrate utility on a real conference scheduling problem. Specifically, we first employ MALLET to identify the topics of interest for a given conference, by considering papers from that conference. We then identify preferences of a given user for the topics of interest at the conference by getting the user's preferred keywords. PRESS also considers community-based recommendation in terms of the correlation among talks. These correlation values are calculated automatically based on their similarity in terms of keywords provided by the users. Based on preferred keywords, PRESS provides a list of recommended talks and optimizes user schedule based on these preferences.

For easy interaction with the users, our agent is built as an application for mobiles, namely PRESS. So, we are able to take change requests on the generated schedule and immediately provide an updated schedule. To demonstrate utility for conference attendees, we first demonstrated PRESS in the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2015) and conducted a survey to collect some data, which are used to verify the results presented in this paper. We show that the papers generated in the schedules for the users have high values of precision, accuracy and recall. We then tested PRESS in the 2015 WI-IAT International Conference (Singapore). Some feedbacks especially related to the client-facing android mobile app were collected. Finally, both android and web-based versions of PRESS have been deployed in AAMAS 2016 [7].

2 Related Work

Resnick and Varians [15] describe a recommender systems as follows: *In a typical recommender system people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients. In some cases the primary transformation is in the aggregation; in others the system's value lies in its ability to make good matches between the recommenders and those seeking recommendations*.

[1] Singapore Management University, Singapore, email: {aldygunawan, hclau, pradeepv, wjwang}@smu.edu.sg

Adomavicius and Tuzhilin [1] provide a survey of the-state-of-the art and possible extensions of the recommender system. Burke et al. [4] describe two basic principles of a recommender system: a) it is personalized to optimize the experience of one user, and b) it is intended to help the user choose among discrete options. Recommender systems have been developed in various domains of applications, such as LIBRA [11] (book recommender) and INTIMATE [9] (movie recommender).

Lops et al. [8] describe two main paradigms of recommender systems. *Content-based* recommender systems generate recommended items based on items that have been liked by a user in the past, whereas *Collaborative* recommendation systems try to recommend items from other users whose preferences are similar to those of the user and recommend items they have liked. In this paper, we concentrate purely on content-based recommendation since our collected data is from a small community of users.

One method that have been used in content-based recommendation is Latent Dirichlet Allocation (LDA). LDA is a fully generative probabilistic topic model. Probabilistic topic models play an important rule in order to capture latent topical information from a large collection of data [12]. The basic underlying idea of probabilistic topic models is documents are mixtures of topics, where a topic is a cluster of words that frequently occur together [17]. By using contextual clues, topic models connect words with similar meanings and distinguish between uses of words with multiple meanings.

MALLET provides an option to use a previously generated inference file as an inference tool [10]. It uses LDA. Each document is produced by selecting a distribution over topics, and then generating each keyword at random from a topic chosen by using the selected distribution. [21] implement different methods for topic inference, such as Gibbs sampling and SparseLDA in the MALLET toolkit on streaming two different sets of documents, 13 years of full papers published in the NIPS conference and a set of journal article abstracts from Pubmed. Other applications of MALLET are in analyzing a set of personal emails [19] and a set of ratings collected on Amazon Mechanical Turk [6].

Sampson [16] introduces "preference-based" conference scheduling (PBCS) problem. Instead of looking at the conference scheduling problem as a classical scheduling problem, the problem is treated from the customer point of view with the main objective is related to a customer-satisfaction. Other works related to the conference scheduling problem can be referred to [14, 18].

Bhardwaj et al. [3] introduce COBI as the most recent web-based, visual scheduling interface in planning a large-scale conference. COBI engages the community to play an active role in the planning process. A process that collects input from attendees and considers them as preferences and constraints in the planning process. To the best of our knowledge, no existing work incorporates the optimization mathematical model in the process of providing the recommendation papers.

3 The Proposed Approach

The overall architecture of PRESS is depicted in Figure 1. PRESS consists of four main components: Native android application (Front-End), Back-end Engine, Optimization Engine and Text Analyzer.

In the following, we provide the formal definition and formulation of the problem in the context of a large academic conference. We further explain the MALLET implementation in Text Analyzer component and two different proposed algorithms in the Optimization Engine component.

3.1 Problem Formulation

A conference consists of a set of main sessions where each main session is scheduled on one particular time period (e.g. from 09.00 - 10.00 am). In most large conferences, each main session is divided into a set of parallel sessions. We assume that each parallel session is scheduled in a particular room. Figure 2 shows an example of a conference setting on a particular day.

Let *P* be a set of papers that will be presented during a conference. Each parallel session consists of a number of talks. In order to generate a schedule that possibly contains talks across sessions, we divide each time period into multiple number of time slots (e.g. every 15 minutes). Each time slot will have one talk and only one paper $i \in P$ would be presented in that time slot for that session. We also assume that each paper will only be presented once throughout the conference. We implement MALLET to generate a set of topics T from P. Each topic $j \in T$ contains a set of keywords W_j^1 that is likely to appear together in topic j [17]. We assume that $|W_j^1| = |W^1|$ ($\forall j \in T$). See Figure 3 for an illustration.

Some methodological issues faced when using MALLET, such as how to determine the values of $|T|$ and $|W^1|$, affect the quality of the outputs. At the moment, the best way to determine the values of $|T|$ and $|W^1|$ is to run multiple analyses with different values of both and comparing the results that seem to fit "best" [2].

In summary, MALLET generates two different outputs (Figures 4 and 5) that would be kept in the database and used as inputs for the optimization engine:

- $\mathbf{M}_{|T| \times |W^1|} = [w_{jk}^1]$, where w_{jk} represents keyword k of topic j ($\forall j \in T, k \in W_j^1$).
- $\mathbf{U}_{|P| \times |T|} = [u_{ij}]$ where u_{ij} represents the utility score of paper i related to topic j ($\forall i \in P, j \in T$).

Let W_i^2 be the set of keywords stated on paper $i \in P$. As mentioned in Section 1, we consider both keywords generated by MALLET and from papers directly and both would be kept in the database.

3.2 MALLET Implementation

The Text Analyzer component consists of two sub-components: the PDFMINER tool and the MALLET topic model package. Take note that both sub-components: PDFMiner and MALLET, are run off-line and generated results would be kept in the database. PDFMiner (`https://pypi.python.org/pypi/pdfminer/`) is a tool for extracting information from PDF documents. This sub-component is responsible for converting a collection of documents (eg. pdf files) into text files and then tagging the part of speech of words in these text files.

In most cases, information has no structure, some pre-processing steps are required to convert unstructured information and extract structured relevant information. The Illinois Chunker (`https://cogcomp.cs.illinois.edu/page/software_view/Chunker`) is used to identify the semantically related words by assigning different tags. For example, in the noun words "reinforcement learning", the word "reinforcement" is identified as the beginning word of a noun phrase and therefore tagged with B-NP (begins a noun phrase), however, the following word "learning" is identified inside the same noun phrase as "reinforcement" and therefore tagged with I-NP (inside a noun phrase). Likewise, other types of phrases such as a verb phrase will be tagged with B-VP (begins a verb phrase) and I-VP (inside a noun phrase), respectively.

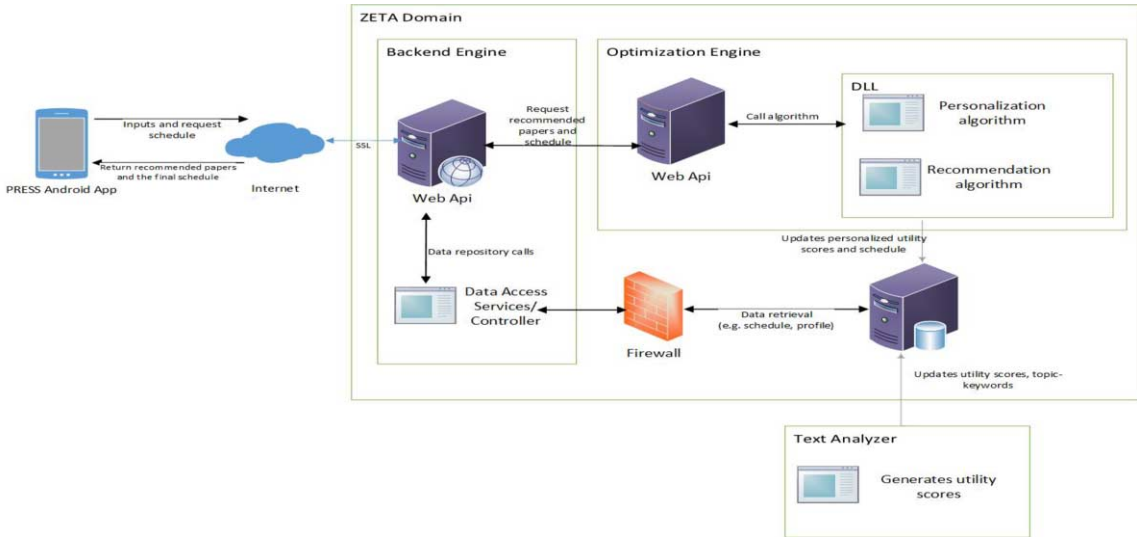


Figure 1: System Architecture of PRESS

Time Period	Main Session	Room			
		Room 1	Room 2	Room 3	Room 4
08.00 - 09.00	Session#1	PS1	PS2	PS3	PS4
09.00 - 10.00	Session#2	PS1	PS2	PS3	PS4
10.00 - 10.30	Coffee break				
10.30 - 11.30	Session#3	PS1	PS2	PS3	PS4
11.30 - 12.30	Session#4	PS1	PS2	PS3	PS4
12.30 - 14.30	Lunch time				

PS: parallel session

Figure 2: Example of conference setting

Time Slot	Parallel Session			
	Room 1	Room 2	Room 3	Room 4
08.00 - 08.15	Paper#1	Paper#2	Paper#3	Paper#4
08.15 - 08.30	Paper#5	Paper#6	Paper#7	Paper#8
08.30 - 08.45	Paper#9	Paper#10	Paper#11	Paper#12
08.45 - 09.00	Paper#13	Paper#14	Paper#15	Paper#16

Figure 3: Example of talks in a particular time period

The second sub-component, the MALLET topic model package [10], is used to extract a set of topics and the highest frequent words for each topic from the text documents and output the statistics of each extracted topic for each text document. MALLET allows us to filter a standard list of English stop-words from documents before processing. Unfortunately, we cannot edit the contents of this list without modifying code and recompiling. In order to rule out some trivial words, we create an extra-word file containing those trivial words.

Figure 4 shows the screenshot of the MALLET output. There are 11 topics generated with 5 keywords for each topic. The topics that compose each document including the statistics of each topic can be seen in Figure 5. For example, PAPER 1 has topic 10 as its principal topic, at about 82.1%; topic 15 at 25.8 % and so on. The topic model also suggests a connection among documents that might not at first have suspected. PAPERS 1, 2, 3 and 4 have topic 10 as their principal topic.

3.3 Proposed Algorithms

Given a set of keywords K that the user is interested in and the results of MALLET tools, we calculate the personalized utility score for each talk and generate a list of recommended talks.

	A	B	C	D	E	F
0	leader	trust	landfill trash	society	merchant	
1	complex returns	bisimulation	autonomous mobile agents	normative systems	collective behaviour	
2	cybersecurity	ranking	envy-free division	ranks	manipulation	
3	theorem	preference	modal logic	logic	reason	
4	tactics	crowd robotics	robot	control	team	
5	election	vote	manipulation	voter	maximin	
6	planning	endogenous discounting	learning	complexity	judgment aggregation	
7	autonomy	learning	platform	planning	attack surface	
8	reasoning	analysts	knowledge base	defects	complexity	
9	technology	violation	obligations	guard function	shared information	
10	mechanism design	market	advertisers	government	partial verification	

Figure 4: Screenshot of MALLET output

	#doc	name	topic	proportion	...							
0	Paper 1	10	0.821207	15	0.25842	23	0.061721	13	0.03214	0	0.017127	
1	Paper 2	10	0.726088	15	0.249853	23	0.082568	18	0.068656	13	0.052124	
2	Paper 3	10	0.870099	15	0.226938	3	0.051107	13	0.038831	19	0.010294	
3	Paper 4	10	0.797018	15	0.26128	23	0.091565	13	0.047208	17	2.63E-04	
4	Paper 5	23	0.737454	10	0.30123	15	0.224803	13	0.033752	17	2.48E-04	
5	Paper 6	22	0.839755	15	0.200866	13	0.065629	10	0.053941	17	0.035124	
6	Paper 7	24	0.741729	15	0.154617	13	0.075463	23	0.020853	16	0.004314	

Figure 5: Screenshot of topic composition

Personalization Algorithm

We present the personalization algorithm for providing a list of recommended talks, as shown in Algorithm 1. The objective is to calculate $\tilde{u}_{ij}$,the modified utility score of paper $i \in P$ related to topic $j \in T$, with respect to the set of keywords K given by the user. We compare the number of keywords $|K|$ which are matched with a set of keywords W_j^1 of topic j, represented as Tot_j ($\forall j \in T$). For each paper i, the utility score u_{ij} is multiplied by Tot_j in order to get the value of $\tilde{u}_{ij}$. Finally, we calculate the total personalized utility score of paper i, $TotU_i = \sum_{j \in T} \tilde{u}_{ij}$ ($\forall i \in P$) (LINES 1 - 18).

The next step is to compare K with the keywords from paper i, W_i^2 ($\forall i \in P$). If a match exists, the value of $TotU_i$ will added by one for each matched keyword (LINES 19 - 27). For each user, all papers would be sorted in descending order with respect to the values of $TotU_i$ (LINES 28 - 29). The recommendation is given from the top $x\%$ of papers. This is a naive way in order to provide a list of recommended talks without considering possible conflicts.

The user will then select or remove some talks from the list. Those selected talks would be in the "must-go" and "must-skipped" lists, respectively. PRESS continues to call the recommendation algorithm in order to provide the final schedule that maximizes the total personalized utility score and ensures there is no conflicts among talks.

Algorithm 1 Personalization Algorithm

for $h = 1\ to\ |K|$ **do**
  for $j = 1\ to\ |T|$ **do**
    $Tot_j = 0$
    for $k = 1\ to\ |W^1|$ **do**
      if (h^{th} keyword from the user is matched with k^{th} keyword of topic j) **then**
        $Tot_j += 1$
      end if
    end for
  end for
end for
for $i = 1\ to\ |P|$ **do**
  for $j = 1\ to\ |T|$ **do**
    $\tilde{u}_{ij} = Tot_j \times u_{ij}$
  end for
end for
for $i = 1\ to\ |P|$ **do**
  $TotU_i = \sum_{j \in T} \tilde{u}_{ij}$
end for
for $h = 1\ to\ |K|$ **do**
  for $i = 1\ to\ |P|$ **do**
    for $k = 1\ to\ |W^2|$ **do**
      if (h^{th} keyword from the user is matched with k^{th} keyword of paper i) **then**
        $TotU_i += 1$
      end if
    end for
  end for
end for
Rank all papers based on $TotU$ values in the descending order
return the top $x\%$ of papers

Recommendation Algorithm

In the recommendation algorithm, we introduce a mathematical model to formulate the scheduling problem. The time slots of talks are taken into consideration in this model. The mathematical programming model is solved by the commercial solver CPLEX Optimization Studio 12.6.1.

The scheduling problem is defined as follows. We define $MUST$ and $SKIP$ as "must-go" and "must-skip" lists, respectively. Let assume the conference is held within a set of days D. Each day $d \in D$ is divided into a set of time slots S_d. Each time slot $s \in S_d$ on day $d \in D$ consists of a set of parallel sessions N_{ds}. A talk would be held in one parallel session at each time slot.

The decision variable X_{dsn} is a binary variable. Its value equals to 1 if a talk in parallel session n on day d at time slot s is selected.

$$Maximize \sum_{d \in D} \sum_{s \in S_d} \sum_{n \in N_{ds}} \hat{u}_{dsn} \times X_{dsn} \quad (1)$$

The objective function (1) is to maximize the total personalized utility score of selected talks. Let $\hat{u}_{dsn}$ is the utility score of the talk in parallel session $n \in N_{ds}$ on day $d \in D$ at time slot $s \in S_d$. The utility scores are collected from $TotU_p$ ($p \in P$) values with respect to the time slot. For example, if paper p_1 is presented on Day 1, time slot 1 and parallel session 1, the value the talk $\hat{u}_{111} = TotU_{p_1}$.

$$\sum_{k \in N_{ds}} X_{dsn} \leq 1 \ \ \forall d \in D, s \in S_d \quad (2)$$

Equation (2) ensures that at each time slot, only one talk is attended.

$$X_{dsn} = 1 \ \ \forall (d, s, n) \in MUST \quad (3)$$

Equation (3) ensures that talks in the "must-go" list, $MUST$, are attended.

$$X_{dsn} = 0 \ \ \forall (d, s, n) \in SKIP \quad (4)$$

Equation (4) enforces that talks are in the "must-skip" list, $SKIP$, would not be attended since they are out of the user interest.

$$X_{dsn} \leq M \times \hat{u}_{dsn} \ \ \forall d \in D, s \in S_d, n \in N_{ds} \quad (5)$$

Equation (5) guarantees that only talks with non-zero personalized utility scores would be selected. Let M be a very large number.

4 Architecture and System Design

Figure 1 illustrates the various individual components and their interactions. All communications among main components are implemented by using RESTful web service published on one of Singapore University Management servers, called ZETA server.

Android Application (Front-end Engine)

This is a client-facing android mobile app that allows a user to enter preferred keywords, view recommended talks, select preferred talks (indicated as "must-go"), remove non-preferred talks (indicated as "must-skip") and view the final schedule. This component serves as an interface for the user to construct the user profile. All information provided by the user will be sent to the back-end engine.

Back-end Engine

This component is responsible for coordinating and delegating tasks between the front-end and the optimization engines. The back-end engine is also responsible for storing and retrieving all information related to the conference in the database, including keywords from papers and text analyzer outputs.

First, it collects the user-profile from the front-end engine and pass it to the optimization engine. The optimization engine will call the personalization algorithm in order to generate a list of recommended talks. This list would be passed back to the front-end engine so the user can indicate and select his preferred talks ("must-go") and remove some non-preferred talks (must-skip").

The back-end then consolidates "must-go" and "must-skip" lists together with other information from database, such as conference schedule, and passed to the optimization engine. The recommendation algorithm will be called in order to generate the final schedule. At the end, the back-end engine pass back the final schedule to the front-end engine an display it to the user.

Optimization Engine

The optimization engine consists of two algorithms: personalization and recommendation algorithms. As described in Section 3, this component interacts with the back-end engine in order to generate the list of recommended papers and the final schedule.

5 Experimental Results

5.1 User Study Details

PRESS was first demonstrated during the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-15) which was held from 4 - 8 May 2015 in Istanbul, Turkey. The conference consists of 6 main sessions. Each main sessions is labeled by an alphabet which represents a particular time period, e.g. main session B is held on Wednesday (6 May 2015) from 11.00 -

12.30. Each main session is further divided into 5 different parallel sessions, numbered from 1 - 5. Each talk is given a predetermined time slot (e.g. 15 minutes). In total, there are 166 talks. Each parallel session is related to one of particular research area/topic, such as Game Theory, Applications and others. The detailed schedule, including the information about the papers, can be found in `http://www.aamas2015.com/en/program.asp`.

In order to verify the effectiveness of PRESS, a user survey was conducted at AAMAS-15. We collected 45 respondents from the AAMAS-15 participants. Each respondent was asked to specify his/her preference keywords together with the list of talks he/she would be interested to attend. This collection of surveys serve as the ground truth and would be used for analysis purpose.

5.2 System Components

We also tested the app in the 2015 WI-IAT International Conference. Some feedbacks especially related to the client-facing android mobile app (e.g. the design of a sign-up page, the layout and so on) have been collected. We include some final screenshots for the Android app. The opening screen requests the user either to sign in or to register (Figure 6(a)). The registration is required for the first times (Figure 6(b)). The user also needs to agree with the terms and conditions of the app (Figure 6(c)). Figure 6(d) summarizes the profile of the registered user.

Figure 7(a) shows the screen for the user to input the preferred keywords. Once the arrow button on the right top corner is clicked, the list of recommended talks which are generated by the personalization algorithm (Algorithm 1) would be displayed. The user then select and remove some talks. Those would be treated as "must-go"and "must-skip" lists, as shown in Figure 7(b). The details of one particular talk can also be displayed (Figure 7(c)). All those information would be sent back to the back-end engine and the recommendation algorithm would be called. Finally, the final schedule for each day would be displayed, as seen in Figure 7(d).

5.3 Insights

After demonstrating PRESS and conducting a survey at the AAMAS-15, we analyze the goodness of PRESS in recommending the list of talks. Out of 14 research areas, the top three most selected areas are Application, Game Theory and Learning which cover up to 42%. Due to a short time taken for each survey, we assume that a user will not be able to exhaustively select all preferred talks. Hence, based on a set of selected talks, we include an additional set of selected talks which have high correlation values with those talks. All those talks are considered as the talks selected by a user as well. The higher the correlation value is, the more similar two papers are in terms of topics including keywords generated. The correlation between two talks is calculated using the Cosine Coefficient formula:

$$\cos(i,i') = \frac{\sum_{j\in T} u_{ij}u_{i'j}}{\sqrt{\sum_{j\in T} u_{ij}^2}\sqrt{\sum_{j\in T} u_{i'j}^2}} \quad \forall (i,i') \in P \tag{6}$$

We evaluate the performance of PRESS by comparing three statistical measures: accuracy, precision and recall rates. The accuracy is the proportion of true results (true positives and true negatives) among the total number of cases examined. Precision (positive predictive value) is the fraction of retrieved cases that are relevant, while recall (sensitivity) is the fraction of relevant cases that are retrieved. Precision can be seen as a measure of quality, whereas recall is a measure of quantity.

By setting the numbers of user-selected papers from the ground truth and recommended papers generated by PRESS to a cut-off of top 10% $\times$ 166 talks which equals to 16 talks with the highest total personalized utility scores and a cut-off correlation value (e.g. 0.75), our experimental results show that the accuracy, precision and recall rates of PRESS are 92.02%, 58.61% and 58.61%, respectively. Other results with different cut-off correlation values can also be seen in Table 1.

We conclude that the higher the cut-off correlation value, the lower the values of measures are. It is intuitive correct since the selected talks by the user during the survey would be fewer. If we do not include talks with high correlation values, the three measures are much lower since the users are not aware with similar talks.

Table 1: Statistical measures

Correlation value	Measure		
	Accuracy	Precision	Recall
0.75	92.02%	58.61%	58.61%
0.80	91.73%	57.08%	57.08%
0.85	91.62%	56.53%	56.53%
0.90	91.57%	56.25%	56.25%
0.95	90.87%	52.64%	52.64%

6 Conclusion

We introduce a personalized event scheduling recommender system, PRESS. PRESS is an android mobile app that gathers personalized information from a user and recommends talks. Although there is a bunch of recommender systems in different domains, so far as we are concerned that PRESS is the first android app that incorporates an optimization model for generating a feasible schedule.

We demonstrated PRESS at AAMAS-15 in Istanbul, Turkey. The generated predictions by PRESS is compared against the ground truth. We observe that PRESS achieves reasonable accuracy, precision and recall rates. Some feedbacks have also been collected during the 2015 WI-IAT conference. We have also deployed the android and web-based versions of PRESS during AAMAS-16 (Singapore). Positive responses have been given by around 140 users.

The current version of PRESS uses the direct keyword matching among keywords generated by MALLET and provided by the user. We will consider more advanced techniques which allow going beyond the direct keyword matching. We also consider other possible scenarios. Some talks may be scheduled in more than one timeslot so the attendee has to decide which timeslot should be attended. This is related to the capacity constraint of rooms which is currently negligible. Other combinations of precision/recall (e.g. product of criteria or using a varying linear coefficient) will also be included for our future work.

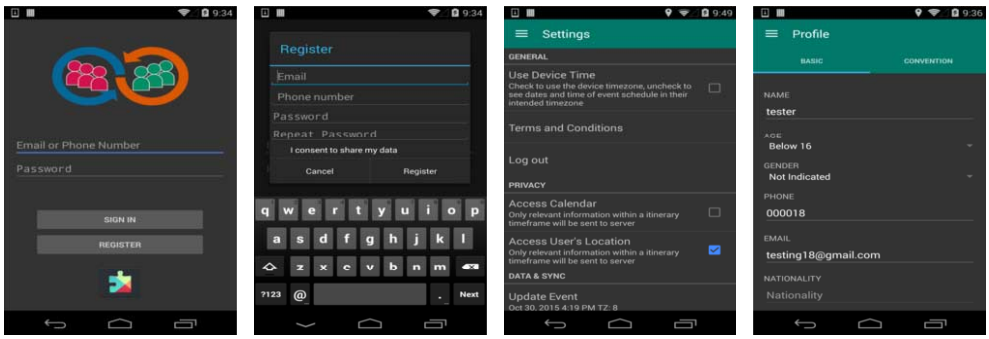
ACKNOWLEDGEMENTS

This research is supported by Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative as well as its Corp Lab@University scheme. The authors wish to thank people who have contributed to this project: Shih-Fen Cheng, Oi Mei Wong, Chun Pong Fan, Elizabeth Lim, Eng Chye Low, Firmansyah Abd Rahman, Xiang Li, Khin Aye Mon & Htar Htar Hlaing.

REFERENCES

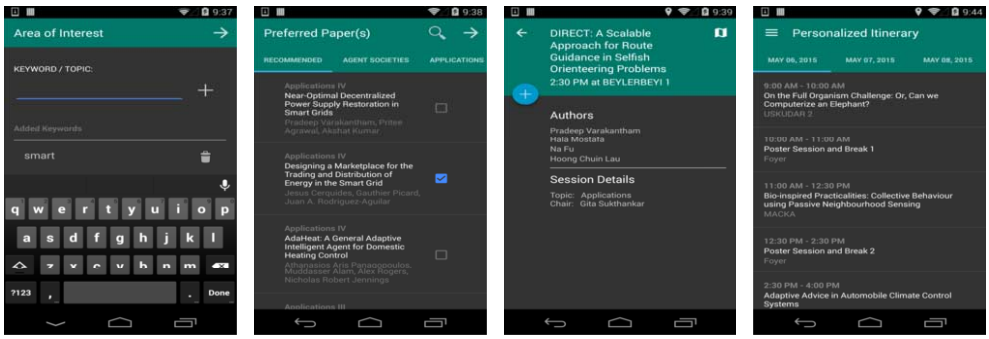
[1] G. Adomavicius and A. Tuzhilin, 'Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions', *IEEE Transactions on Knowledge and Data Engineering*, **17**(6), 734–749, (2005).

[2] L. AlSumait, D. Barbará, J. Gentle, and C. Domeniconi, 'Topic signif-



(a) Front screen (b) Registration (c) Setting (d) Basic profile

Figure 6: Screenshots of PRESS showing: a) the opening screen, b) the registration screen, c) settings of the app and d) the user profile



(a) Keywords selection (b) Recommended papers (c) Paper details (d) Schedule

Figure 7: Screenshots of PRESS showing: a) the list of selected keywords, b) the list of recommended talks, c) the details of one particular talk and d) the final schedule

icance ranking of LDA generative models', in *Machine Learning and Knowledge Discovery in Databases*, eds., W. Buntine, M. grobelnik, D. Mladenić, and J. Shawe-Taylor, volume 5781 of *Lecture Notes in Computer Science*, 67–82, Springer, (2009).

[3] A. P. Bhardwaj, J. Kim, S. Dow, D. R. Karger, S. Madden, R. Miller, and H. Zhang, 'Conference scheduling: a personalized approach', in *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2014)*, pp. 2–10, (2014).

[4] R. Burke, A. Felfernig, and M. H. Goker, 'Recommender systems: an overview', *AI Magazine*, **32**(3), 13–18, (2011).

[5] O. Celma and X. Serra, 'FOAFing the music: bridging the semantic gap in music recommendation', *Web Semantics*, **6**(4), 250–256, (2008).

[6] J. Chuang, S. Gupta, C. Manning, and J. Heer, 'Topic model diagnostics: Assessing domain relevance via topical alignment', in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, eds., S. Dasgupta and D. Mcallester, volume 28, pp. 612–620, (2013).

[7] H. C. Lau, A. Gunawan, P. Varankantham, and W. Wang, 'PRESS: PeRsonalized Event Scheduling recommender System (Demo track)', in *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016)*, (2016).

[8] P. Lops, M. de Gemmis, and G. Semeraro, 'Content-based recommender systems: state of the art and trends', in *Recommender Systems Handbook*, eds., F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, 73–105, Springer, (2011).

[9] H. Mak, I. Koprinska, and J. Poon, 'INTIMATE: a web-based movie recommender using text categorization', in *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, pp. 602–605. IEEE Computer Society, (2003).

[10] A. K. McCallum, 'MALLET: A machine learning for language toolkit'. http://mallet.cs.umass.edu, 2002.

[11] R. J. Mooney and L. Roy, 'Content-based book recommending using learning for text categorization', in *Proceedings of the 5th ACM Conference on Digital Libraries*, pp. 195–204. ACS Press, (2000).

[12] M. Ovsjanikov and Y. Chen, 'Topic modeling for personalized recommendation of volatile items', in *Machine Learning and Knowledge Discovery in Databases*, eds., J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, volume 6322 of *Lecture Notes in Computer Science*, 483–498, Springer, (2010).

[13] D. Parra, W. Jeng, P. Brusilovsky, C. Lopez, and S. Sahebi, 'Conference Navigator 3.0: an online social conference support system', in *Proceedings of the 20th Conference on User Modeling, Adaptation, and Personalization (UMAP 2012)*, (2012).

[14] J. Quesnelle and D. Steffy, 'Scheduling a conference to minimize attendee preference conflicts', in *Proceedings of the 7th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2015)*, pp. 379–392, (2015).

[15] P. Resnick and H. R. Varian, 'Recommender systems', *Communications of the ACM*, **40**(3), (March 1997).

[16] S. E. Sampson, 'Practical implications of preference-based conference scheduling', *Production and Operations Management*, **13**(3), 205–215, (2004).

[17] M. Steyvers and T. Griffiths, 'Probabilistic Topic Models', in *Latent Semantic Analysis: A Road to Meaning*, Hillsdale, NJ: Erlbaum, (2007).

[18] B. Vangerven, W. Passchyn, A. Ficker, D. Goossens, and F. Spieksma, 'Conference scheduling: a personalized approach', in *Proceedings of the 30th Annual Conference of the Belgian Operations Research Society*, pp. 15–16, (2016).

[19] X. Wang and A. McCallum, 'Topics over time: A non-markov continuous-time model of topical trends', in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 424–433. ACM, (2006).

[20] B. Yang and J. Geunes, 'A single resource scheduling problem with job-selection flexibility, tardiness costs and controllable processing times', *Computers & Industrial Engineering*, **53**(3), 420–432, (2009).

[21] L. Yao, D. Mimno, and A. McCallum, 'Efficient methods for topic model inference on streaming document collections', in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 937–946, New York, NY, USA, (2009). ACM.

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

doi:10.3233/978-1-61499-672-9-1803

Continuous Live Stress Monitoring with a Wristband

Martin Gjoreski[12] and **Hristijan Gjoreski**[12] and **Mitja Luštrek**[12] and **Matjaž Gams**[12]

Abstract. In this paper we propose a method for continuous stress monitoring using data provided by a commercial wrist device equipped with common physiological sensors and an accelerometer. The method consists of three machine-learning components: a laboratory stress-detector that detects short-term stress every 2 minutes; an activity recognizer that continuously recognizes user's activity and thus provides context information; and a context-based stress detector that first aggregates the predictions of the laboratory detector, and then exploits the user's context in order to provide the final decision in a 20 minute interval. The method was trained on 21 subjects in a laboratory setting and tested on 5 subjects in a real-life setting. The accuracy on 55 days of real-life data was 92%. The method is currently being implemented as a smartphone application, which will be demonstrated at the conference.

1 Introduction and motivation

Stress is a process triggered by a demanding physical and/or psychological event [12]. It is not necessarily a negative process, but continuous exposure can result in chronic stress, which has negative health consequences such as raised blood pressure, bad sleep, increased vulnerability to infections, decreased mental performance and slower body recovery [11]. It also has substantial economic consequences: the European Commission estimated the costs of work-related stress at €20 billion a year due to absence from work and decreased productivity [1]. Therefore, a stress-detection system would be useful for self-management of mental (and consequently physical) health of workers [3], students and others in the stressful environment of today's world.

Thanks to the recent technological advances, some of the stress-response components (e.g., increased heart rate) can be captured using an unobtrusive wrist device equipped with sensors, e.g., Empatica[3] or Microsoft Band. Our method is also based on the data captured by such a device, on which we use advanced machine learning (ML) along with context information.

The pioneers in the field of stress detection are Healey and Picard who showed in 2005 that stress can be detected using physiological sensors [5]. Since 2005, various studies were conducted to implement stress detection using a combination of signal processing and ML using data from physiological sensors and accelerometers [5][6][7][9][10]. The problem of stress detection was first analyzed in constrained environments such as a laboratory/office [10], car [5], and call center [6]. Some approaches in which the subjects were allowed to be active based on a predefined scenario came one step closer to the real world [9]. Most recently, Hovsepian et al. [7] proposed cStress, a method for continuous stress assessment in real-life using a chest belt. Similarly, our method is tested in real life, however, we use a commercial wrist device instead of a chest belt. For future work Hovsepian et al. [7] suggested better handling of physical activity (which can reduce stress detection performance) and using context information in the process of stress detection – which is what we have done in our study.

2 Method for stress detection in real life

For the purpose of this study, two datasets were recorded: a laboratory dataset, which includes 21 subjects, and a real-life dataset, which includes 5 subjects. The Empatica[2] wrist device was used to collect data for both datasets. It provides heart rate (HR), blood volume pulse (BVP), galvanic skin response (GSR), skin temperature (ST), time between heartbeats (IBI) and accelerometer data. To collect the laboratory data we used a standardized stress-inducing experiment as proposed by Dedovic et al. [2]. The main stressor was solving a mental arithmetic task under time and evaluation pressure[3]. The real-life data was gathered on ordinary days, when the subjects were wearing the wrist device and were keeping track of their stressful events.

Figure 1 presents the proposed method for stress detection in real-life. The method consists of three main ML components: a laboratory stress detector, an activity recognizer, and a context-based stress detector which provides the final output.

The laboratory stress detector is a ML classifier that distinguishes stressful vs. non-stressful events in 4-minute data windows with a 2-minute overlap. For each data window, features for stress detection are computed. From each physiological signal (BVP, HR ST and GSR), statistical and regression features are computed: mean, standard deviation, quartiles, quartile deviation, slope and intercept. Additional features to quantify the GSR response are computed with an algorithm for peak detection [8]. For the IBI signal, we use features obtained through heart-rate-variability analysis in the frequency and time domain. These features are fed into a classifier trained with the Random Forest ML algorithm, which was chosen experimentally.

The activity recognition (AR) classifier is a ML classifier that uses the accelerometer data to recognize the user's activity: sitting, walking, running, and cycling. It is based on our previous approach for AR [4]. The classifier outputs an activity label every 2 seconds. When aggregating these activities over the data window of 4 minutes, each activity is changed into an activity level (e.g., lying = 1, walking = 3, running = 5) and averaged over the window. The average activity level is passed as a feature to the context-based stress detector.

[1] Jožef Stefan Institute, Department of Intelligent Systems
[2] Jožef Stefan International Postgraduate School

[3] https://www.empatica.com/
[4] http://dis.ijs.si/thestest/

Figure 1. Method for stress detection in real life.

The context-based stress detector was developed to distinguish between genuine stress in real life and the many situations which induce a similar physiological arousal (e.g., exercise, eating, hot weather, etc.). As features, it uses the distribution of the last 10 outputs of the laboratory stress detector, the previous output of the context-based detector, and context features: whether there was any high-level activity in the last 20 minutes, the hour of the day, the type of the day – workday/weekend, etc. It classifies every 20 minutes as stressful or non-stressful. The context-based stress detector was trained with SVM, which was chosen experimentally.

3 Experiments

The evaluation of our method was performed on the real-life data. Because labeling stress is quite subjective [6] and it is almost impossible to strictly define starts and ends of stressful situations, we used a technique that splits the stream of real-life data into discrete events. Each event had a minimum length of one hour. If there was a stressful situation in the event (labeled by the user), the event's duration was extended to capture the stressful situation plus one hour before and after the situation. This allows a labeling lag of one hour. The 55 days of the real-life data was split into nearly 900 events, each lasting at least an hour.

Table 1 presents the confusion matrices for the event-based evaluation using leave-one-subject-out (LOSO) cross-validation. On the left are the classification results without context (based only on the predictions of the laboratory stress detector) and on the right are the results for the context-based stress detector. The accuracy achieved by the context-based stress detector (for distinguishing stressful vs. non-stressful events) is 92%, which is for 16 percentage points better than the no-context classifier. Additionally, Figure 1. depicts the output of the context-based stress detector for the real-life dataset. On the x-axis is the day, on the y-axis is the hour of the day, the black stripes label which subject the data belongs to, and the colored squares correspond to the false positive (FP), false negative (FN), true positive (TP) and true negative events (TN). From the figure it can be seen that subject 1 (S1) has many FN events, and subject 2 (S3) has more FP events compared to the rest of the subjects.

	No Context		With context	
	0	1	0	1
0	638	175	790	23
1	44	70	51	63
Recall	78%	61%	97%	55%
Prec.	94%	29%	94%	73%
F1	85%	39%	96%	63%
Acc.	76%		92%	

Table 1. Confusion matrices for event-based evaluation. Context vs. no-context.

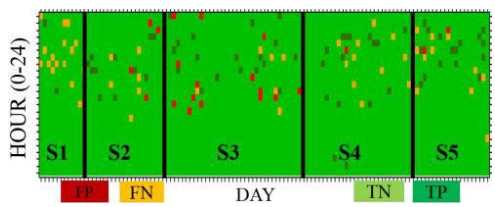


Figure 2. Context-based output with LOSO evaluation.

4 Discussion and conclusion

We developed a method that can continuously detect stress in real life. By introducing a context-based classifier we provided more information about real-life circumstances and the user, which improved the detection performance.

While still leaving room for improvement, the results are encouraging for such a challenging problem. For now, the context-based stress detector receives information from the laboratory detector and the activity recognizer. Additional context information can be provided from other components that recognize events which induce similar physiological arousal to a stress event (e.g., exercise, eating, hot weather etc.). Because stress is perceived differently, we plan to implement personalization to allow to the general model to adapt to new users. Figure 1 confirms the need for personalization where it shows that the distribution and the type of the classification errors (e.g., FP vs. FN) is subject-specific.

We are currently implementing the method as a real-time smartphone application. It will be demonstrated at the conference, where the participants will wear the wristband during stressful events, e.g., while giving a presentation. We will also integrate our method into an existing application that provides relaxation and lifestyle advice upon detected stress. It is intended for older workers and will be used in the European project Fit4Work [3].

REFERENCES

[1] Calculating the cost of work-related stress and psychosocial risks [Online]. Available: https://osha.europa.eu/en/tools-and publications/publications/literature_reviews/calculating-the-cost-of-work-related-stress-and-psychosocial-risks, [Accessed 15.10.2014].

[2] K. Dedovic, R. Renwick, N. K. Mahani, and V. Engert, "The Montreal Imaging Stress Task: using functional imaging to investigate the effects of perceiving and processing psychosocial stress in the human brain," Journal of Psychiatry and Neurosciencevol. 30, no. 5, pp. 319–325, 2005.

[3] Fit4Work Project, http://www.fit4work-aal.eu/

[4] H. Gjoreski et al. "Competitive Live Evaluation of Activity-recognition Systems," IEEE Pervasive Computing, vol:14, 2015.

[5] J. A. Healey and R. W. Picard, "Detecting Stress During Real-World Driving Tasks Using Physiological Sensors," IEEE Trans. Intell. Transp. Syst., vol. 6, no. 2, pp. 156–166, 2005.

[6] J. Hernandez, R. R. Morris, and R. W. Picard, "Call Center Stress Recognition with Person-Specific Models with Person-Specific Models," pp. 125–134, 2011.

[7] K. Hovsepian, M. Absi, T. Kamarck, and M. Nakajima, "cStress : Towards a Gold Standard for Continuous Stress Assessment in the Mobile Environment," In the Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 493-504, 2015.

[8] G.K. Palshikar, "Simple Algorithms for Peak Detection in Time-Series". Ii the Proc. 1st Int. Conf. Advanced Data Analysis, Business Analytics and Intelligence, June 2009.

[9] J. Ramos, J. Hong, and A. K. Dey, "Stress Recognition - A Step Outside the Lab," Proc. Int. Conf. Physiol. Comput. Syst., 2014.

[10] N. Sharma and T. Gedeon, "Modeling a stress signal," Appl. Soft Comput. J., vol. 14, no. PART A, pp. 53–61, 2014.

[11] S.C. Segerstrom, G.E. Miller, "Psychological stress and the human immune system: a meta-analytic study of 30 years of inquiry," Psychological Bulletin 130, 601, 2004.

[12] H. G. Ice, G. D. James, "Measuring Stress in Humans: A practical Guide for the field," Cambridge university press, 2007.

[13] S. Tyburski, et al., "Finding Significant Stress Episodes in a Discontinuous Time Series of Rapidly Varying Mobile Sensor Data", In Proc. of the 2016 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, 4489-4501.

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

doi:10.3233/978-1-61499-672-9-1805

An Intelligent System for Aggression De-Escalation Training

Tibor Bosse[1] and **Charlotte Gerritsen**[2] and **Jeroen de Man**[1]

Abstract. Artificial Intelligence techniques are increasingly being used to develop smart training applications for professionals in various domains. This paper presents an intelligent training system that enables professionals in the public domain to practice their aggression de-escalation skills. The system is one of the main products of the STRESS project, an interdisciplinary research project involving partners from academia, industry and society. The system makes use of a variety of AI-related techniques, including simulation, virtual agents, sensor fusion, model-based analysis and adaptive support. A preliminary evaluation of the system has been conducted with two groups of potential end users, namely tram conductors and police academy students.

1 INTRODUCTION

"A train conductor was assaulted at Nuneaton railway station after asking to see a man's ticket. The 48-year-old victim was working on a service from Crewe to London when he asked to see a passenger's ticket before he boarded the train in Nuneaton. The man became abusive and started to push the conductor before leaving the station." [28]

Although it is just one example, this incident illustrates the vulnerability of employees in the public sector to aggressive behaviour of people like customers, patients, travellers or other citizens. Aggressive behaviour against public service workers (e.g. police officers, ambulance personnel, public transport employees) is an ongoing concern in many countries [14, 21]. According to a national safety investigation in the Netherlands in 2011, almost 60% of the employees is confronted with unwanted behaviour on a daily basis [1]. Most incidents of aggression are of a verbal nature (e.g., insulting, swearing, intimidating), but in about 10% of the cases the conflicts escalate into physical aggression (e.g., threatening, abusing, robbing).

To better prepare them for these incidents, professionals in the public domain often receive dedicated resilience training. Such training is typically performed in a group setting based on role-play, where employees learn to communicate with aggressive clients in a de-escalating manner. Although this form of training has shown to be successful, it is quite expensive with respect to both money and time. Furthermore, the training is not always easy to control or repeat systematically.

As a complementary approach, the aim of the STRESS project [26] was to develop a simulation-based training system for aggression de-escalation. This is in line with a number of recent initiatives that show promising results regarding the possibility to train social and communicative skills based on simulated environments involving virtual humans [2, 12, 13, 17, 18, 19, 22]. The main idea of the current system is that employees in the public domain can practice their aggression de-escalation skills by engaging in conversations with aggressive virtual characters. By designing the scenarios in such a way that the characters calm down if they are being approached correctly, but become more aggressive if they are being treated inappropriately, trainees will receive immediate feedback on their performance. Meanwhile, they are monitored by intelligent software that observes and analyses their behaviour and physiological state (e.g., heart rate, skin conductance, brain activity) and provides tailored feedback. Feedback consists of two categories, namely hints and prompts on the one hand, and run-time modifications in the scenarios on the other hand. By using such a system, employees have the ability to practice their aggression de-escalation skills in a cost-effective, personalised and systematic manner.

The purpose of this paper is to describe the overall architecture of the system, some details of the various components of which it consists, and the preliminary results of evaluation studies with potential end users. Many of these elements have been published in previous papers [4-11], but this is the first time in which all of them are combined into our coherent description. Hence, the main contribution of the paper is in explaining how the various parts of the STRESS project come together in a practical application.

The remainder of the article is structured as follows. In Section 2, some background knowledge is presented about aggressive behaviour and the prescribed techniques to de-escalate aggression. Section 3 describes the architecture of the overall training system, as well as some details about its separate components. Section 4 summarises two case studies that have been conducted to test the system: one in the domain of public transport, and one in the domain of law enforcement. Finally, Section 5 concludes the paper with a discussion.

2 AGGRESSION DE-ESCALATION

To design an effective training tool, a first question to be asked is what should be the learning goals of the system. For the current context, these learning goals are similar to the ones used in real world aggression de-escalation training, and are related to the development of *emotional intelligence*: employees should learn to recognize the emotional state of the (virtual) conversation partner, and choose the communication style that suits this emotional state.

More specifically, when it comes to aggressive behaviour, it is important that employees learn to recognize the nature of the aggression. Here, two main categories can be distinguished: aggression can be either *emotional* (or *reactive*) or *instrumental* (or *proactive*) [15]. In case of emotional aggression, the aggressive

[1] Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands, email: t.bosse@vu.nl, jeroendeman@gmail.com
[2] Netherlands Institute for the Study of Crime and Law Enforcement, Amsterdam, The Netherlands, email: cgerritsen@nscr.nl

behaviour typically is caused by an angry reaction to a negative event that frustrates a person's desires [3]. Such a person is likely to be angry with respect to whatever stopped him from achieving his goal. An example in the public transport domain is a traveller getting angry because the tram is late while he has to attend an important meeting. When dealing with an emotional aggressor, *supportive* behaviour from the de-escalator is required, for example by ignoring the conflict-seeking behaviour, calmly making contact with the aggressor, actively listening to what he has to say, showing empathy, and suggesting solutions to his problems.

In contrast, in case of instrumental aggression, the aggressive behaviour is only used 'instrumentally', to achieve a certain goal. Such behaviour is not a direct response to a negative event and is less strongly related to heavy emotions. A well-known example in the domain of public transport involves someone who wants to travel without paying for his ticket. This type of aggression often starts with an attempt to persuade the conversation partner, e.g. "Oh, I forgot my wallet, can I just come along for two stops?". Often, in case the employee does not give the aggressor what he wants, the aggressive behaviour will reveal itself through more threatening remarks like "I'll be back tomorrow with my friends". To de-escalate instrumental aggression, a *directive* response is assumed to be most effective. It is necessary to show the aggressor that there is a limit to how far he can pursue his aggressive behaviour, and to make him aware of its consequences [10].

To conclude, the presented training environment is centred around two main learning goals, namely 1) *recognizing* the type of aggression of the conversation partner (i.e., emotional or instrumental), and 2) selecting the appropriate *communication style* towards the conversation partner (i.e., supportive or directive). To assess the type of aggression, employees need to carefully observe the verbal and non-verbal behaviour of the aggressive virtual character. In general, reactive aggressors will show more arousal (e.g., flushed face, emotional speech) than proactive aggressors. Also, the context should be taken into account (e.g., someone who just finds out that he lost his ticket will be more emotional that someone who knew this all along, and just tries to intimidate the tram driver to ride for free).

3 TRAINING SYSTEM

The main aim of the STRESS project is to develop an intelligent training system that is able to analyse the trainee's behaviour during confrontations with aggressive individuals, and provide appropriate feedback, enabling trainees to improve their performance. During the training, users will be placed in a virtual scenario in a particular domain (e.g., selling tram tickets, or issuing parking tickets), which involves one or more virtual agents that at some point in time start behaving aggressively (e.g., insulting the tram driver because he is late). The user's task is to de-escalate the aggressive behaviour of the virtual agents by applying the appropriate communication techniques. An important asset of the system is that it can adapt various aspects of the training (e.g., scenarios, difficulty level, feedback) at runtime on the basis of its estimation of the trainee's physiological state and performance.

3.1 System Overview

Figure 1 depicts the global architecture of the system [6]. The rounded rectangles denote components of the system, and the arrows denote information flows. The normal rectangles indicate clusters of components that have the same function (i.e. the *analysis* and *support* layer). In the training environment, the trainee will be engaged in a *virtual reality environment* shown on a computer screen (or possibly on a head-mounted display), while being monitored by an intelligent *training agent*. The virtual scenario is generated by a separate module within the agent, which contains knowledge about relevant scenarios in a particular domain. The trainee observes the events that happen in the scenario, and has to act in the scenario this by selecting the most appropriate action (currently this is simply implemented by means of a multiple choice menu). During training, the user is connected to (non-intrusive) devices that measure physiological states related to arousal and stress; in particular heart rate, skin conductance and electroencephalogram (EEG) signals. The data measured by these devices are then used by the agent as input for a computational model that integrates them at runtime, to assess the trainee's levels of stress and (negative) emotions (the *affective model*). This assessment of the trainee's affective state is combined with information about the status of the task (e.g., the actions performed by the trainee), and used by another computational model (the *decision making model*) to assess whether (and why) the trainee made certain mistakes. The outputs of both models are analyses of the trainee's emotional state (e.g., how much stress does (s)he experience?) and decision making behaviour (e.g., are any mistakes made?), respectively. This information is used for two purposes: by the *scenario development module*, to modify the running scenario (e.g., to repeat a certain scenario that is considered difficult), and by the *feedback determination module*, to provide feedback to the trainee (e.g., advice to change the conversation style).

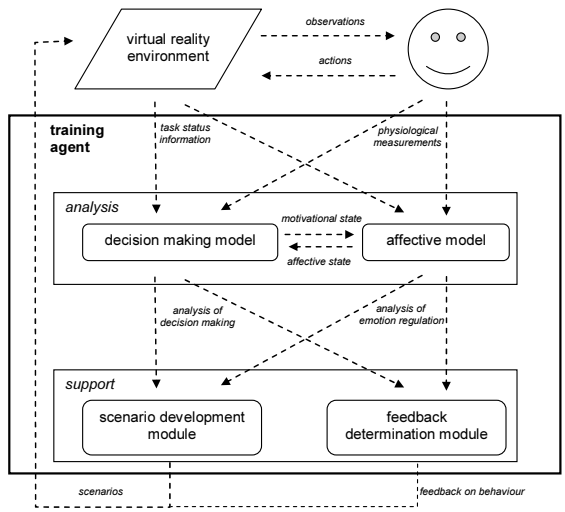


Figure 1. Global architecture of the training system

3.2 Virtual Reality Environment

The virtual reality environment is based on the InterACT software [30], developed by the company IC3D Media [25]. InterACT is a software platform that has been specifically designed for simulation-based training of interpersonal skills. Unlike most existing software, it focuses on smaller situations, with high realism and detailed interactions with virtual characters. True-to-life animations and photo-realistic characters are used to immerse the player in the game. An example screenshot of a training

scenario for the public transport domain is shown in Figure 2. In this example, the user plays the role of a tram conductor that has the task of calming down an aggressive virtual traveller.

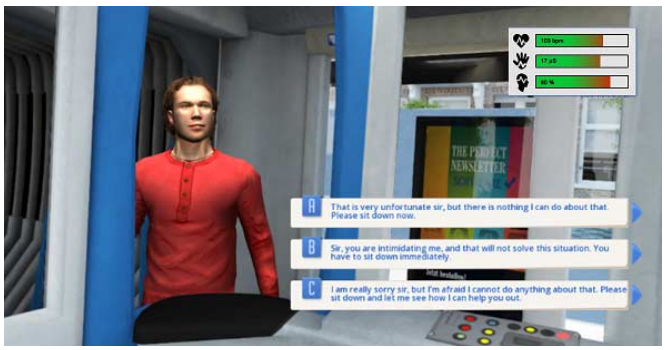


Figure 2. Screenshot of a training scenario for tram conductors

To enable users to engage in a conversation with an embodied conversational agent (ECA), a dialogue system based on conversation trees is used. The system assumes that a dialogue consists of a sequence of spoken sentences that follow a turn-taking protocol. That is, first the ECA says something (e.g. "I forgot my public transport card. You probably don't mind if I ride for free?"). After that, the user can respond, followed by a response from the ECA, and so on. In InterACT, these dialogues are represented by conversation trees, where vertices are either atomic ECA behaviours or decision nodes (enabling the user to determine a response), and the edges are transitions between nodes.

The atomic ECA behaviours consist of pre-generated fragments of speech, synchronised with facial expressions and possibly extended with gestures. Scenario developers can generate their own fragments using a motion sensing input device such as the Microsoft Kinect camera and a commercial software package FaceShift [29]. As the recorded fragments are independent from a particular avatar, they can be projected on arbitrary characters.

Each decision node is implemented as a multiple choice menu. Via such a menu, the user has the ability to choose between multiple sentences. Hence, the emphasis of the current system is on the verbal aspects of aggression de-escalation. In most of the scenarios, three options are available within every decision node. Typically, these options have been created in such a way that one of them is clearly *supportive*, another one is clearly *directive*, and the third option is neutral. Here, the supportive and directive option relate to the communication styles explained in Section 2. Figure 2 illustrates how these three options can be instantiated in terms of concrete sentences (where A=neutral, B=directive, C=supportive). The user's choice determines how the scenario continues, by triggering a corresponding branch in the tree.

Although this approach works well, there is a risk that the behaviour of the ECAs becomes predictable in the long term. For example, in the situation shown in Figure 2, choosing option B (the 'directive' option) will always result in the ECA becoming irritated, no matter how often the scenario is played, or what has happened before. To overcome this problem, in [11] an approach was put forward to endow the agent with *internal states* that are either set beforehand (e.g. whether the agent is a reactive or a pro-active aggressor) or are the result of earlier interactions (e.g. a state of anger that gradually increases during the scenario). This means that the agent is equipped with a cognitive model of aggression, replacing the direct connections between user choices and ECA responses. As shown in [11], the resulting conversations indeed provided more variation, and were perceived as less predictable.

3.3 Physiological Measurements

Simulation-based training can only be effective if the virtual scenarios trigger emotional responses that are comparable to the reactions people show to the same stimuli in real world scenarios. To investigate to what extent this is the case, in [4] an experiment was performed in which the impact of an aggressive virtual agent was compared with that of an aggressive human. By randomly distributing a group of 28 participants over two conditions (virtual and human) and measuring their physiological and subjective emotional state before and after an aggressive outburst of their conversation partner, the difference between virtual and human aggression was studied. The 'outburst' was realized by having the (virtual or human) conversation partner suddenly get extremely angry towards the participant, while shouting and accusing him or her of not paying attention. The results showed that both conditions induced a substantial stress response, but that the impact of the human aggression was stronger than that of the virtual aggression.

Part of these results are illustrated by Figure 3. This figure depicts the dynamics of skin conductance (also called electrodermal activity, EDA, one of the most common indicators for arousal) over time in microSiemens during the relevant part of the experiment, averaged over all participants in each condition. The horizontal axis denotes a period of 2 minutes, i.e., 1 minute before the start of the aggressive outburst and 1 minute after it. The vertical line indicates the moment the outburst started. More details about the study can be found in [4].

Although these results show that there is still a gap between the intensity of the stress response caused by a virtual character and by a real human, they are nevertheless promising, because they indicate that virtual characters can at least elicit responses that are comparable to the ones triggered in reality.

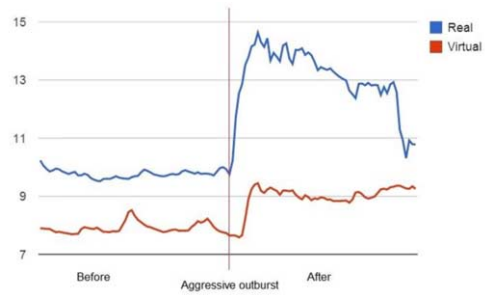


Figure 3. Dynamics of electrodermal activity over time

Inspired by these results, an interface was developed to connect the training system to two devices that measure physiological states related to arousal and stress. In particular, the Plux wireless biosensors toolkit [27] is used to measure heart rate and skin conductance, and the Myndplay Brainband [31] is used to measure EEG signals. Additionally, within InterACT a visualisation window was implemented in which the measurements of these three sensors can be displayed at run-time during training (see upper right corner of Figure 2). From top to bottom, this windows shows a user friendly interpretation of the trainee's current heart rate (in beats per minute), skin conductance (in microSiemens) and EEG signals (in terms of 'meditation value', one of the outputs of the Myndplay Brainband that is correlated with a state of

relaxation). This allows trainees to receive instant (bio)feedback on their physiological state while training, thereby helping them to stay calm during aggressive confrontations.

3.4 Analysis Layer

As explained in Section 3.1, the purpose of the analysis layer is to process data about the user (in particular: physiological data and task status information from the virtual environment), in order to draw high-level conclusions about the user's state. It consists of two sub-models, the affective model and the decision making model, which are described in the following sub-sections. Both models have been formalised using the LEADSTO language [9], which enables modellers to describe mental processes in terms of transitions between states that are expressed in terms of logical and/or numerical variables.

3.4.1 Affective Model

The affective model was inspired by the work of Gross [16], and is explained in detail in [6]. A high-level overview is shown in Fig. 4. The circles represent different states, which are all formalised in a numerical manner, in terms of a variable with a real value between *0* and *1*. In an actual application, real world data should be mapped to values in this interval. For instance, a very threatening stimulus in the training system (e.g., an aggressive virtual character) could be represented as a *world state* with value 0.9. Similarly, a moderately intensive feeling of fear (e.g., as measured by the physiological devices) could be represented as a *feeling* with value 0.5.

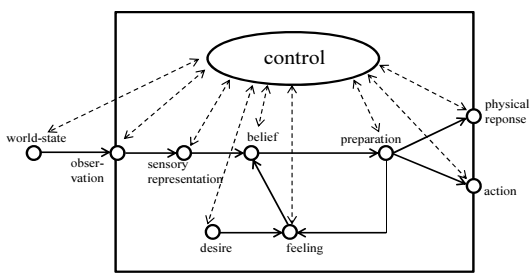


Figure 4. Overview of affective model

Arrows in Figure 4 denote the influence of one state on another state. The model that represents *generation* of emotions is depicted by using solid arrows. Additionally, *regulation* of emotions is represented by the *control* state. Each regular state has a positive effect on the control state (representing a monitoring process), but can in turn be suppressed by the control state (representing a regulation process), as indicated by the dashed arrows.

3.4.2 Decision Making Model

The decision making model is explained in detail in [10], and is shown graphically in Figure 5. The circles on the left denote observations made by the user, the circles on the right (communicative) actions, and the remaining circles internal states.

Roughly, the dynamics of the model can be split into three sub-processes. First, as shown in the lower part of the figure, the emotional state of the user is updated based on the observed (verbal and non-verbal) behaviour of the conversation partner, and has in turn an impact on his or her own non-verbal behaviour. Next, as shown in the upper left part of Figure 5, there is a sub-process related to the evaluation of (both the nature and the intensity of) the conversation partner's emotional state. More specifically, this boils down to deciding whether we are dealing with reactive or proactive aggression (or no aggression). Finally, as shown in the upper right part of the figure, the evaluation of the conversation partner's emotional state serves as input for a decision about which 'de-escalation approach' to select. For this, domain-specific knowledge is used about which approach works best in which situation (as explained in Section 2).

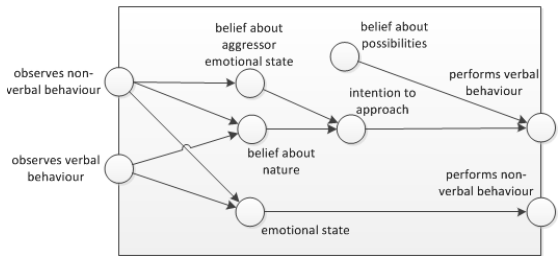


Figure 5. Overview of decision making model

To conclude, the affective model and decision making model enable the system to draw conclusions about the user's emotional state and decisions (and errors) made, respectively. Preliminary evaluations of the models are discussed in [6] and [10]. In the next section we explain how the output of the models can be used to provide dedicated support to increase training effectiveness.

3.5 Support Layer

Like the analysis layer, the support layer also consists of two sub-models, namely the scenario development module and the feedback determination module. Both modules are described below.

3.5.1 Scenario Development Module

The main purpose of the scenario development module is to generate interesting training scenarios that fit to the learning goals of the trainee. In particular, the concept of adaptive training (or scaffolding) is used, where the difficulty level of the scenarios adapts to the performance of the trainee. To this end, based on the learning goals identified in Section 2, a score was introduced to keep track of how well the goals were achieved. This score was calculated based on the output of the analysis layer. Next, a number of difficulty levels were established, as well as a mechanism to navigate up and down between these levels based on the user's score. This mechanism is visualised in Figure 6. As can be seen, a separate score is maintained for cases of emotional (or reactive) aggression as well as for cases of functional (or pro-active) aggression. The main idea is that the user's score for a particular type of aggression needs to be sufficiently high to reach a higher level. However, after losing two points, the user falls back a level. In the first part of the training (level 1-3), the type of aggression is already given. Instead, in the second part of the training (level 4-6) the trainee needs to identify the type of aggression by him- or herself. Levels are traversed per aggression type separately, with one exception: after the first part of the training (i.e., level 1-3), the trainee needs to have sufficient knowledge of both types of aggression before (s)he can continue.

Details of this module are presented in [8]. An initial evaluation reported in that paper demonstrated that the system successfully

adapted its difficulty level to the user's performance, and that users were positive about the effect of this adaptation mechanism.

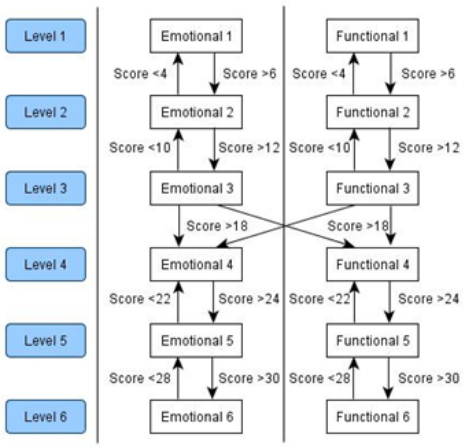


Figure 6. Transitions between difficulty levels

3.5.2 Feedback Determination Module

The feedback determination module is described in detail in [11]. This model uses the output of the analysis layer to generate appropriate feedback on the user's performance in terms of after-session hints. Essentially it checks whether the situation was successfully de-escalated or not, and in the latter case, it analyses what the cause of this unsuccessful de-escalation was. In this analysis, several types of mistakes are distinguished such as 1) the user failed to judge the type of aggression correctly (i.e. reactive or proactive), 2) the user failed to apply the appropriate communication style (supportive or directive), and 3) the user failed to control his or her own emotional state. The decision tree that is used by the module is shown in Figure 7.

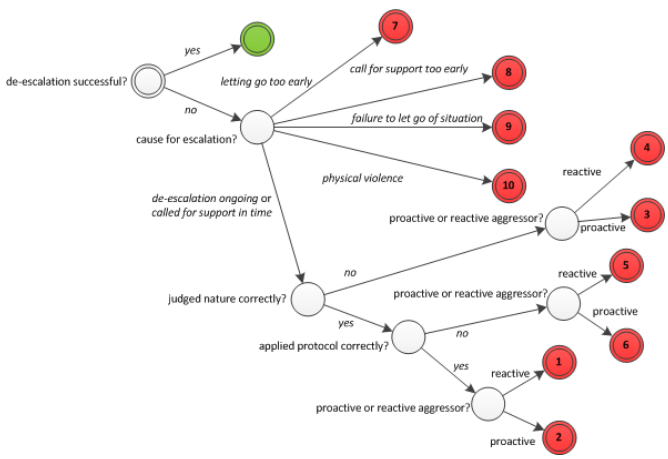


Figure 7. Overview of decision making model

Here, a green end state indicates successful de-escalation, whereas a red end state indicates unsuccessful de-escalation. Based on the specific end state a corresponding feedback message is generated, represented by the numbers in the figure. As an example, in case a scenario is classified as category (6), the following feedback is presented:

> *6. User applies wrong approach towards a proactive aggressor.*
> "You correctly judged the nature of the aggression, but you used the wrong verbal approach. A proactive aggressor should always be approached in a directive manner.

To test the module, a specific scenario has been worked out in the context of a man who has no cash money to pay for a tram ticket. A group of users have extensively played the scenario under varying the parameter settings. These user tests pointed out that the module indeed offered the desired support at the appropriate times (see [11] for more details).

4 CASE STUDIES

As explained in the previous section, most components of the system were tested separately based on user studies in a laboratory setting. As a follow up, additional evaluation studies have been performed with potential end users from two domains of interest: public transport and law enforcement. In these experiments, the focus was on testing these users' experience with respect to the virtual reality environment (and the underlying dialogue system); hence, the training agent was disabled during these tests. The two experiments are briefly described in the following sub-sections.

4.1 Public Transport

For this evaluation study (see [7]), a number of scenarios have been developed, in collaboration with (and approved by) domain experts of GVB, the public transport company in Amsterdam. All scenarios are perceived from the perspective of a tram conductor. In total, they address 9 different situations in which a conflict may arise, such as 'traveler is not allowed to take hot coffee on board' and 'tram arrives 10 minutes late'. Moreover, for each scenario, different variants have been written: some in which the virtual character shows emotional aggression, and some in which it shows instrumental aggression. In total, there were 40 scenarios.

Twenty-four people participated in the experiment (13 male and 11 female), all of which were employees of GVB. Their average age was 45,4 (σ = 12.0). The experiment was executed in a computer room at GVB. Participants had to play all 40 scenarios during 4 different sessions distributed over 4 weeks (i.e., 10 scenarios per session). After the last session, the participants were asked to fill out a usability questionnaire, consisting of 20 statements about which the participants had to express their opinion on a 5-point Likert scale. The questionnaire was inspired by Witmer and Singer [24], and included statements about issues such as user experience, presence, and perceived effectiveness. In the end, the statements were grouped into 4 categories, namely *content*, *interaction*, *engagement*, and *effect*, to obtain an average score on these aspects. The *content* category contained statements about the perceived realism of the scenarios and the characters (e.g., 'the scenarios were representative for real world situations'). The *interaction* category contained statements about how natural it was to interact with the characters (e.g., 'I felt that my answers had an influence in the behaviour of the virtual characters'). The *emotional* category addressed the perceived sense of presence of the participants (e.g., 'during training I felt engaged in the scenarios'). Finally, the *effect* category contained statements asking the participants for their opinion about the effectiveness of the training (e.g., 'I think this type of training is a useful addition to real world training').

The aggregated results are summarised in Figure 8 (on a scale from -2 to 2). As shown there, participants were generally positive about the content of the scenarios, the interaction possibilities of the system, and (in particular) about its perceived learning effect,

but were less enthusiastic about the system's emotional impact. More details can be found in [7].

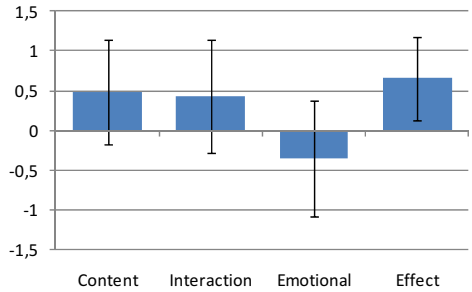


Figure 8. User experience results for the Public Transport case

4.2 Law Enforcement

To investigate how stable these results are across different application domains, a second evaluation study was conducted (see [5]), in the domain of law enforcement. This experiment was executed in collaboration with the Dutch Police Academy, and focused on the module 'Noodhulp' (Emergency Assistance), which is part of their education program. As part of this module, students have to learn to correctly handle the so-called 'Door Scene'. This is a situation in which a police officer has just been informed about an incoming emergency call. For the current experiment, we focused on the domain of domestic violence (e.g., a call from a crying woman who claims that her boyfriend is abusing her). The scenario starts at the moment that the police officer (together with his or her partner) arrives at the address in question, and rings at the door.

The setup of this experiment was very similar to the previous one, with some minor differences: instead of 40 scenarios, only 4 (slightly longer) scenarios were used. Moreover, these scenarios were not played during 4 different sessions, but during one single session. The experiment was executed in a computer room at the Police Academy. Also the demographics of the participants were different: in total, 41 Police Academy students participated in the experiment (31 male and 10 female), and their average age was 27.1 (σ = 6.5). The questionnaire consisted of the same 20 statements as the previous one; however, instead of a 5-point Likert scale, a 7-point scale was used.

The aggregated results of this study are summarised in Figure 9 (on a scale from -3 to 3). As shown, the outcomes are similar to those for the public transport domain, with the main difference that the participants were more positive about the system's interaction possibilities, but believed a bit less in the effectiveness of the tool. More details of this study can be found in [5].

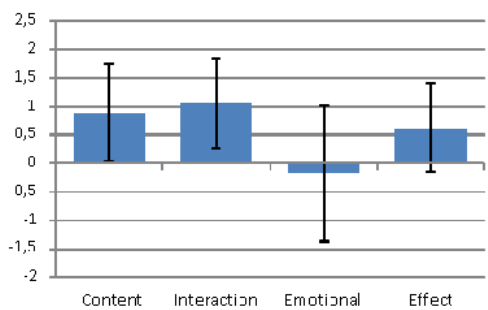


Figure 9. User experience results for the Law Enforcement case

5 DISCUSSION

The current paper introduced a prototype of an intelligent simulation-based training system that enables professionals in the public domain to practice their verbal aggression de-escalation skills during face-to-face conversations. The system has been designed in a modular fashion, and integrates various AI-related techniques, including simulation, virtual agents, sensor fusion, model-based analysis and adaptive support. The various modules have been tested separately in a lab setting. Additionally, case studies in two real world domains have been used to obtain feedback on the virtual reality environment from potential end users.

The presented system has similarities with several recent approaches to train social skills through conversations with virtual humans. These projects have addressed a variety of tasks in different domains, including job interviews [2], police interviews [13], leadership training for naval officers [17], medical consultations [18], negotiation exercises [12, 19], and manager-employee conversations [22]. The current system differs from these systems in the sense that it focuses on a domain (aggression de-escalation) in which the stimuli from the virtual environment are mainly negative. As a result, more effort was put into creating visually and behaviourally believable characters, and into measuring the users' physiological response to the behaviour of these characters. Additionally, the system was enhanced with an intelligent 'training agent' that gives adaptive personalised feedback based on the user's state and behaviour (cf. [20, 23]).

The results from the case studies indicate that with respect to user satisfaction, participants were generally positive about the content of the virtual scenarios, the mechanisms to interact with the characters, and the potential of the system as a learning tool. Nevertheless, also a number of points for improvement were identified, which mainly have to do with the emotional aspect of the system: for several participants, the perceived sense of presence was limited because they did not 'feel' the emotion in the virtual conversation partner. One interesting way to improve this situation is to combine the scenarios with haptic feedback. Another possible extension would be to use a head-mounted display instead a flat video screen. Our expectation is that such extensions will lead to a more engaging, and therefore more effective training tool.

ACKNOWLEDGEMENTS

This research was supported by funding from the National Initiative Brain and Cognition, coordinated by the Netherlands Organisation for Scientific Research (NWO), under grant agreement No. 056-25-013.

REFERENCES

[1] M. Abraham, S. Flight, and W. Roorda. Agressie en geweld tegen werknemers met een publieke taak. Research for the program 'Veilige Publieke Taak 2007 - 2009 – 2011'. Amsterdam: DSP.

[2] A. Ben Youssef, M. Chollet, H. Jones, N. Sabouret, C. Pelachaud, and M. Ochs. Towards a Socially Adaptive Virtual Agent. In: Intelligent Virtual Agents, Springer LNCS, vol. 9238, pp. 3-16, 2015.

[3] L. Berkowitz Whatever Happened to the Frustration-Aggression Hypothesis? American Behavioral Scientist, 21, 691-708, 1978.

[4] R.A.M. Blankendaal, T. Bosse, C. Gerritsen, T. de Jong, and J. de Man. Are Aggressive Agents as Scary as Aggressive Humans? In: Proceedings of AAMAS'15. ACM Press, 2015, pp. 553-561.

[5] T. Bosse, and C. Gerritsen. Towards Serious Gaming for Communication Training - A Pilot Study with Police Academy Students. In: Proceedings of Intetain 2016. Springer Verlag, 2016.
[6] T. Bosse, C. Gerritsen, and J. de Man. Agent-Based Simulation as a Tool for the Design of a Virtual Training Environment. In: Proceedings of the 14th International Conference on Intelligent Agent Technology, IAT'14. IEEE Computer Society Press, pp. 40-47, 2014.
[7] T. Bosse, C. Gerritsen, and J. de Man. Evaluation of a Virtual Training Environment for Aggression De-escalation. In: Proceedings of Game-On 2015. Eurosis, 2015.
[8] T. Bosse, C. Gerritsen, J. de Man, and S. Tolmeijer. Adaptive Training for Aggression De-escalation. In: C.J. Headleand et al. (Eds.), Proc. of ALIA'14. Springer CCIS, vol. 519, 2015, pp. 80-93.
[9] T. Bosse, C.M. Jonker, L. van der Meij, and J. Treur. A Language and Environment for Analysis of Dynamics by Simulation. Int. J. of AI Tools, volume 16, issue 3, 2007, pp. 435-464, 2007.
[10] T. Bosse, and S. Provoost. Towards Aggression De-escalation Training with Virtual Agents: A Computational Model. In: Proceedings of HCI'14. Springer Verlag, pp. 375-387, 2014.
[11] T. Bosse, and S. Provoost. Integrating Conversation Trees and Cognitive Models within an ECA for Aggression De-escalation Training. In: Proceedings of PRIMA'15. Springer Verlag, Lecture Notes in Artificial Intelligence, 2015.
[12] J. Broekens, M. Harbers, W.P. Brinkman, C.M. Jonker, K. van den Bosch, and J.J. Meyer. Virtual reality negotiation training increases negotiation knowledge and skill. In: Intelligent Virtual Agents, Springer LNCS, vol. 7502, pp. 218-230, 2012.
[13] M. Bruijnes, J.M. Linssen, H.J.A. op den Akker, M. Theune, S. Wapperom, C. Broekema, and D.K.J. Heylen. Social Behaviour in Police Interviews: Relating Data to Theories. In: Conflict and Multimodal Communication, Springer Verlag, pp. 317-347, 2015.
[14] J. DelBel. De-escalating workplace aggression. Nurning Management (USA) 34, 31-34, 2003.
[15] K.A. Dodge. The structure and function of reactive and proactive aggression. In D. Pepler and H. Rubin, (eds.), The development and treatment of childhood aggression (pp. 201-218). Hillsdale, NJ: Erlbaum, 1990.
[16] J.J. Gross. Emotion regulation in adulthood: timing is everything. Current directions in psychological science, vol. 10, issue 6, pp. 214-219, 2001.
[17] M. Hays, J. Campbell, M. Trimmer, J. Poore, A. Webb, C. Stark, and T. King,. "Can Role-Play with Virtual Humans Teach Interpersonal Skills?. In Interservice/Industry Training, Simulation and Education Conference (I/ITSEC), 2012.
[18] J. Jeuring, F. Grosfeld, B. Heeren, M. Hulsbergen, R. Ijntema, V. Jonker, N. Mastenbroek, M. van der Smagt, F. Wijmans, M. Wolters, and H. van Zeijts. Communicate! – A Serious Game for Communication Skills. In: G. Conole et al. (Eds.): EC-TEL 2015, Springer LNCS, vol. 9307, pp. 513-517, 2015.
[19] J. Kim, R.W. Hill, P. Durlach, H.C. Lane, E. Forbell, C. Core, S. Marsella, D. Pynadeth, and J. Hart. BiLAT: A game-based environment for practicing negotiation in a cultural context. Int. J. of AI in Education, vol. 19, issue 3, pp. 289-308, 2009.
[20] M. Peeters. Personalized Educational Games. Developing Agent-Supported Scenario-Based Training. PhD Thesis, Utrecht University, 2014.
[21] M.J. Smith and R.V. Clarke. Crime and public transport. In R. Clarke (Ed.), Crime and Justice: A Review of Research (pp. 169-234). Chicago, IL: The University of Chicago Press, 2000.
[22] F. Vaassen and J. Wauters. deLearyous: Training interpersonal communication skills using unconstrained text input. In: Proceedings of ECGBL. pp. 505–513, 2012.
[23] J. Westra, H. van Hasselt, F. Dignum, and V. Dignum. Adaptive Serious Games Using Agent Organizations. In: F. Dignum et al. (eds.), Agents for Games and Simulations, Springer LNCS, vol. 5920, pp. 206-220, 2009.
[24] B.G. Witmer. and M.J. Singer. Measuring presence in virtual environments: a presence questionnaire. Presence: Teleoperators and Virtual Environments 7, pp. 225-240, 1998.
[25] http://ic3dmedia.com/.
[26] http://stress.few.vu.nl/.
[27] http://www.biosignalsplux.com/.
[28] http://www.coventrytelegraph.net/news/local-news/conductor-assaulted-platform-railway-station-10757452.
[29] http://www.faceshift.com/.
[30] http://www.interact-training.nl/.
[31] http://www.myndplay.com/.

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

doi:10.3233/978-1-61499-672-9-1812

A Practical Approach to Fuse Shape and Appearance Information in a Gaussian Facial Action Estimation Framework

Teena Hassan[1] and **Dominik Seuss**[1] and **Johannes Wollenberg**[1] and **Jens Garbas**[1] and **Ute Schmid**[2]

Abstract. In many domains of computer vision, such as medical imaging and facial image analysis, it is necessary to combine shape (geometric) and appearance (texture) information. In this paper, we describe a method for combining geometric and texture-based evidence for facial actions within a Kalman filter framework. The geometric evidence is provided by a face alignment method. The texture-based evidence is provided by a set of Support Vector Machines (SVM) for various Action Units (AU). The proposed method is a practical solution to the problem of fusing categorical probabilities within a Kalman filter based state estimation framework. A first performance evaluation on upper face AUs demonstrates the practical applicability of the proposed fusion method. The method is applicable to arbitrary imaging domains, apart from facial action estimation.

1 INTRODUCTION

Two important types of information commonly used in computer vision are shape and appearance. For example, in medical imaging, shape and appearance information have been used for automatic classification of cells [18]. In this paper, we focus on combining both sources of information for facial expression analysis. Facial expression is accompanied by deformation of the shape of facial features and by changes in facial texture. Changes in shape of facial features–such as eyes, nose, eyebrows, or lips–constitute geometric information. Changes in facial appearance–such as wrinkles, furrows, edges, or bulges–constitute texture information.

The emergence of robust face detection algorithms [31] in early 2000s accelerated research on automatic analysis of faces recorded in images and videos. Automatic analysis of facial expressions is one of the research fields that received increased attention since then [27, 35, 10]. Research in this field is pursued mainly in two directions [35]: One of the directions focuses on an objective analysis of basic facial movements based on the Facial Action Coding System (FACS) [9]. The other focuses on detecting a set of prototypical facial expressions of emotions.

Since more than ten years, the Intelligent Systems group at Fraunhofer IIS has been developing the Sophisticated High-speed Object Recognition Engine SHORE™[3], which is a general framework for various object detection tasks in images and videos, with a focus on face detection and analysis [25]. SHORE™ detects and tracks faces in real-time [20], estimates age and gender, and identifies four basic expressions of emotions, namely happiness, sadness, anger and surprise.

In collaboration with the GfK Verein (a think tank for market research and majority shareholder of Europe's largest market research company GfK SE), the software EMOScan was developed by customising and extending SHORE™ for valence detection [12]. EMOScan is currently deployed in nine European, three Asian and two American countries. It is used in online and offline studies for analysing viewers' facial responses to advertisements. Currently, SHORE™ is also being applied to several other application areas of automatic facial expression analysis, such as affective computing, pain detection and autism research.

While the focus of SHORE™ was primarily on emotion detection, our current research focuses on FACS-based analysis, where image sequences are processed for detecting facial motion and coding it in terms of FACS Action Units (AU). This approach enables detection of more subtle facial expressions. The detected AUs and their intensities could be used in a subsequent processing step for a detailed analysis of human affective behaviour. This is planned as an extension of SHORE™ with a similar deployment strategy.

In the next section, previous research on the fusion of geometric and texture information in the domain of facial expression analysis is reviewed. Afterwards, an overview of the main components of our AU estimation framework is provided. This is followed by a detailed description of the proposed fusion approach within a Gaussian state estimation framework. We present a performance evaluation of the fusion approach based on a dataset collected in a private study conducted by GfK Verein. Finally, we conclude with an outlook to future work.

2 RELATED WORK

Recently, fusion of shape (geometric) and appearance (texture) features has shown to give promising results in the field of AU detection and emotion recognition. Geometric features are usually computed using the location of facial landmarks defined according to a deformable face model. Typical approaches for landmark localization include Active Appearance Models (AAM) [5] and Constrained Local Model Fitting (CLM) [26]. Texture features encode visual texture information using, for example, histograms of oriented gradients (HOG) [6], histograms of local binary patterns (LBP) [13] or histograms of local Gabor binary patterns (LGBP) [2].

When it comes to fusion of geometric and texture features, machine learning approaches for facial expression analysis have em-

[1] Fraunhofer Institute for Integrated Circuits IIS, Germany, email: teena.hassan@iis.fraunhofer.de
[2] Faculty of Information Systems and Applied Computer Science, Otto-Friedrich-Universität Bamberg, Germany
[3] http://www.iis.fraunhofer.de/shore

ployed different strategies, namely, feature-level fusion, decision-level fusion, and fusion based on multiple kernel learning. Feature-level fusion is usually done by concatenation of all features into one large feature vector, which is then passed to a machine learning algorithm [3, 34, 33, 23, 36]. Decision-level fusion usually trains separate classifiers with geometric or texture features, and later combines their scores. For example, [17] uses a variant of artificial neural networks, and [16] uses an SVM to fuse scores from classifiers. Multiple kernel learning based fusion approaches use separate kernels for geometric and texture features. In [22], a multi-kernel SVM is used for feature fusion. A Gaussian kernel is used for geometric and gradient-based texture features, and an intersection kernel is used for higher-dimensional Gabor-based texture features. A multi-kernel SVM is also used in [28] for feature fusion, where kernels of the same type are applied to geometric and texture features.

Fusion of geometric and texture information can also be viewed as a sensor fusion problem. Traditionally, state estimation methods are used to perform sensor fusion, for example, in applications like robot navigation and satellite tracking. State estimation methods fuse a stochastic model of the temporal dynamics of the state of a system, and the stochastically distributed measurements provided by multiple sources. Dynamic state estimation methods such as Kalman filtering, particle filtering, Hidden Markov Models (HMM), and Dynamic Bayesian Networks (DBN) have been used for facial expression analysis [29, 8, 14, 37], in order to estimate intensities of AUs [8, 14, 7] and facial expressions of emotions [8, 37], as well as to infer the temporal phases of AUs [29]–such as onset, apex and offset.

AU intensity estimation is a continuous state-space estimation problem. Therefore, Kalman and particle filtering based approaches–such as [8, 14]–have been applied. These methods commonly use deformable face models that encode the semantics of the different facial expressions to be estimated, and use physics-based models to describe the temporal dynamics of facial motion. The parameters of pre-defined temporal models are learned from annotated data. Geometric or texture features constitute the measurements. These are fused by defining a measurement model that provides the likelihood of observing the geometric or texture features, given the current state estimate. The advantages of such approaches are their lower data requirements, and their simple and intuitive way of combining semantic, spatial and temporal aspects of facial motion.

However, the performance of these state estimation methods depends on the accuracy of the models and the correctness of the stochastic assumptions. For example, precise modelling of all possible variations in the shape, appearance and dynamics of facial expressions is not possible. To overcome these limitations, outputs from data-driven machine learning approaches could be used to improve the discriminative power of state estimation based approaches [24, 29, 30, 19]. The integeration of probability outputs from SVMs in an HMM-based discrete state estimation framework has been explored in application domains such as speech recognition [19] and recognition of temporal phases of AUs [29, 30]. However, their integration in a continuous, Gaussian-distributed state estimation framework appears to be unexplored.

The use of a state estimation method allows the possibility of filtering noisy measurements and enables the use of a dynamic model that approximates the mechanics of facial muscles. It provides a temporal smoothing effect on the output estimates that is usually lacking in machine learning approaches. Temporally smooth estimates improve the chances of finding good thresholds for discretisation, if needed. State estimation frameworks also provide an inherent mechanism to integrate information from multiple modalities or sources. In the light of these arguments, we use a state estimation based approach for estimating continuous-valued AU intensities.

We use anatomically-inspired models to capture the relevant spatio-temporal aspects of facial motion, such as the facial deformations resulting from the motion and the biomechanics of facial muscles that produce the motion. Since the state space is high dimensional, a Kalman filter based framework is used. To improve the discriminative capacity of the system, we use SVMs that are trained on texture features to discriminate between two or more AU classes. The Kalman filter assumes the state and measurements to be Gaussian-distributed. This gives rise to the problem of integrating categorical probabilities from SVMs into a Gaussian-distributed continuous state estimation framework. In this paper, we propose a practical approach to solve this problem. The applicability of the proposed approach is demonstrated within our AU estimation framework.

3 SYSTEM OVERVIEW

Estimating action unit intensities under real conditions, without the need for explicitly adapting the system to the monitored person, is very challenging. Our system tackles variations in lighting, head pose and interpersonal facial shape with the help of robust face detection, face normalisation and an online face calibration. The flowchart of our system is shown in Figure 1.

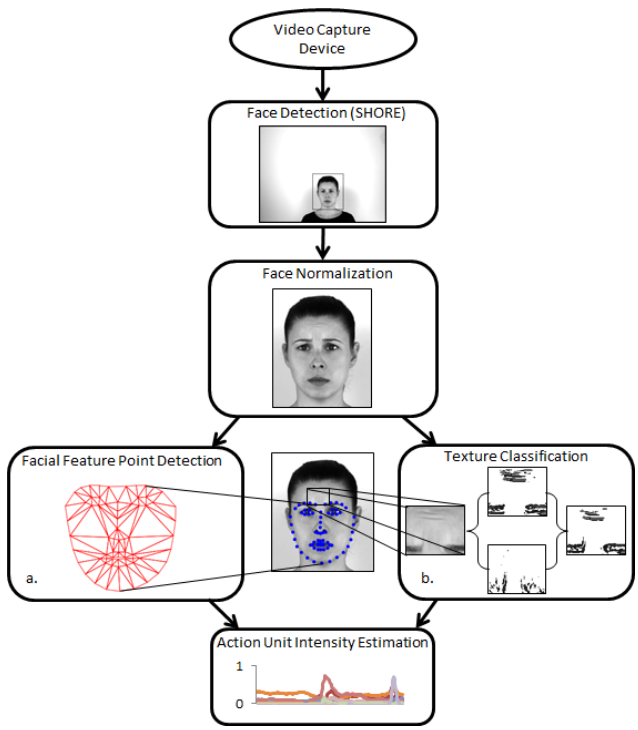


Figure 1. Flow chart of the system. The extracted face mesh is shown in (a). Exemplary texture features extracted from the forehead region using vertical and horizontal edge filters are shown in (b).

- **Face Detection:** The face detection unit processes the image frames provided by a video capture device. It uses SHORE™ [20] to locate the person's face in the image and obtain the region of the face. In addition, SHORE™ provides the location of eyes, nose and mouth corners, in case a face could be found. If more than one

face is present, the face detector selects the most prominent face on the basis of the face size in the image.

- **Face Normalization:** The face normalization unit rotates and scales the person's face using the five facial feature points provided by SHORE™ as reference points. Thus, the normalized image always has the same resolution and pose. In this way, some of the variations in the appearance of the face that are caused by head rotations and movements of the person in front of the video capture device, are mitigated.
- **Facial Feature Point Detection:** This unit determines the location of additional facial feature points within the detected face region (Figure 1.a). These points (face mesh) cover the whole face and track the location of prominent spots on the human face, such as lip boundaries, eye brows, chin, etc. Changes in the location of these points from one frame to another provide geometric information about facial motion and the activated AUs.
- **Texture Classification:** However, it is not possible to detect all AUs just by observing motion in these facial feature points, since some of them are more prominently expressed by wrinkles. Such AUs are easier to recognise from the transient changes in the appearance of the face; for example, the combination of AU01 (Inner Brow Raiser) and AU04 (Brow Lowerer) is easily recognised from the appearance of wrinkles on the forehead region (Figure 1.b). So, in addition to the facial feature points, the facial texture is analysed. SVM classifiers trained on texture features are used for detecting specific AUs and AU combinations.
- **Action Unit Intensity Estimation:** The Kalman filter based action unit intensity estimator fuses the outputs from the facial feature point detector and the texture classification unit to get a final estimate of the intensity of each AU in a pre-defined set of 22 AUs. A biomechanical model of facial muscles is used to model the dynamics of the AUs. The intensities of AUs are modelled as the parameters of an anatomically based deformable face model. In addition to estimating AU intensities, this unit also simultaneously performs an online dynamic calibration of the person's neutral face. This online calibration is neccesary because it is often not possible to acquire a neutral face on demand.

4 FUSION APPROACH

In our system, we fuse observations (measurements) from two types of sources, namely geometric and texture, within the Kalman filter framework to estimate the intensities of various AUs. The geometric measurements are the positions of facial feature points and the texture measurements are the class-wise success probabilities from SVMs trained on texture features extracted from the image.

The Kalman filter [15] is a special form of dynamic Bayesian network that is applied to continuous state spaces with Gaussian transition and observation probability density functions. Kalman filtering involves two steps: *predict* and *update*. In the *predict* step, a dynamic process model is applied to the previous state to predict the current state. The predicted estimate is called the *apriori* state estimate. In the *update* step, one or more measurements are used to correct the *apriori* state to obtain the filtered or *aposteriori* state estimate. The noise in the measurements are assumed to follow the (multivariate) zero-mean Gaussian distribution.

In the *update* step, the Kalman filter allows to fuse measurements from multiple sources, provided each source has a Gaussian noise model. The fusion is performed on the basis of the uncertainties in the measurements. The more reliable measurements contribute more to the state update than the less reliable measurements. To incorporate a measurement into the Kalman filter, two components are required:

1. A measurement model that maps the state variables to the measured variables.
2. A covariance matrix that describes the Gaussian noise in the measured variables.

The following subsections describe how to model these two components for the geometric and texture measurements used in our system.

4.1 Geometric measurements

The geometric measurements include the positions of 68 facial feature points detected by the facial feature point detection unit. These are detected using a face alignment method. The measurement model for geometric measurements is given by a 3D 68-point deformable shape model that is similar to the CANDIDE face model [1]. The covariance matrix for the noise associated with the geometric measurements is determined empirically by applying the face alignment method to an annotated dataset, such as the Extended Cohn-Kanade dataset [21]. The differences between the detected positions of facial feature points and their annotated positions are normalised relative to the annotated eye distance. The variances and covariances of these normalised errors constitute the elements of the covariance matrix. The noise in each point measurement could be assumed to be independent of that in others. In this case, the matrix would be block-diagonal. Alternatively, the noise in each point measurement could be assumed to be correlated to that in every other. In this case, the matrix would be a full matrix. At runtime, the covariance matrix is scaled by the square of the measured eye distance.

4.2 Texture measurements

The texture measurements include the probability outputs for different AU classes provided by SVMs trained on texture (appearance-based) features such as HOG or LBP. These measurements are provided by the texture classification unit of the system. The open source software library LIBSVM [4] is used to realise the texture classifiers. The probabilities of AU classes are interpreted as the intensities of corresponding AUs. This is under the assumption that the higher the intensity of an AU, the stronger the textural changes and the greater the class probability, and viceversa. Therefore, the measurement models for texture measurements are the identity functions involving the corresponding AU intensity parameters. This directly maps the probability of an AU class to its intensity of expression.

The probability of an AU class is determined in different ways, depending on the output configuration of SVM (two-class or multi-class). LIBSVM provides the probability of each SVM output class using the method based on pairwise coupling [32]. These are converted into AU class probabilities according to the output class definitions. Three cases are discussed below.

- **Case A:** A two-class SVM that detects the presence or absence of an AU A. This is the simplest case, where the probability output for class A provided by the SVM is used as-is.
- **Case B:** A multiclass SVM that detects all possible boolean combinations of occurrence of two or more AUs. In Table 1, an example involving two AUs A and B is given. In such cases, the probability of each AU can be obtained through marginalisation. From Table 1, the probability of A is computed as $p + q$, and that of B is computed as $p + r$.

- **Case C:** A multiclass SVM that detects several individual AUs. For example, a four-class SVM for A, B, C and rest. The probability of A is the output of the SVM for class A. The probability of absence of A is the sum of the probabilities for the other three classes. This is a generalisation of Case A.

Table 1. Table listing the four boolean combinations of occurrence of two AUs: A and B, and notations for the probability outputs from a corresponding four-class SVM.

Boolean Combinations	Probability Notations
A and B	p
A and not B	q
not A and B	r
not A and not B	s

p, q, r and s add to 1 (exhaustive)

The probabilities of AU classes so computed define a Bernoulli distribution for individual AUs. For a Bernoulli distributed AU variable A, the outcome '1' indicates the presence of A and the outcome '0' indicates the absence of A. If the probability of presence of A is p, then the first moment or expected value μ is computed as $\mu = 0(1-p) + 1(p) = p$. The expected value is therefore identical to the probability of presence of A. The second moment or variance σ^2 is given by $p(1-p)$. Figure 2 illustrates how the variance varies according to the probability p (a.k.a probability of success). As the probability approaches the extremities 0 or 1, the variance decreases, which indicates increasing confidence in the texture measurements. As the probablity approaches 0.5, the variance increases, which indicates decreasing confidence in the texture measurements. The skewness of the Bernoulli distribution of A is computed as $(1-2p)/\sqrt{p(1-p)}$. As a rule of thumb, normality could be assumed when the skewness varies between -2 and 2. This corresponds approximately to the probability range $[0.146, 0.854]$ for p. For simplicity and practical convenience, we assume normality throughout the probability range $[0, 1]$. Therefore, we use the second moment σ^2 of the Bernoulli distribution as the variance of the Gaussian noise associated with the texture measurement for A.

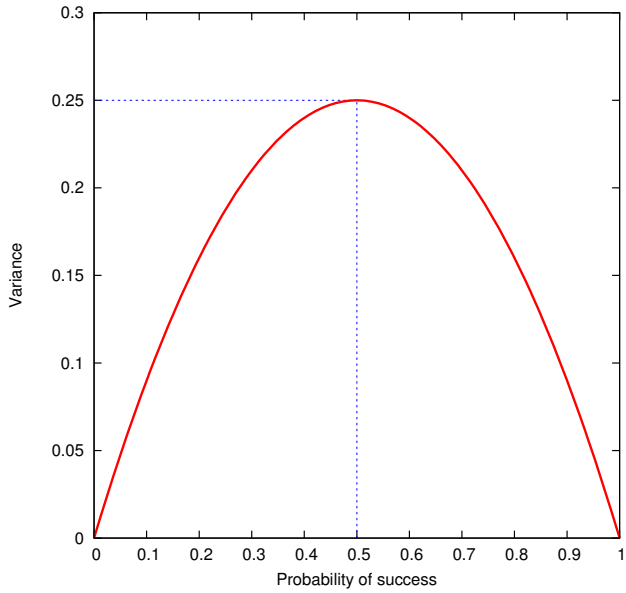


Figure 2. Variance of a Bernoulli distributed random variable

5 EVALUATION AND RESULTS

5.1 Dataset

We use an undisclosed dataset provided by GfK Verein for evaluation of the system and the proposed fusion approach. This dataset contains 301 videos of 93 different subjects (female and male). Several short advertisement clips were shown to the subjects and their facial reaction was recorded using a webcam, which was placed on top of the display screen. Each recording lasts approximately eight seconds and was recorded at 25 frames per sec. The subjects were recorded in a light-controlled setting and were instructed to act naturally. The responses often comprised of very subtle changes in facial expressions when exposed to the specific stimuli in the advertisement. Each frame is annotated by FACS experts with a list of active AUs.

5.2 Performance measure

We measure the performance of our system by creating the Receiver Operating Characteristic (ROC) curve [11] for each AU. ROCs illustrate the performance of a binary classifier system as its discrimination threshold is varied. The curve is a plot of the true positive rate against the false positive rate for various thresholds. Since ROCs are graphical plots, we compute the Area Under Curve (AUC) for each curve to get a single numerical metric. This makes comparison of performance of different configurations easier. A value for AUC that is closer to unity would indicate better performance.

5.3 Evaluation

We evaluate the proposed fusion approach on three upper face AUs, namely AU01 (Inner Brow Raiser), AU04 (Brow Lowerer) and AU06 (Cheek Raiser). As an illustration of **Case A** mentioned in Section 4.2, we use a pre-trained SVM for detecting the presence of AU06. As an illustration of **Case B** mentioned in Section 4.2, we use a pre-trained multiclass SVM for detecting the four possible Boolean combinations of AU01 and AU04 as mentioned in Table 1. **Case C** is a generalisation of **Case A**, and therefore, it is not separately evaluated. Tuning of system parameters as well as training and cross-validation of SVMs were performed using images selected from the CK+ dataset and an internal dataset of actors performing different AUs. The dataset provided by GfK Verein was not used for these purposes.

Table 2. AUC values for AU01, AU04 and AU06 for the fused, geometry-only and texture-only configurations.

Configuration	AU01	AU04	AU06
Fused	0.82	0.84	0.73
Geometry-only	0.77	0.79	0.64
Texture-only	0.78	0.79	0.66

In Table 2, the AUC values obtained after fusing geometric and texture information are provided. The corresponding ROC curves are shown in Figure 3. The improvement in performance obtained over (a) using only geometric information (geometry-only configuration),

and (b) using SVM probability outputs independently of Kalman filter framework (texture-only configuration), is illustrated in Figure 4. The evaluation shows that fusing geometric and texture information is better than using either of them alone. This indicates that, for AU01, AU04 and AU06, shape and appearance provide complementary information that helps in improving their recognition rates.

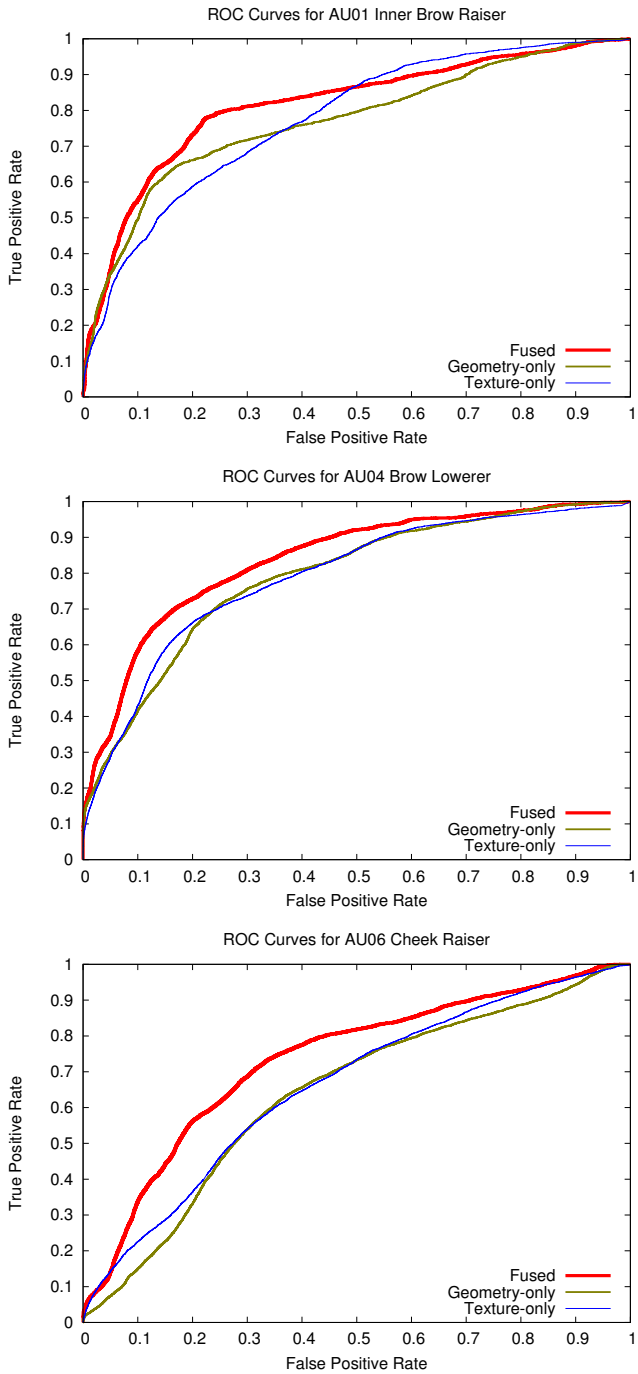


Figure 3. ROC curves for AU01, AU04 and AU06, for the fused, geometry-only and texture-only configurations.

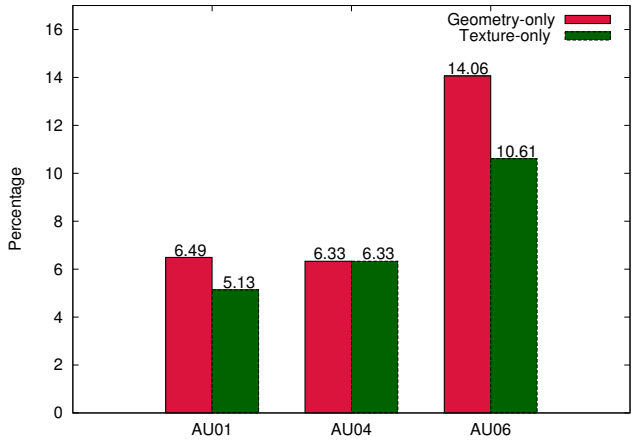


Figure 4. Percentage improvement in AUC of fused configuration over geometry-only and texture-only configurations for AU01, AU04 and AU06.

6 CONCLUSION

In this paper, we proposed a method to fuse categorical probabilities within a Kalman filter framework. This was illustrated in an application for facial action estimation by fusing geometric measurements of facial feature points with AU probabilities from SVM based texture classifiers. The results show that fusion of geometric and texture information using the proposed method clearly outperforms the geometry-only and texture-only configurations. This also illustrates the practical applicability of the proposed fusion method.

The proposed fusion method currently assumes normality across the entire range of probability output. However, the skewness of the distribution increases rapidly when the probability output approaches the extremities 0 and 1. Therefore, possible future work could include strategies for dealing with the skewness of the Bernoulli distribution of texture information.

ACKNOWLEDGEMENTS

We would like to thank the GfK Verein, in particular Anja Dieckmann and Matthias Unfried, for their support, valuable feedback and provision of the facial expression dataset. We would also like to thank Andreas Ernst and Sebastian Hettenkofer of Fraunhofer IIS for their feedback and active participation in discussions on the topic of this paper.

REFERENCES

[1] J. Ahlberg, 'CANDIDE-3 – an updated parameterized face', Technical Report LiTH-ISY-R-2326, Department of Electrical Engineering, Linköping University, Sweden, (2001).

[2] T. R. Almaev and M. F. Valstar, 'Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition', in *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pp. 356–361, (Sept 2013).

[3] T. Baltrusaitis, M. Mahmoud, and P. Robinson, 'Cross-dataset learning and person-specific normalisation for automatic action unit detection', in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 06, pp. 1–6, (May 2015).

[4] Chih-Chung Chang and Chih-Jen Lin, 'LIBSVM: A library for support vector machines', *ACM Transactions on Intelligent Systems and Technology*, **2**, 27:1–27:27, (2011). Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[5] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor, 'Active appearance models', *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6), 681–685, (2001).

[6] N. Dalal and B. Triggs, 'Histograms of oriented gradients for human detection', in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pp. 886–893 vol. 1, (June 2005).

[7] F. Dornaika and F. Davoine, 'On appearance based face and facial action tracking', *IEEE Transactions on Circuits and Systems for Video Technology*, **16**(9), 1107–1124, (Sept 2006).

[8] Fadi Dornaika and Franck Davoine, 'Simultaneous facial action tracking and expression recognition in the presence of head motion', *International Journal of Computer Vision*, **76**(3), 257–281, (2007).

[9] Paul Ekman and Wallace V. Friesen, *Facial action coding system: A technique for the measurement of facial movement*, Consulting Psychologists Press, Palo Alto, CA, 1978.

[10] B. Fasel and Juergen Luettin, 'Automatic facial expression analysis: a survey', *Pattern Recognition*, **36**(1), 259 – 275, (2003).

[11] Tom Fawcett, 'An introduction to {ROC} analysis', *Pattern Recognition Letters*, **27**(8), 861 – 874, (2006). {ROC} Analysis in Pattern Recognition.

[12] J. U. Garbas, T. Ruf, M. Unfried, and A. Dieckmann, 'Towards robust real-time valence recognition from facial expressions for market research applications', in *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pp. 570–575, (Sept 2013).

[13] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, 'Local binary patterns and its application to facial image analysis: A survey', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **41**(6), 765–781, (Nov 2011).

[14] Nils Ingemars. A feature based face tracker using extended Kalman filtering, 2007.

[15] R.E. Kalman, 'A new approach to linear filtering and prediction problems', *Journal of Basic Engineering*, **82**(1), 35–45, (March 1960).

[16] I. Kotsia, N. Nikolaidis, and I. Pitas, 'Fusion of geometrical and texture information for facial expression recognition', in *Image Processing, 2006 IEEE International Conference on*, pp. 2649–2652, (Oct 2006).

[17] I. Kotsia, S. Zafeiriou, N. Nikolaidis, and I. Pitas, 'Texture and shape information fusion for facial action unit recognition', in *Advances in Computer-Human Interaction, 2008 First International Conference on*, pp. 77–82, (Feb 2008).

[18] Sebastian Krappe, Michaela Benz, Thomas Wittenberg, Torsten Haferlach, and Christian Mnzenmayer. Automated classification of bone marrow cells in microscopic images for diagnosis of leukemia: a comparison of two classification schemes with respect to the segmentation quality, 2015.

[19] Sven E Krüger, Martin Schafföner, Marcel Katz, Edin Andelic, and Andreas Wendemuth, 'Speech recognition with support vector machines in a hybrid system.', in *INTERSPEECH*, pp. 993–996. Citeseer, (2005).

[20] Christian Küblbeck and Andreas Ernst, 'Face detection and tracking in video sequences using the modified census transformation', *Image Vision Comput.*, **24**(6), 564–572, (June 2006).

[21] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, 'The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression', in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pp. 94–101, (June 2010).

[22] Zuheng Ming, A. Bugeau, J. L. Rouas, and T. Shochi, 'Facial action units intensity estimation by the fusion of features with multi-kernel support vector machine', in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 06, pp. 1–6, (May 2015).

[23] Ahmad Poursaberi, Hossein Ahmadi Noubari, Marina Gavrilova, and Svetlana N. Yanushkevich, 'Gauss–laguerre wavelet textural feature fusion with geometrical information for facial expression identification', *EURASIP Journal on Image and Video Processing*, **2012**(1), 1–13, (2012).

[24] Ognjen Rudovic, Mihalis A Nicolaou, Vladimir Pavlovic, R Walecki, O Rudovic, V Pavlovic, M Pantic, S Eleftheriadis, O Rudovic, M Pantic, et al., 'Machine learning methods for social signal processing', *Social Signal Processing*, **30**, 469–484, (2014).

[25] Tobias Ruf, Andreas Ernst, and Christian Küblbeck, 'Face detection with the sophisticated high-speed object recognition engine (shore)', in *Microelectronic Systems*, 243–252, Springer, (2011).

[26] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn, 'Deformable model fitting by regularized landmark mean-shift', *Int. J. Comput. Vision*, **91**(2), 200–215, (January 2011).

[27] E. Sariyanidi, H. Gunes, and A. Cavallaro, 'Automatic Analysis of Facial Affect: A Survey of Registration, Representation, and Recognition', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **37**(6), 1113–1133, (June 2015).

[28] T. Senechal, V. Rapp, H. Salam, R. Seguier, K. Bailly, and L. Prevost, 'Facial action recognition combining heterogeneous features via multikernel learning', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **42**(4), 993–1005, (Aug 2012).

[29] M. F. Valstar and M. Pantic, 'Fully automatic recognition of the temporal phases of facial actions', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **42**(1), 28–43, (Feb 2012).

[30] Michel F. Valstar and Maja Pantic, *Human–Computer Interaction: IEEE International Workshop, HCI 2007 Rio de Janeiro, Brazil, October 20, 2007 Proceedings*, chapter Combined Support Vector Machines and Hidden Markov Models for Modeling Facial Action Temporal Dynamics, 118–127, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[31] Paul Viola and MichaelJ. Jones, 'Robust real-time face detection', *International Journal of Computer Vision*, **57**(2), 137–154, (2004).

[32] Ting-Fan Wu, Chih-Jen Lin, and Ruby C. Weng, 'Probability estimates for multi-class classification by pairwise coupling', *J. Mach. Learn. Res.*, **5**, 975–1005, (December 2004).

[33] Jizheng Yi, Xia Mao, Lijiang Chen, Yuli Xue, and A. Compare, 'Facial expression recognition considering individual differences in facial structure and texture', *IET Computer Vision*, **8**(5), 429–440, (October 2014).

[34] Hui Yu and Honghai Liu. Combining appearance and geometric features for facial expression recognition, 2015.

[35] Z. Zeng, M. Pantic, G.I. Roisman, and T.S. Huang, 'A survey of affect recognition methods: Audio, visual, and spontaneous expressions', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **31**(1), 39–58, (Jan 2009).

[36] Ligang Zhang, Dian Tjondronegoro, and Vinod Chandran, *Neural Information Processing: 18th International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings, Part III*, chapter Geometry vs. Appearance for Discriminating between Posed and Spontaneous Emotions, 431–440, Springer, Heidelberg, 2011.

[37] Yongmian Zhang and Qiang Ji, 'Active and dynamic information fusion for facial expression understanding from image sequences', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(5), 699–714, (May 2005).

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

doi:10.3233/978-1-61499-672-9-1818

Planning Tourist Agendas for Different Travel Styles

Jesus Ibañez and **Laura Sebastia** and **Eva Onaindia**[1]

Abstract. This paper describes *e-Tourism2.0*, a web-based recommendation and planning system for tourism activities that takes into account the preferences that define the travel style of the user. *e-Tourism2.0* features a recommender system with access to various web services in order to obtain updated information about locations, monuments, opening hours, or transportation modes. The planning system of *e-Tourism2.0* models the taste and travel style preferences of the user and creates a planning problem which is later solved by a planner, returning a personalized plan (agenda) for the tourist. *e-Tourism2.0* contributes with a special module that calculates the recommendable duration of a visit for a user and the modeling of preferences into a planning problem.

1 INTRODUCTION

A **Recommender System** (RS) [13] is a personalization tool aimed to provide the items that best fit the individual tastes of people. A RS infers the user preferences by analyzing the available user data, information of other users and of the environment. The target of the extensively popularized Tourism RSs (TRSs) is to match the user preferences with the leisure resources and tourist activities of a city [15] by using some initial data, usually explicitly provided by the user. The relevance of TRSs relies in their capacity of automatically inferring the user preferences, through an explicit or implicit feedback of the user, as well as providing the user with a personal tourist activity agenda. Typically, TRSs use a hybrid approach of recommendation techniques such as demographic, content-based or collaborative filtering [2] and they are confined to recommendations within a delimited area or city since tourism infrastructure is usually developed to promote the tourism demand in particular spots [6, 11].

The latest developments in TRSs share a common mainstream, that of providing the most user-tailored tourist proposal. Hence, some tools like *SAMAP* [3] elicits a tourist plan with recommendations about the transportation mode, restaurants and bars or leisure attractions such as cinemas or theaters, all this accompanied with a detailed plan explanation. Scheduled routes presented in a map along with a timetable are nowadays a common functionality of many TRSs, like *e-Tourism* [6], which also include context information such as the opening and closing hours of the Points Of Interest (POIs) to visit and the geographical distances between POIs to compute the time to move from one place to another. Some other tools allow the user to interact with the plan or develop interfaces specifically designed to be used in mobile devices ([12, 16]). Personalization is interpreted in *CT-Planner* [9] as emphasizing the concept of interactive assistance between the user and a tour advisor, where the advisor offers several plans, learns the tourist preferences, requests feedback from the users and customizes the plans accordingly. *CT-Planner* also accounts for user preferences like the walking speed or reluctance to walk, in which case the planner will suggest short walking distances in the plan.

Recent advances in TRSs go one step ahead towards personalization and propose to adapt the duration of the visits to the user preferences. For instance, *PersTour* [10] calculates a personalized duration of a visit to POI using the POI popularity and the user interest preferences, which are automatically derived from real-life travel sequences based on geotagged photos. And the work in [14] considers user preferences based on the the number of days of the trip and the pace of the tour, that is, whether the user wants to perform many activities in one day or travel at a more relaxed pace.

In this paper, we present *e-Tourism2.0*, a TRS that draws upon the recommendation model and planning module of *e-Tourism* [6] and significantly enhances the personalization of the recommendations. *e-Tourism2.0* improves *e-Tourism* in two main aspects:

- *context-aware tool*: it establishes a connection to several web services to capture up-to-date context information such as opening hours of POIs to visit, location of POIs, ratings of users, modes of transport in the city, etc.;
- *preference temporal planning*: it handles a full range of user preferences such as the user interest in visiting a POI, the pace of the tour (relaxed vs busy) and variable durations of the visits within a temporal interval; all these preferences represent the user travel style. *e-Tourism2.0* uses OPTIC [1], a state-of-the-art planner that addresses the full set of preferences defined in PDDL3.0 language [7].

This paper is organized as follows. Section 2 summarizes the main aspects of *e-Tourism*. Section 3 explains the procedure to calculate the recommended duration of an activity for a given user and section 4 details the construction of the planning problem and the encoding of the user preferences within the planning problem. Section 5 shows several cases of study to test whether the defined preferences are taken into account correctly by the planner and last section concludes.

2 *e-Tourism2.0* TOOL

e-Tourism [6] was developed as a web application to generate recommendations about personalized tourist tours in the city

[1] Universitat Politècnica de València, Valencia, Spain, Email: {jeibrui, lstarin, onaindia}@dsic.upv.es

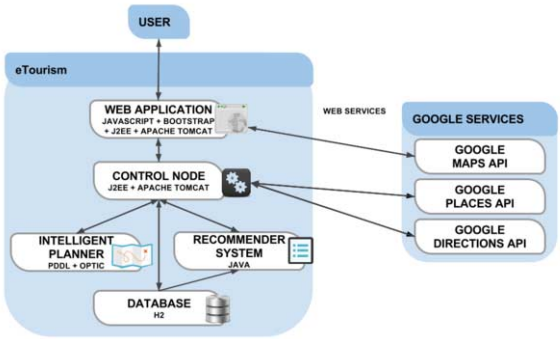


Figure 1. *e-Tourism2.0* Architecture

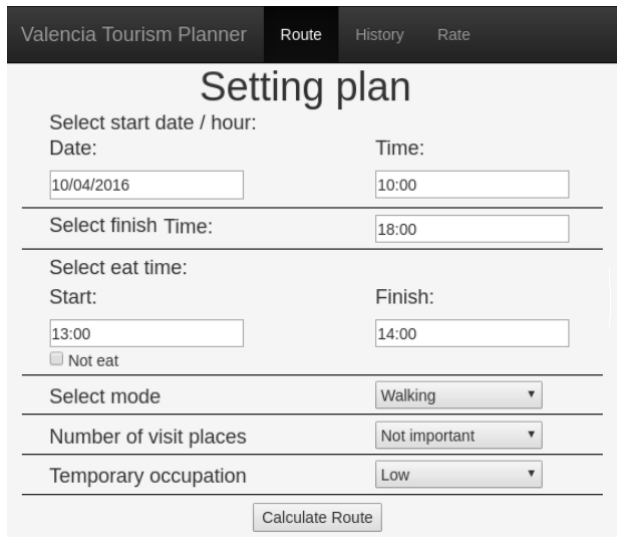


Figure 2. *e-Tourism2.0* system: agenda preferences.

of Valencia (Spain). It was intended to be a service for foreigners and locals to become deeply familiar with the city and plan leisure activities. *e-Tourism* makes recommendations based on the user's tastes, her demographic classification, the places visited by the user in former trips and, finally, her current visit preferences. One of the main components of *e-Tourism* is the planning module, aimed at scheduling the recommended activities. Thus, the output of *e-Tourism* is a real agenda of activities which not only reflects the user's tastes but also provides details on when to perform the recommended activities. Specifically, the construction of the agenda takes into account duration of the activities to perform, the opening hours of the places to visit and the geographical distances between places (time to move from one place to another). All this information is compiled into a planning problem that can be formulated as a *Constraint Satisfaction Problem* or as an *Automated Planning Problem* [8].

Let us take three tourists, Rose, Mark and David, interested in visiting Valencia. Rose and Mark like visiting museums, but Rose likes museums more than Mark. Both decide to visit the *National Museum of Ceramics*. Rose wishes to visit the museum for 2h30min, whereas Mark only wants to be there for about 1h30min. Moreover, since this is Mark's first time in Valencia, he would like to include quite a few POIs in his agenda, namely 5 POIs, and not to have much spare time between activities. Rose, however, visited Valencia last year and she would like to explore in depth two museums that she already visited last time. Therefore, she would like her agenda to contain only these two visits over a full day and no much free time between them. In contrast, David has been in Valencia several times and he would rather include in his agenda two or three quick visits and spare time to walk around and sit in a terrace to have a beer. These three examples show different travel styles around two preferences: the number of visits and the time spent in each visit. *e-Tourism2.0* handles taste preferences of the user as well as this new type of *travel style preferences*.

The *e-Tourism2.0* architecture is composed of five subsystems (Figure 1): the control node, responsible of coordinating the whole recommendation-planning process, the web application, the recommender system, the intelligent planner and the database.

2.1 Tourist agenda

We developed a new web-based interface which can be accessed through different devices such as computers, smartphones, tablets, etc. The first step in the construction of the tourist agenda is to **build the user model**. The user registers in the system and enters her personal details and general preferences. With this information the system builds an initial user profile. Besides, each time the user enters the system for a new visit she will be requested to introduce her specific preferences for the current visit, shown in Figure 2: the date of the visit *date*, her available time slot (T_s^{tour}, T_e^{tour}), the time interval reserved for lunch $(T_s^{lunch}, T_e^{lunch})$, the mode of transport she prefers - walking, driving or public transport -, her initial location $location_{initial}$ and final destination $location_{final}$. Moreover, she also indicates her preferences related to her travel style: $pref_{\#visits}$ indicates if the user prefers to include many or few visits in the tour or has no preference over it; and $pref_{occupation}$ indicates if the user prefers to obtain an agenda with a high or a low temporal occupation or has no preference over it.

The second step is to **generate a list of activities that are likely of interest to the user** by means of the *Generalist Recommender System Kernel (GRSK)*, which uses a mixed hybrid recommendation technique. A detailed description of GRSK can be found in [5]. The *intelligent planner* is in charge of **calculating the tourist agenda**, scheduling the activities recommended by the GRSK according to the restrictions of the environment and user preferences with respect to the configuration of the agenda. Figure 4 shows two agendas computed for a particular user and a map with the path she should follow. When the user logs again in the system, she is asked to **rate the activities** in the last recommended plan (through the option *Rate* in the top bar menu of Figure 2). The information obtained from these ratings is further used to improve the user profile and provide more suitable recommendations.

2.2 Database

The database schema of *e-Tourism2.0* is shown in Figure 3. We manage two sets of tables: those used for the recommendation process and those used for the planning process. Table `places` stores information about the POIs to recommend such as the name or the geographical coordinates. Table `users` contains personal details of the user, such as the name and other demographic data (this is neglected in Figure 3 for the sake of clarity). These two tables are used in both processes. The information used by the GRSK is: (1) tables `preferences`, `places_preferences` and `users_preferences`, which store the characteristics of the POIs to recommend and the user preferences inferred by the GRSK, respectively[2]; (2) tables `history` and `history_data`, which store the past interaction of the user with the system. The planner uses the information in table `timetables`, which stores a list of opening hours for each POI, and `movements_time`, that keeps the estimated and actual travelling time between two locations according to the value of `travel_mode` (see Figure 3).

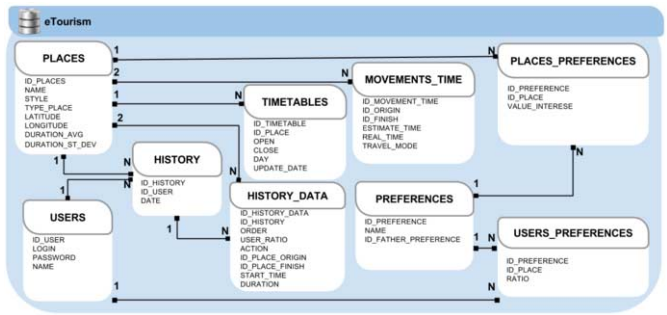


Figure 3. *e-Tourism2.0* system: database.

2.3 External data sources

As explained above, *e-Tourism2.0* accesses various web services in order to obtain some up-to-date information about location of restaurants and POIs, opening hours, transportation modes, etc. For obtaining this information, we selected the Google location and mobility web services, specifically:

- Google Directions[3] for obtaining a route (path) between two given coordinates, addresses or name of places. It is also possible to add some intermediate points in the path and to select the travel mode (walking, cycling, driving or with public transport)
- Google Places[4] for obtaining information about a given place. In *e-Tourism2.0*, this service has been used to elicit the opening hours of the places to visit and to find restaurants close to an specific place.
- Google Maps[5] for the visualization of the map along with the route provided to the user with the recommended places to visit.

Information like the catalog of POIs or the route between two places is stored in the database, which allows us to accelerate the process of calculating the recommendations and the plan. However, since information can become obsolete and needs to be updated, Google web services are periodically queried to update the data (see section 4 for more details).

3 RECOMMENDATION OF THE VISIT DURATION

The GRSK of *e-Tourism2.0* elicits the list of POIs or activities to include in the travel agenda of the user according to her preferences. This list is an ordered set of tuples of the form: $\langle a, Pr^a \rangle$, where a denotes the recommended activity and $Pr^a \in [0, 300]$ is the estimated degree of interest of the user in activity a.

For each activity a, we assign a duration in average, denoted by μ_a, which represents the recommendable duration of a for a typical tourist. The value of μ_a joint with σ_a define a normal distribution $X(\mu_a, \sigma_a^2)$. This is used by the GRSK to return a time interval that encompasses the minimum and maximum recommendable duration of a for the user according to Pr^a. Following the definition of the normal distribution, σ_a is computed as μ_a divided by α, so that, 68% of tourists spend $[\mu_a - \mu_a/\alpha, \mu_a + \mu_a/\alpha]$ minutes in visiting a, whereas about 4% of the tourists spend less than $\mu_a - 2 * \mu_a/\alpha$ or more than $\mu_a + 2 * \mu_a/\alpha$ minutes. In our experiments, we set $\alpha = 5$ and we empirically tested that consistent durations are returned. Our future objective is to estimate this distribution by studying the actual behaviour of tourists by means of an analysis of Twitter interactions, similarly to the analysis described in [10].

Once the normal distribution $X(\mu_a, \sigma_a^2)$ for each activity is defined, the recommended interval $(dur^a_{min}, dur^a_{max})$ is computed as $(X(Pr^a/300/2), X(Pr^a/300))$. That is, the values of probability that leave an area of the corresponding argument on the right. For example, let's assume that the *a=National Museum of Ceramics* has $\mu_a = 180$ and, therefore, $\sigma_a = 36$, meaning that a typical tourist would spend 180 minutes visiting this museum, and the dispersion for the other tourists is 36 minutes. Then, by the normal distribution, 68% of the tourists spend between [144,216] minutes in this visit and approximately 4% of the tourists spend less than 108 or more than 252 minutes. If the GRSK determines a degree of interest of 100 out of 300 for a given user, the duration interval will be $[145, 164]$, whereas if Pr^a is 260, the duration interval will be $[174, 220]$.

In [10], the visit duration is adjusted with the category of the activity a and the interest of the user in the category. However, durations in *eTourism2.0* are more accurate because we consider the degree of interest of the user in a, not in the category of a. Moreover, since the GRSK returns a tuple of the form $\langle a, Pr^a, dur^a_{min}, dur^a_{max} \rangle$ for each a, the planner can select the most appropriate duration within the interval the according to the travel style preferences of the user.

[2] A more detailed explanation about the domain ontology can be found in [4].
[3] https://developers.google.com/maps/documentation/directions/
[4] https://developers.google.com/places/
[5] https://developers.google.com/maps/documentation/javascript/

4 PLANNING PROBLEM SOLVING

The Control node receives the list of the recommended activities along with the recommended duration interval from the GRSK and generates the planning problem. Planning a set of recommended activities for a tourist requires some functionalities: (1) temporal planning and management of durative actions (e.g., duration of visits, time spent in transportation, etc.); (2) ability of reasoning with temporal constraints (e.g., scheduling the activities within the opening hours of places, planning the tour within the available time slot of the tourist, etc.) and (3) ability of reasoning with the tourist preferences (e.g., selecting the preferred activities of the user for planning the tour). Reasoning with time constraints and preferences simultaneously is a big challenge for current temporal planners.

Among the few automated planners are capable of handling temporal planning problems with preferences, we opted for OPTIC because it handles the version 3.0 of the popular Planning Domain Definition Language (PDDL) [7], including non-fixed durations and soft goals. Soft goals are preferences that we wish to satisfy in order to generate a good plan, but that do not have to be achieved in order for the plan to be correct. We need to identify and describe the preferences in PDDL3.0 as well as stating how the satisfaction, or violation, of these constraints affects the quality of a plan. Thus, the violation costs (penalties) associated to the preferences are considered at the time of selecting the best tourist plan; i.e., the plan that satisfies most tourist preferences and thereby minimizes the violation costs. This section describes the automatic generation of the corresponding planning problem in PDDL3.0.

4.1 Initial state

The specific values of the variables of a problem are described in the initial state by means of predicates and functions. The predicates and functions for an activity are:

- The interval duration of an action (activity) a is defined through the functions `(min_visit_duration ?a)` and `(max_visit_duration ?a)`. They will be assigned the values dur^a_{min} and dur^a_{max} returned by the GRSK, respectively.
- An activity a has an opening hour and a closing hour that are specified by a timed-initial literal: `(at` t_{open} `(open` a`))` and `(at` t_{close} `(not (open` a`)))`, to indicate when the activity is not longer available.

The duration of moving from one location p_j to another location p_k is defined by the function `(travelling_time` p_j p_k`)` that returns the time in minutes needed to travel from p_j to p_k by using the travel mode indicated by the user. If the duration of this action is not available in the DB from a past user, an estimated duration is calculated with the *Haversine* formula, used for calculating Earth distances, and the classical *uniform linear motion* formula, where *speed* depends on the mode of transport, and adding a small correction θ for awaiting times:

$$EstimTime(A,B) = \frac{Haversine(A,B)}{speed} + \theta * Haversine(A,B)$$

The predicate `(person_at ?l)` is used to represent the location of the user and the function `(total_available_time)` returns the available time of the user, which is initially set to $T_{finish} = T^{tour}_e - T^{tour}_s$.

We must note that web services are queried to obtain the initial data of the planning problem and that most of these data (timetables, distances between monuments) are stored in the database in order to keep the number of queries as low as possible and quickly retrieve the data during planning. In case a particular distance is not found in the database during the construction of a plan, we estimate the distance with the Haversine formula explained above, thus avoiding access to web services at planning time. Estimated times will be then updated after the planning process with the actual values by querying the corresponding web services.

4.2 Goal and preferences

We handle two types of goals: *hard goals*, that represent the realization of an activity that the user has specified as mandatory (e.g., the final destination at which the user wants to finish up the tour `(person_at id_hotelastoria)`); and *soft goals or preferences*, that represent the realization of a desirable but non-compulsory activity; e.g., visiting the *National Museum of Ceramics*: `(preference p1 (visit_location id_museumceramics))`.

The objective is to find a plan that achieves all the hard goals while minimizing a plan metric to maximize the preference satisfaction. This is expressed in the form of penalties, so that when a preference is not fulfilled, a penalty is added to the metric. Specifically, we define three types of penalties: for non-visited POIs, travelling times and the non-fulfillment of other configuration parameters of the agenda.

The *penalty for non-visited places*, aimed to help the planner to select the activities with a higher priority for the user, is calculated as the ratio between the priority of the activities not included in the plan Π and the priority of the whole set of recommended activities RA:

$$P_{non\_visited} = \frac{\sum_{a \in RA - \Pi} Pr^a}{\sum_{a \in RA} Pr^a}$$

The *penalty for movements* forces the planner to reduce the time spent in travelling from one location to another, so that closer activities are visited consecutively. This penalty is calculated as the duration of the *move* actions of Π, Π_m:

$$P_{move} = \sum_{a \in \Pi_m} dur(a)$$

Initially, the user defines her travel style preferences (see section 2): $pref_{\#visits}$ represents the preference for the number of visits and $pref_{occupation}$ denotes the user preference for the time to be spent in the visits or, conversely, for the free time between activities. The idea of combining both preferences is to give response to the different travel styles described in section 2. For example, Rose would set $pref_{\#visits}$ to "few" and $pref_{occupation}$ to "high". In order to take into account these preferences, two penalties are included.

$P_{\#visits}$ is the penalty that considers the user preference for the number of visits. It takes into account the number of visits in the plan with respect to the number of recommended activities:

$$\begin{cases} \frac{(|RA|-|\Pi_v|)}{|RA|} * T_{finish} & : \quad pref_{\#visits} = many \\ \frac{|\Pi_v|}{|RA|} * T_{finish} & : \quad pref_{\#visits} = few \\ 0 & : \quad pref_{\#visits} = indifferent \end{cases}$$

$P_{occupation}$ is the penalty that considers the user preference for the temporal occupation. Similarly to $P_{\#visits}$, $P_{occupation}$ takes into account the time that remains available in the plan with respect to the total time of the user.

$$\begin{cases} T_{finish} - \sum_{a \in \Pi} dur(a) & : \quad pref_{occupation} = high \\ \sum_{a \in \Pi} dur(a) & : \quad pref_{occupation} = low \\ 0 & : \quad pref_{occupation} = indifferent \end{cases}$$

Both penalties return a value in the interval $[0, T_{finish}]$. The combination of all these penalties defines the plan metric or optimization function to minimize by the planner:

$$P_{total} = P_{non\_visited} + P_{move} + P_{\#visits} + P_{occupation}$$

4.3 Actions

Three different types of actions are defined in this tourism domain. Due to space restrictions, we will only focus on the **visit** action. The input parameters of this action are the activity to perform `?a` and the user `?y`. The duration of the action is defined within the interval `(min_visit_time ?a)` and `(max_visit_time ?a)`. Moreover, this duration must be smaller than the remaining available time `(total_available_time)`. The planner will choose the actual duration of the action according to these constraints. The conditions for this action to be applicable are: (1) the user must be located in `?a` during the whole execution of the action; (2) the POI `?a` is open during the whole execution of the action and (3) the activity `?a` has not been performed yet. The effects of the action assert that (1) the activity is done, (2) the number of visited locations is increased and (3) the user available time is updated according to the activity duration. The action to perform the activity of *having lunch* is similarly defined to the action `visit`. The action of *moving* between locations essentially modifies the current location of the user, the available time of the user and the time spent in travelling from one location to another according to the duration stored in the database.

Regarding the periodical update of the information, only the location of restaurants and distances between restaurants and monuments are not retrieved beforehand because the list of restaurants is rather changeable. The planner deals with a 'dummy' restaurant, which is instantiated to a real restaurant that matches the user's tastes after planning.

5 CASES OF STUDY

In this section, we show some cases of study and we analyze whether the resulting plans of the OPTIC planner are compliant with the user preferences. We use two metrics to measure the plan quality:

$$O_\Pi = \frac{\sum_{a \in \Pi} dur(a)}{T_{finish}} \qquad U_\Pi = \frac{\sum_{a \in \Pi_v} (Pr^a * dur(a))}{\sum_{a \in \Pi_v} dur(a)}$$

O_Π is the occupation rate of the plan; i.e., the total time during which the user is performing some action (visiting, moving or having lunch). U_Π is the utility of the plan, defined as the rate between the priority of the activities performed in a given interval and the total duration of such activities. U_Π returns a value in [0, 300].

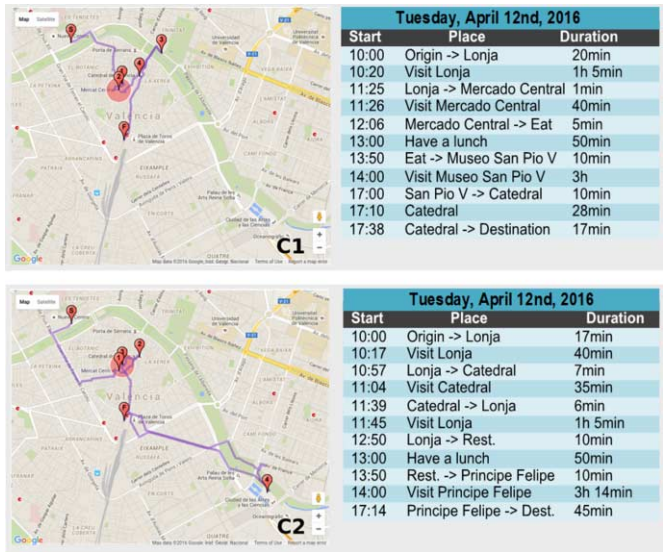


Figure 4. Plan generated for case studies C1 and C2

First, we performed a comparison to see how the selection of the mode of transport affects the final plan. Figure 4 shows the paths obtained for two cases: C1 and C2. C1 represents a basic case, where the user only specifies he would rather walk. In this case, the focus of the planner is on finding the best route taking into account the degree of interest of the user in the POIs, the opening hours and the reduction of the walking time. The resulting plan is an agenda with $O_\Pi = 90.625\%$ and $U_\Pi = 217.22$. The case C2 differs from C1 in that the user can either walk, or use the public transport when the distance between two consecutive places is greater than a threshold. In this case, the system generates routes that include POIs in which the user is highly interested but are far away from each other, returning an agenda with a higher utility. For example, the user is advised to use the public transport to visit *Museo Principe Felipe*, given that this POI is not within walking distance of the previous visited POI in the plan. In this case, $O_\Pi = 64.57\%$ and $U_\Pi = 251.62$.

In the next experiment, we selected a fixed initial and final locations, the available time slot, time reserved for lunch and transportation means, and we generated a set of cases with all the possible combinations of $pref_{\#visits}$ and $pref_{occupation}$. The results are shown in Table 1. Columns *#visits* and *occupation* indicate the value of the preferences $pref_{\#visits}$ and $pref_{occupation}$, respectively. Column *#POIs* shows the number of POIs included in the agenda, whereas columns *move* and *visit* indicates the percentage of the time devoted to move and visit actions, respectively. Finally, columns O_Π and U_Π indicate the occupation rate and the utility of the plan.

The results show that the preferences indicated by the user are effectively reflected in the agenda. We can observe that

#visits	*occupa*	#POIs	move	visit	O_Π	U_Π
Indiff	Indiff	3	7.2	28.7	52.59	220.96
Indiff	High	4	21.6	61.64	99.97	242.35
Indiff	Low	2	9.81	15.37	42.22	216.74
Many	Indiff	4	10.37	41.11	68.14	246.74
Many	High	4	21.66	61.64	99.97	242.35
Many	Low	4	10.37	37.59	64.62	217.68
Few	Indiff	2	6.6	19.44	42.77	236.66
Few	High	3	21.66	60.9	99.23	242.01
Few	Low	2	9.81	15.37	42.22	216.74

Table 1. Cases of study with different travel styles

when only one preference is set, clearly the other preference influences the final result of the agenda. For example, when $pref_{\#visits}$ is set to 'Indiff', the difference in O_Π is more than 57%. This also happens when $pref_{occupation}$ is set to 'Indiff', where the number of visited POIs goes from 4 to 2, depending on the value of $pref_{\#visits}$.

When $pref_{\#visits}$ is set to 'Many', the number of POIs is the highest (4), but we can observe a clear difference in O_Π depending on the value of $pref_{occupation}$: if it is set 'High', O_Π almost reaches 100%; and if it set to 'Low', then the value of O_Π is lower than the value obtained when $pref_{occupation}$ is 'Indiff'. We can find a similar situation when the number of visits is 'Few', where the only difference is that number of POIs to include in the agenda increases in 1 when $pref_{occupation}$ is set to 'High'.

In the resulting plans, that we do not show due to space restrictions, we have observed that when $pref_{occupation}$ is'Low', irrespective of the number of visits, the duration of the activity is usually set to the minimum value of the duration interval returned by GRSK. This is reflected in that the time of *visit* when $pref_{occupation}$ is 'Low' is always lower than the *visit* times when $pref_{occupation}$ is 'High' or 'Indiff'. Obviously, U_Π is also the lowest in these cases and the highest utility is always obtained when O_Π is also the highest.

The percentage of the time devoted to travelling actions is usually around 10%, except in the cases where the $pref_{occupation}$ is 'High'. This is because, in this particular case of study, the user must travel to a distant POI to obtain a high value of occupation.

The tourist-tailored plans obtained in the cases of study are the result of the planner's performance and of a faithful and consistent modeling of the user preferences and corresponding penalties.

6 CONCLUSIONS

This paper describes *e-Tourism2.0*, an enhanced recommendation and planning system for tourist activities in the city of Valencia (Spain). *e-Tourism2.0* offers a personalized recommendation of the duration of the visits suited to the interest of the user in the place to visit. It also handles user preferences related with the configuration of the agenda, particularly travel style preferences in terms of the number of places to visit and the desired temporal occupation of the tour.

We tested the adaptiveness of the plans to the user preferences through some cases of study. From the results we can conclude that an accurate modeling of the user preferences is very relevant to obtain plans that effectively reflect the tastes and travel style preferences of the tourist.

ACKNOWLEDGEMENTS

This work has been partly supported by the Spanish MINECO under the project TIN2014-55637-C2-2-R, and the Valencian project PROMETEOII/2013/019.

REFERENCES

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

[2] Joan Borràs, Antonio Moreno, and Aïda Valls, 'Intelligent tourism recommender systems: A survey', *Expert Syst. Appl.*, **41**(16), 7370–7389, (2014).

[3] Luis A. Castillo, Eva Armengol, Eva Onaindia, Laura Sebastia, Jesús González-Boticario, Antonio Rodríguez, Susana Fernández, Juan D. Arias, and Daniel Borrajo, 'samap: An user-oriented adaptive system for planning tourist visits', *Expert Syst. Appl.*, **34**(2), 1318–1332, (2008).

[4] Inma Garcia, Sergio Pajares, Laura Sebastia, and Eva Onaindia, 'Preference elicitation techniques for group recommender systems', *Information Sciences*, **189**, 155–175, (2012).

[5] Inma Garcia, Laura Sebastia, and Eva Onaindia, 'On the design of individual and group recommender systems for tourism', *Expert Systems with Applications*, **38**(6), 7683–7692, (2011).

[6] Inma Garcia, Laura Sebastia, Eva Onaindia, and Cesar Guzman, 'e-Tourism: a tourist recommendation and planning application', *International Journal on Artificial Intelligence Tools (WSPC-IJAIT)*, **18**(5), 717–738, (2009).

[7] Alfonso Gerevini, Patrik Haslum, Derek Long, Alessandro Saetti, and Yannis Dimopoulos, 'Deterministic planning in the 5th International Planning Competition: PDDL3 and experimental evaluation of the planners', *Artificial Intelligence*, **173**(5-6), 619–668, (2009).

[8] Ghallab M., Nau D., Traverso P., *Automated Planning. Theory and Practice.*, Morgan Kaufmann, 2004.

[9] Yohei Kurata and Tatsunori Hara, 'Ct-planner4: Toward a more user-friendly interactive day-tour planner', in *Information and Communication Technologies in Tourism 2014 - Proceedings of the International Conference in Dublin, Ireland, January 2014*, pp. 73–86, (2014).

[10] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera, 'Personalized tour recommendation based on user interests and points of interest visit durations', in *24th International Joint Conference on Artificial Intelligence, IJCAI*, pp. 1778–1784, (2015).

[11] Antonio Moreno, Aida Valls, David Isern, Lucas Marin, and Joan Borràs, 'Sigtur/e-destination: ontology-based personalized recommendation of tourism and leisure activities', *Engineering Applications of Artificial Intelligence*, **26**(1), 633–651, (2013).

[12] Arturo Montejo Ráez, José M. Perea-Ortega, Miguel Angel García Cumbreras, and Fernando Martínez Santiago, 'Otiŭm: A web based planner for tourism and leisure', *Expert Syst. Appl.*, **38**(8), 10085–10093, (2011).

[13] Paul Resnick and Hal R. Varian, 'Recommender systems', *Communications of the ACM*, **40**(3), 56–58, (1997).

[14] Beatriz Rodríguez, Julián Molina, Fátima Pérez, and Rafael Caballero, 'Interactive design of personalised tourism routes', *Tourism Management*, **33**(4), 926–940, (2012).

[15] Steffen Staab, Hannes Werthner, Francesco Ricci, Alexander Zipf, Ulrike Gretzel, Daniel R. Fesenmaier, Cécile Paris, and Craig A. Knoblock, 'Intelligent systems for tourism', *IEEE Intelligent Systems*, **17**(6), 53–64, (2002).

[16] Pieter Vansteenwegen, Wouter Souffriau, Greet Vanden Berghe, and Dirk Van Oudheusden, 'The city trip planner: An expert system for tourists', *Expert Syst. Appl.*, **38**(6), 6540–6546, (2011).

This page intentionally left blank

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

Author Index

Abdullaev, S. 1754
Abreu, R. 569
Adams, J.A. 1716
Aerts, B. 234
Aghighi, M. 184
Agmon, N. 1493, 1579
Aha, D.W. 1702
Akasiadis, C. 175
Al-Bakri, M. 698
Aleksandrova, M. 1728
Alford, R. 1702
Alomari, M. 1062
Al-Otaibi, R. 1467
An, B. 515
Antoine, N. 133
Arava, R. 1752
Arioua, A. 55
Arour, K. 1678
Atencia, M. 698
Atif, J. 1756
Atkinson, K. 680
Augello, A. 1732
Aziz, H. 787
Baader, F. 1096
Bäckström, C. 184, 716
Bajada, J. 1712
Baker, C.A.B. 1777
Baker, E. 1716
Baroni, P. 489
Bart, A. 613
Bartolini, A. 1598
Bartos, K. 1132
Basile, V. 1458
Baumeister, D. 1656
Becerra-Bonache, L. 1764
Beck, J.C. 1044
Behnke, G. 225
Beierle, C. 1149
Belanche, L.A. 311
Belardinelli, F. 286, 725
Bellodi, E. 1602
Bellotto, N. 107
Ben Amor, N. 202, 1203
Benabbou, N. 1318
Bench-Capon, T. 680, 1664
Benferhat, S. 604
Benini, L. 1598
Bercher, P. 225
Berg, J. 630
Bernstein, A. 1660
Berry, M. 1648
Bertolaso, A. 1720
Bhatnagar, S. 1026, 1690
Bhatt, H.S. 1622
Bhattacharya, S. 1370
Bicego, M. 1256
Bielza, C. 855
Bikakis, A. 1672
Bin, S. 1551
Bisquert, P. 1642
Bistaffa, F. 125
Bit-Monnot, A. 1698
Biundo, S. 225
Black, E. 1736
Blankendaal, R. 1388
Bliem, B. 1105
Bloch, I. 1756
Blockeel, H. 1674, 1764
Bloembergen, D. 1658
Blom, M. 480
Blum, J. 1256
Böhnlein, T. 1656
Bolt, J.H. 646
Bombieri, N. 125
Bordini, R.H. 1708
Borgi, A. 1678
Bosse, T. 1805
Botea, A. 295, 1684
Bottarelli, L. 1256
Bounhas, M. 1662
Bouzeghoub, A. 1678
Boyarski, E. 810
Boyer, A. 1728
Brandán Briones, L. 1718
Bravo-Marquez, F. 498
Brewka, G. 734
Bridi, T. 1598
Brown, K.N. 1291
Brown, M. 1476, 1750
Brun, A. 1728

Buechel, S. 1114
Bühmann, L. 1551
Bulling, N. 751, 1748
Cabrio, E. 1458
Carapelle, C. 1440
Carbonell, J.G. 1620
Cardoso, N. 569
Caridroit, T. 1053
Carvalho, J. 769
Casanova, G. 930
Casini, G. 921
Cavazza, M. 846
Cerutti, F. 966
Ceylan, İ.İ. 1414
Chalkiadakis, G. 175, 1577, 1604
Chander, A. 1775
Chandra, P. 778, 1654
Chang, C. 1559
Chang, L. 1596
Chaplin, J.C. 1449
Charles, F. 846
Chatterjee, K. 1432
Chekol, M.W. 381, 1017
Chen, B. 435
Chen, G. 1569
Chen, H. 1485
Chen, J. 355
Chen, M. 1080
Chen, S. 533
Chen, Y.Q. 304
Cheng, Y. 399, 1626
Chertov, O. 1728
Chesani, F. 1734
Chiarandini, M. 639
Chliaoutakis, A. 1577
Chmait, N. 542
Chocron, P. 871
Christianos, F. 1604
Çilden, E. 1644
Cimiano, P. 1123
Cire, A.A. 1044
Claßen, J. 1265, 1309
Cohn, A.G. 1062, 1769
Coles, A. 1694, 1736
Cooper, M.C. 193, 1563
Coppola, C. 107
Costantini, S. 1680
Cota, G. 1602
Cottone, P. 999
Cranefield, S. 622
Creignou, N. 390
Cristea, A.I. 1158
Croitoru, M. 55, 1642
D'Agostino, M. 141
d'Avila Garcez, A.S. 974
da Rocha Costa, A.C. 1708
Dague, P. 1718
Daly, E. 295
Dao, T.-B.-H. 453, 462
Dastani, M. 751, 886
David, J. 698
Davidson, I. 453
Davis, J. 1239, 1283
De Aldama, R. 1756
De Bock, J. 646
de Campos, T. 216
De Clercq, S. 329
De Cock, M. 329
De Giacomo, G. 408
de Haan, R. 1502
de Kleer, J. 569
de Lima, T. 1053
de Man, J. 1805
De Masellis, R. 1734
de Melo, G. 435
De Natale, F.G.B. 1089
de Nijs, F. 1724
De Raedt, L. 1283
de Silva, L. 1300, 1449
de Weerdt, M.M. 417, 1724
Dechter, R. 1567
Delgrande, J. 671, 1221
DeLoach, S.A. 1606
Demir, A. 1644
Demner-Fushman, D. 1248
Deng, Z. 1760
Di Francescomarino, C. 1734
Dias, G. 1616, 1634
Dias, M. 1632
Dignum, F. 1732
Dignum, V. 3
Dimitrova, V.G. 1769
Do, M. 1698
Döcker, J. 802
Doder, D. 1511
Dorn, B. 802
Dowe, D.L. 542
Dowell, A. 1738
Dragičević, S. 974
Driessens, K. 158
Drissi Oudghiri, S. 1406
Du, X. 99

Dubois, D. 1203
Duckett, T. 107
Duckworth, P. 1062
Dumančić, S. 1674
Duong, K.-C. 453, 462
Dupin de Saint-Cyr, F. 1642
Durand, F. 707
Edgington, P.D. 1397
Efstathiou, I. 1640
Eichhorn, C. 1149
El Khalfi, Z. 202
Endriss, U. 802
Ertel, W. 12
Faber, W. 966
Faltings, B. 1035
Fan, T.-F. 1600
Fan, X. 1573, 1590
Fargier, H. 202
Farinelli, A. 125, 1256, 1720
Fazzinga, B. 1588
Feldman, A. 569
Felli, P. 1449
Felner, A. 810
Feng, K. 435
Feng, Z. 533
Feniger, A. 1610
Feretzakis, G. 1575
Fernández Gil, O. 1096
Ferrari, S. 1634
Finance, B. 1678
Finnestad, S. 1670
Flach, P. 1467
Flesca, S. 1588
Flouris, G. 1672
Fong, K.C.K. 1520
Formisano, A. 1680
Fox, M. v, 1185, 1712
Fraga Pereira, R. 1706
Franc, V. 1132
França, M.V.M. 974
Frank, E. 498
Friedrich, G. 1692
Furfaro, F. 1588
Gaggl, S. 819
Gaglio, S. 999
Gajdoš, P. 1555
Galván, M. 1764
Gams, M. 1803
Ganegedara, T. 577
Gao, L. 507
Garbas, J. 1812
Gasparini, L. 444
Gatsoulis, Y. 1062, 1769
Gatti, N. 560, 1167
Ge, X. 1762
Gentile, M. 1732
Georgala, K. 948
Gerritsen, C. 1805
Ghidini, C. 1734
Gholami, S. 1750
Giacomin, M. 966
Gierse, G. 1265
Gjoreski, H. 1803
Gjoreski, M. 1803
Glass, J. 1688
Glimm, B. 73
Goedemé, T. 234
Goldman, C.V. 595
Gonçalves, R. 957
Gönen, M. 21
Gouider, H. 1203
Gourvès, L. 1423
Governatori, G. 489
Greco, S. 1668
Green, D.G. 542
Grégoire, E. 795
Gribanova, I. 1594
Gu, Y. 1682
Gujar, S. 778, 1035, 1654
Gunawan, A. 1797
Günay, A. 1726
Guns, T. 462
Guo, J. 363
Gutiérrez-Basulto, V. 837
Hachimi, M. 1406
Haddawy, P. 1783
Hafen, R. 1660
Hahn, E.M. 1726
Hahn, U. 1114
Halbersberg, D. 1638
Hammoodi, M. 1549
Hampson, C. 1736
Hao, J. 533, 1344
Haque, M. 1716
Haret, A. 372
Harland, J. 1648
Hasan, A.H.M.I. 1783
Hasanuzzaman, M. 1616, 1634
Haslum, P. 655
Hassan, T. 1812
Hassanzadeh, O. 1581
Hauskrecht, M. 1618

Hawe, G. 1710
Hawes, N. 1458, 1537
Hayashi, N. 1586
He, H. 664
Hecham, A. 1642
Hemaspaandra, E. 1071
Hennes, D. 1658
Hernández-Orallo, J. 471, 1140
Herzig, A. 193, 1563
Hindriks, K.V. 1748
Hogg, D.C. 1062
Höller, D. 225
Hu, H. 664
Hu, R. 355
Huang, F. 399, 1626
Huang, J. 338, 507
Huang, W. 524
Huber, J. 1017
Hug, N. 689
Hunter, A. 150
Ibañez, J. 1818
Ibsen-Jensen, R. 1432
Ignatiev, A. 1327
Indelman, V. 1610
Insa-Cabrera, J. 542
Izzo, D. 1658
Jabbour, S. 913, 991, 1676
Jacquenet, F. 1764
Jameel, S. 1353
Jamroga, W. 1528
Janovsky, P. 1606
Jansen, P.J. 1620
Jardim, D. 1632
Jarraya, A. 1678
Järvisalo, M. 551, 630
Jebbara, S. 1123
Jennings, N.R. 1777
Jensen, F. 743
Jiang, H. 939
Jiang, Y. 1646
Jiang, Y.-G. 167
Jin, C. 399, 1626
Jonker, C.M. 3, 417
Jonsson, A. 1274
Jonsson, P. 37, 184, 716
Joseph, A.G. 1026
Joshi, V. 1212
Jung, J.C. 837
Kajino, H. 1730
Kakas, A.C. 1722
Kalinowski, T. 787
Kalles, D. 1575
Kaminka, G.A. v
Kamps, J. 28
Kanhabua, N. 1158
Karwowski, J. 1746
Kashima, H. 1730
Kask, K. 1567
Keidar, O. 1579
Keküllüoğlu, D. 1608
Kern-Isberner, G. 1149
Keshet, J. 595
KhudaBukhsh, A.R. 1620
Kilicoglu, H. 1248
Kimura, K. 261
King, T.C. 3
Kirkpatrick, D. 1716
Kishimoto, A. 295, 1684
Klima, R. 1658
Knobbout, M. 886, 1636
Knorr, M. 957
Knudsen, A.N. 639
Kochemazov, S. 1594
Koie, Y. 1624
Kökciyan, N. 1608
Kollingbaum, M.J. 444
Komenda, A. 1714
Komusiewicz, C. 895
Konieczny, S. 1053
Koriche, F. 613, 1194
Kottke, D. 586
Kouvaros, P. 1230
Krajník, T. 107
Kraus, S. 320, 595
Kreinovich, V. 604
Krempl, G. 586
Krüger, D. 802
Ktari, R. 390
Kubota, M. 1624
Kudo, M. 261
Kuhr, F. 1583
Kull, M. 1467
Kumar, V. 1370
Kunze, L. 1458
Kurz, S. 880
Kutsch, S. 1149
Kuželka, O. 1239
Lacerda, B. 1537
Lagniez, J.-M. 613, 795
Lakemeyer, G. 1265, 1379
Lakshminarayanan, C. 1690
Lalande, S. 698

Lam, W. 1567
Lamma, E. 1602
Lang, D. 586
Lange, M. 1583
Langseth, H. 743
Larrañaga, P. 855
Larrosa, J. 1567
Larsen, K.S. 639
Latecki, L.J. 1571
Lau, H.C. 347, 1797
Le, T.V. 347
Le Berre, D. 1194
Leake, D. 1176
Lécué, F. 1596
Lee, J.H. 828, 1762
Lehmann, J. 1551
Leite, J. 957
Lejeune, D. 158
Lemon, O. 1640
Lerner, B. 1638
Lesca, J. 1423
Lesire, C. 930
Lespérance, Y. 408
Leung, C. 1590
Levray, A. 604
Li, B. 99, 1646
Li, C.-M. 939
Li, H.-X. 1553
Li, M. 1520
Li, N. 760, 1569, 1571
Li, S. 828
Li, X. 46, 533
Li, Y. 507
Li, Y.-F. 542
Li, Z. 64
Liang, C. 355
Liau, C.-J. 1600
Lin, H.-T. 1362
Lin, H.-X. 1682
Lin, T. 167
Lin, Z. 1710
Linsbichler, T. 252
Liu, J. 304
Liu, L. 1652
Liu, S. 347, 1573, 1590
Liu, T. 99
Liu, T.-Y. 515
Liu, W. 167
Liu, X. 64
Liu, X.A. 304
Liu, Y. 304, 304, 760, 982, 1080, 1726
Lo Re, G. 999
Logan, B. 1449, 1700
Lombardi, M. 1598
Lomuscio, A. 286, 725, 1230
Lonca, E. 1194
Long, D. 1185, 1712
Long, Z. 828
Longuet, D. 1718
Lotinac, D. 1274
Lü, S. 1652
Lu, X. 167
Luengo-Sanchez, S. 855
Lukasiewicz, T. 1008, 1414
Luo, J. 1636
Luštrek, M. 1803
Ma, B. 664
Madalinski, A. 1718
Madsen, A.L. 743
Maffre, F. 193, 1563
Magazzeni, D. 1185
Magee, D.R. 1769
Maida, A.S. 1397
Mailly, J.-G. 734
Maity, R.K. 1690
Malvone, V. 1686
Mańdziuk, J. 1746
Manyà, F. 939
Marinescu, L. 1694
Marinescu, R. 1684
Maris, F. 193, 1563
Marques, T. 1744
Marques-Silva, J. 1327
Marquis, P. 613, 1053, 1194
Martínez, A.M. 743
Martínez-Plumed, F. 1140
Martínez-Usó, A. 1140
Marx, M. 28
Masegosa, A.R. 743
Mathieu, F. 707
Maushagen, C. 277
McBurney, P. 1754
Mehmood, M.O. 1769
Mehta, R. 1476
Meilicke, C. 381, 1017
Mello, P. 1734
Melo, F.S. 863
Meneguzzi, F. 622, 1706
Meng, Z. 1344
Metcalf, K. 1176
Meyer, J.-J. 886, 1636
Meyer, T. 921

Miao, C. 1573, 1590
Micalizio, R. 1650
Michalak, T.P. 90, 1738
Michalewski, H. 1738
Michaliszyn, J. 286
Milano, M. 1598
Mo, P. 760
Modgil, S. 141
Molinero, X. 880
Möller, R. 1583
Molter, H. 895
Montali, M. 1734
Moraitis, P. 1722
Moreira, A. 1740
Morgado, A. 1327
Mrabet, Y. 1248
Mudrova, L. 1537
Mueller, F. 46
Muradore, R. 1720
Murano, A. 1686
Musial, K. 1754
Nagórko, A. 1738
Nakada, T. 1586, 1624
Narahari, Y. 778, 1622, 1654
Narodytska, N. 1704
Nayak, A. 671
Nemirovski, G. 381
Neufeld, E. 1670
Neuss, M. 1309
Ngonga Ngomo, A.-C. 948, 1551
Niedermeier, R. 895
Nielsen, T.D. 743
Niemueller, T. 1265
Niskanen, A. 551
Nitti, D. 1283
Niu, D. 1652
Niu, Z. 1758
Noirie, L. 707
Nolle, A. 381
Nongaillard, A. 1561
Nonoyama, T. 1624
Norman, T.J. 444
Nowé, A. 329
Nunes, L. 1632
Obradovic, Z. 1688
Oliehoek, F.A. 1628
Olsen, M. 880
Onaindia, E. 1818
Ordyniak, S. 1105
Oren, N. 622
Ortolani, M. 999
Osman, N. 1614
Otpuschennikov, I. 1594
Ott, L. 577
Ozaki, A. 837
P., Deepak 1212
Pacheco, J.M. 1592
Padakandla, S. 1690
Padgham, L. 1300
Pagallo, U. 209
Paiva, A. 1592
Paladino, S. 560
Palm, G. 12
Pan, S. 1742
Panisson, A.R. 1708
Papagelis, A. 1575
Papini, O. 390
Parinussa, S. 1388
Parisi, F. 1668
Passerini, A. 1089
Pathak, S. 1610
Patkos, T. 1672
Pearce, A.R. 408
Peñaloza, R. 1414
Percy, C. 974
Pereira, T. 1740
Perny, P. 1318, 1406
Pfahringer, B. 498
Pfandler, A. 372
Picault, S. 1561
Plaat, A. 1791
Plastino, A. 769
Polat, F. 1644
Polevoy, G. 417
Ponti, M. 216
Porteous, J. 846
Postma, E. 1682
Pozzato, G.L. 1650
Prade, H. 689, 1203, 1662
Pralet, C. 930
Predoiu, L. 1008
Provan, G. 524, 1565
Prudêncio, R.B.C. 1140
Pührer, J. 252
Qi, G. 73, 338
Qiao, L. 167
Qin, T. 515
Raddaoui, B. 913
Radecký, M. 1555
Raeissi, M.M. 1720
Raffe, W.L. 46
Rahwan, T. 90, 1738

Ramaithitima, R. 1370
Ramchurn, S. 1777
Ramchurn, S.D. 133
Ramirez, M. 655
Ramoly, N. 1678
Ramos, F. 577
Ramos-López, D. 743
Ratchev, S. 1449
Ravkic, I. 1283
Régnier, P. 193, 1563
Reingold, O. 1704
Ren, C. 1716
Renooij, S. 646
Rens, G. 921, 1557
Renz, J. 1762
Restelli, M. 560
Rey, L. 1656
Riabov, A.V. 1581
Richard, G. 689, 1662
Riguzzi, F. 1602
Riva, M. 216
Rivera-Villicana, J. 1648
Riveret, R. 489
Roberts, M. 1702
Rocco, M. 1167
Roefs, A. 1612
Roessingh, J.J. 1791
Roijers, D.M. 1628
Romain, C. 133
Rong, J. 515
Rosenfeld, A. 320, 595
Rothe, J. 277
Rousset, M.-C. 698
Rovatsos, M. 1744
Roy, S. 1622
Rudolph, S. 819
Sabbadin, R. 116, 202
Sage-Vallier, B. 1769
Saikko, P. 630
Sais, L. 913, 991
Salhi, Y. 913
Salim, M.P. 1616
Salmerón, A. 743
Saltz, M. 311
Samir, A. 133
Sanderson, D. 1449
Sansone, E. 1089
Santos, F.C. 1592
Santos, F.P. 1592
Sardina, S. 1300
Sato, Y. 1586, 1624
Sattar, A. 671
Savani, R. 1658
Savarimuthu, B.T.R. 622
Savaş, E. 1185
Sawada, Y. 1586, 1624
Say, B. 1044
Scala, E. 655
Schaub, T. 1221
Schaudt, O. 1656
Schlenker, A. 1476
Schmid, U. 1812
Schneider, M. 12
Schnoor, H. 1071
Schockaert, S. 329, 1239, 1353
Schorlemmer, M. 871
Schweizer, L. 819
Schwering, C. 1379
Sebastia, L. 1818
Segura-Bedmar, I. 1666
Selker, A.-K. 1656
Semenov, A. 1594
Sen, S. 1344
Serafino, P. 1167
Serrurier, M. 689
Seuss, D. 1812
Shan, H. 426
Shanavas, N. 1710
Sharf, A. 435
Shatkay, H. 1336
Sherif, M.A. 948
Shi, C. 1758
Shi, D. 355
Shivashankar, V. 1702
Shrestha, Y.R. 363
Sierra, C. 1614
Silva, R. 863
Simha, R. 1336
Sinha, A. 1476
Sintov, N. 1750
Sioutis, M. 828
Skibski, O. 1738
Slabaugh, G. 974
Slavkovik, M. 1528
Smith, D.E. 1698
Snášel, V. 1555
Sofka, M. 1132
Sohrabi, S. 1581
Song, H. 1646
Sorge, M. 895
Sorrentino, L. 1686
Spaan, M.T.J. 904, 1724

Spanakis, G. 1612
Spanjaard, O. 1406
Spanoudakis, N.I. 1722
Speranzon, A. 1370
Spiliopoulou, M. 586
Spronck, P. 1791
Srinivasan, R. 1775
Srivastava, S. 1370
Stahl, F. 1549
Ståhlberg, S. 184
Stern, R. 810
Štolba, M. 1714
Strass, H. 252
Stuckenschmidt, H. 381, 1017
Stuckey, P.J. 480
Stutz, P. 1660
Su, H. 64
Su, S. 1559
Suárez-Paniagua, V. 1666
Subramaniam, L.V. 1212
Sun, L. 261, 1569
Sun, P. 1758
Surynek, P. 810
Szczepański, P.L. 90
Sze, W.L. 1616
Tabia, K. 604
Tamassia, M. 46
Tambe, M. 1476, 1750
Tang, Q. 82
Teacy, W.T.L. 1777
Teague, V. 480
Tennant, M. 1549
Teppan, E.C. 1692
Teschner, J. 586
Tessaris, S. 1734
Testerink, B. 751
Thangarajah, J. 243, 1700
Thiaudiere, P. 1769
Thiebaux, S. 655
Thielscher, M. 1630
Thomas, A. 1610
Thomas, D. 1750
Tian, L. 304
Tidhar, R. 480
Toubman, A. 1791
Tožička, J. 1714
Treur, J. 1388
Trovò, F. 560
Turhan, A.-Y. 1440
Tuyls, K. 533, 1658
Udrea, O. 1581
Uher, V. 1555
Ujimoto, K. 1586
Valdes, J. 1769
Vallati, M. 966
van Bevern, R. 895
van den Herik, J. 1682, 1791
van der Zwaan, J.M. 28
van Zee, M. 1511
Varakantham, P. 1752, 1797
Varga, L.Z. 1696
Veloso, M. 863, 1740
Vennekens, J. 234
Ventre, C. 1167
Verman, M. 1660
Verykios, V.S. 1575
Vicol, P. 1221
Vidal, T. 930
Viet, A.-F. 116
Vrain, C. 453, 462
Wahbi, M. 1291
Wallner, J.P. 551
Walraven, E. 904
Walsh, T. 787, 895
Wan, H. 1596
Wang, C.-J. 1485
Wang, H. 1710
Wang, K. 1520
Wang, S. 1760
Wang, W. 1797
Wang, Y. 82, 435, 1553, 1559
Wang, Z. 355, 355, 1569, 1760
Wei, H. 269
Wei, M. 664
Weideveld, L. 1732
Weiss, G. 1612
Weyde, T. 974
Wiggers, A.J. 1628
Wilczynski, A. 1423
Wilder, B. 1750
Wollenberg, J. 1812
Woltran, S. 372, 734, 1105
Wooldridge, M. 90, 1738
Wu, J. 64, 1485
Wu, T. 338
Wu, W. 435
Wu, Z. 1726
Xia, L. 787
Xia, S.-T. 82
Xie, J. 1485
Xiong, L. 982
Xiong, M. 355

Xu, H. 435
Xu, J. 1596
Xu, K. 338
Xue, Y. 1618
Yadav, N. 243
Yakut Kılıç, I.D. 1742
Yamamura, O. 1624
Yang, C. 269
Yang, F. 167
Yang, T. 1344
Yang, Y. 363, 1756
Yao, Y. 1700
Yasutake, M. 1624
Ye, L. 1718
Yeh, C.-K. 1362
Yehoshua, R. 1493
Yokoi, S. 1730
Yolum, P. 1608
Young, J. 1458
Yu, C. 1344
Yu, Q. 269, 1596
Yuan, H. 1520
Zaikin, O. 1594
Zambetta, F. 46, 1648
Zese, R. 1602
Zhang, C. 533
Zhang, G. 304
Zhang, H. 1590
Zhang, J. 426
Zhang, L. 1080, 1485, 1726
Zhang, P. 1762
Zhang, Q. 1758
Zhang, T. 399, 1626
Zhang, Y. 399, 1520, 1626
Zhang, Z. 1559
Zhao, Q. 1080
Zhao, Z. 99
Zhou, S. 507
Zhou, Y. 1158
Zhou, Z. 73
Zhu, J. 1646
Zhuang, Z. 671
Zia, M.A. 1559

This page intentionally left blank