Towards Lifted Maximum Expected Utility*

Marcel Gehrke[1], Tanya Braun[1], Ralf Möller[1],
Alexander Waschkau[2], Christoph Strumann[2], and Jost Steinhäuser[2]

[1] Institute of Information Systems, University of Lübeck, Germany
`{gehrke, braun, moeller}@ifis.uni-luebeck.de`
[2] Institute of Family Medicine, University Medical Center Schleswig-Holstein, Campus Lübeck, Germany
`{alexander.waschkau, christoph.strumann jost.steinhaeuser}@uksh.de`

Abstract. The lifted dynamic junction tree algorithm (LDJT) efficiently answers exact filtering and prediction queries for probabilistic relational temporal models by building and then reusing a first-order cluster representation of a knowledge base for multiple queries and time steps. We extend the underling model of LDJT to provide means to calculate a lifted dynamic solution to the maximum expected utility problem.

1 Introduction

Areas like healthcare and logistics involve probabilistic data with relational and temporal aspects and need efficient exact inference algorithms. These areas involve many objects in relation to each other with changes over time and uncertainties about object existence, attribute value assignments, or relations between objects. More specifically, healthcare systems involve electronic health records (relational part) for many patients (objects), streams of measurements over time (temporal part), and uncertainties [11] due to, e.g., missing information caused by data integration from different hospitals. For query answering, we perform deductive reasoning by computing marginal distributions at discrete time steps. In this paper, we study the problem of exact decision making under uncertainty in large probabilistic temporal models that exhibit symmetries.

We [4] propose parameterised probabilistic dynamic models (PDMs) to represent probabilistic relational temporal behaviour, and furthermore introduce the lifted dynamic junction tree algorithm (LDJT) to answer multiple exact filtering and prediction queries efficiently. LDJT combines the advantages of the interface algorithm [7] and the lifted junction tree algorithm (LJT) [3]. Specifically, this paper contributes action and utility nodes for PDMs to calculate a lifted dynamic solution to the maximum expected utility (MEU) problem. Action nodes are well-motivated candidates to model, e.g., treatments, while utility nodes can represent, e.g., the well being of patients, risk scores, or treatment costs.

Practical related work for inference on relational temporal models consists of approximative approaches. Additional to being approximative, these approaches

* This research originated from the Big Data project being part of Joint Lab 1, funded by Cisco Systems Germany, at the centre COPICOH, University of Lübeck

involve unnecessary groundings or are only designed to handle single queries efficiently. Ahmadi et al. [1] propose lifted (loopy) belief propagation. Geier and Biundo [5] present an online interface algorithm for dynamic Markov logic networks [9], similar to the work of Papai et al. [8]. Vlasselaer et al. [12] introduce an exact approach for relational temporal models involving computing probabilities of each possible interface assignment on a ground level.

Apsel and Brafman [2] show a solution to calculate an exact lifted static solution to the MEU problem. Similar research relates to first-order (PO)MDP [6,10].

The lifting approach exploits symmetries in the model to reduce the number of instances or patients to perform inference on. Additionally, LDJT clusters a model into submodels to answer queries, like the condition of each patient, for a time step and also reuses the clustered structure to answer queries for all time steps $t > 0$. Therefore, LDJT is suitable to handle healthcare related data.

2 Lifted Maximum Expected Utility

We begin by recapitulating PDMs and then extend the work and example from [4] with action and utility nodes to support decision making.

2.1 Parameterised Probabilistic Dynamic Models

A parameterised probabilistic model (PM) combines first-order logic with probabilistic models, representing first-order constructs using logical variables as parameters. Using two PMs, we define the temporal behavior from one time step to the next. Therefore, making PDMs well-suited to model temporal medical processes, which are assumed to be identical for all patients.

Figure 1 shows the example from [4] represented as a PDM with action and utility nodes added in grey. In the example, we remotely infer the condition of patients with regards to water retaining. To determine the condition of patients, we use the change of their weights and additionally use the change of weights of people living with the patient to reduce the uncertainty to infer conditions. An increase in weight could either be caused by overeating or retaining water. In case both gain weight, overeating is more likely, while if only the patient gains weight retaining water is more likely. In Fig. 1, each node is a parameterised random variable (PRV), which is connected by edges to parametric factors (parfactors)

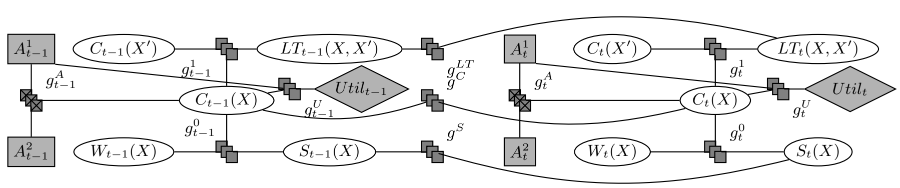

Fig. 1. Retaining water example with action and utility nodes in grey

to set them into relation. For example, the PRV $C(X)$ models the condition of all patients, e.g., $\{alice, bob, eve\}$, and can evaluate to $\{normal$, $deviation$, $retains\ water$, $stopped\}$. Additionally, parfactor g^C models that the condition of a patient influences the condition in the next time step. Using the PDM, LDJT can answer filtering and prediction queries. Hence, LDJT answers queries like "given all weight observation up until now, what is the condition of *bob*" and "how will his condition probably be in ten time steps". For more details, see [4].

2.2 Maximum Expected Utility for PDMs

Let us now extend the example with action and utility nodes. In Fig. 1, one can see two disjoint action nodes (squares) and one utility node (diamond) for each time step. In our example, action A^1 is visit patient and action A^2 is do nothing. Obviously, other actions could also be included in the model, e.g., diet related actions or obtaining a more accurate scale. The condition of patients and A^1 influence the utility. For example, patients with a chronic heart failure might tend to retain water. In case water retention is detected early on, treatment is easier. However, if this water retention remains undetected, water can also retain in the lung, which can lead to a pulmonary edema, making a treatment more costly. More importantly, pulmonary edema is an acute life-threatening condition. In addition to the condition of patients, A^1 also influences the utility as a doctor, with limited time, visiting a patient is expensive. Thus, one always needs to consider that alerting the doctor too early generats unnecessary costs and alerting the doctor too late can have serious consequences for the patient.

We encode the trade-off in the utility node, which normally is time-dependent. Connecting $Util_{t-1}$ and $Util_t$ with a parfactor g^U makes the utility node time-dependent and allows discounting. Actually, a utility PRV can also have parameters, for example $Util_t(X)$ and thereby encode the utility for each patient. For the utility PRV with the parameter, LDJT can maximise the expected utility for each patient, while without the parameter, LDJT maximises a global utility over all patients. The actions to maximise the utility can differ given constraints. Now, we need to select the best action, which maximises the utility. With the action and utility nodes, we can use LDJT to calculate which action maximises the expected utility for the current time step as well as for the one in, e.g., five time steps. Due to the inherent uncertainty of PDMs, which is similar to a belief state in POMDPs, LDJT can only calculate the best action for a finite horizon.

Unfortunately, as all actions have a different impact, currently we cannot combine all actions into one PRV, which would result in $Action(A)$, where $A \in \{a^1, a^2\}$. Nonetheless, LDJT directly reasons over all patients instead of reason over each patient individually. Additionally, LDJT can provide alerts based on observations of each patient. Apsel and Brafman [2] extend C-FOVE to solve MEU queries, which significantly outperforms the propositional case. Braun and Möller [3] show that LJT outperforms GC-FOVE, an extension to C-FOVE, for multiple queries. We [4] present how LDJT efficiently handles temporal aspects and thus, outperforms LJT for temporal queries. Therefore, LDJT is a well-suited candidate to support lifted decision making, including for healthcare processes.

3 Conclusion

We present an extension to PDMs to support lifted decision making by calculating a solution to the MEU problem. Areas like healthcare extremely benefit from the lifting idea for many patients in combination with the efficient handling of temporal aspects of LDJT and the support of different kinds of queries. By extending the underlying model with action and utility nodes, complete healthcare processes including treatments can be modelled. Additionally, by maximising the expected utility, LDJT can calculate the best action.

The next step is to generalise PRVs to allow one action PRV for all possible actions with different impacts and investigate whether, for our application, evidence can reduce the MEU problem from a POMDP to an MDP. Further, we are working on calculating a most probable explanation with LDJT to, e.g., determine the most likely cause for observed changes in the condition of patients.

References

1. Ahmadi, B., Kersting, K., Mladenov, M., Natarajan, S.: Exploiting Symmetries for Scaling Loopy Belief Propagation and Relational Training. Machine learning 92(1), 91–132 (2013)
2. Apsel, U., Brafman, R.I.: Extended Lifted Inference with Joint Formulas. In: Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence. pp. 11–18. AUAI Press (2011)
3. Braun, T., Möller, R.: Lifted Junction Tree Algorithm. In: Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz). pp. 30–42. Springer (2016)
4. Gehrke, M., Braun, T., Möller, R.: Lifted Dynamic Junction Tree Algorithm. In: Proceedings of the 23rd International Conference on Conceptual Structures. Springer (2018), [to appear]
5. Geier, T., Biundo, S.: Approximate Online Inference for Dynamic Markov Logic Networks. In: Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI). pp. 764–768. IEEE (2011)
6. Joshi, S., Kersting, K., Khardon, R.: Generalized First Order Decision Diagrams for First Order Markov Decision Processes. In: IJCAI. pp. 1916–1921 (2009)
7. Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning. Ph.D. thesis, University of California, Berkeley (2002)
8. Papai, T., Kautz, H., Stefankovic, D.: Slice Normalized Dynamic Markov Logic Networks. In: Proceedings of the Advances in Neural Information Processing Systems. pp. 1907–1915 (2012)
9. Richardson, M., Domingos, P.: Markov Logic Networks. Machine learning 62(1), 107–136 (2006)
10. Sanner, S., Kersting, K.: Symbolic Dynamic Programming for First-order POMDPs. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence. pp. 1140–1146. AAAI Press (2010)
11. Theodorsson, E.: Uncertainty in measurement and total error: tools for coping with diagnostic uncertainty. Clinics in laboratory medicine 37(1), 15–34 (2017)
12. Vlasselaer, J., Van den Broeck, G., Kimmig, A., Meert, W., De Raedt, L.: TP-Compilation for Inference in Probabilistic Logic Programs. International Journal of Approximate Reasoning 78, 15–32 (2016)