A Language for Relational Decision Theory

Aniruddh Nath NATH@CS.WASHINGTON.EDU
Pedro Domingos PEDROD@CS.WASHINGTON.EDU
Department of Computer Science and Engineering University of Washington Seattle, WA 98195-2350, U.S.A.

Abstract

In recent years, many representations have been proposed that combine graphical models with aspects of first-order logic, along with learning and inference algorithms for them. However, the problem of extending decision theory to these representations remains largely unaddressed. In this paper, we propose a framework for relational decision theory based on Markov logic, which treats weighted first-order clauses as templates for features of Markov networks. By allowing clauses to have utility weights as well as probability weights, very rich utility functions can be represented. In particular, both classical planning and Markov decision processes are special cases of this framework.

1. Introduction

Intelligent agents must be able to handle the complexity and uncertainty of the real world. First-order logic is useful for the first, and probability for the second. Combining the two has been the focus of much recent research (Getoor & Taskar, 2007). However, there is little work to date on extending these representations to the decision-theoretic setting, which is needed to allow agents to intelligently choose actions. The one major exception is relational reinforcement learning and first-order MDPs. (e.g., Džeroski and De Raedt (1998); van Otterlo (2005); Sanner (2008)). However, the representations and algorithms in these approaches are geared to the problem of sequential decision-making, and many decision-theoretic problems are not sequential. In particular, relational domains often lead to very large and complex decision problems for which no effective general solution is currently available (e.g., influence maximization in social networks, combinatorial auctions with uncertain supplies, etc.).

Presented at *ILP-MLG-SRL*, Leuven, Belgium, 2009.

The goal of this paper is thus to provide a general decision-theoretic extension of first-order probabilistic representations. Our starting point is Markov logic, one of the most powerful representations available (Richardson & Domingos, 2006). We extend it to represent decision-theoretic problems.

2. Background

2.1. Markov Networks and Decision Theory

Graphical models compactly represent the joint distribution of a set of variables $\mathbf{X} = (X_1, X_2, \ldots, X_n) \in \mathcal{X}$ as a product of factors (Pearl, 1988): $P(\mathbf{X}=\mathbf{x}) = \frac{1}{Z}\prod_k \phi_k(\mathbf{x}_k)$, where each factor ϕ_k is a non-negative function of a subset of the variables $\mathbf{x}_k$, and Z is a normalization constant. Under appropriate restrictions, the model is a *Bayesian network* and $Z = 1$. A *Markov network* or *Markov random field* can have arbitrary factors. Graphical models can also be represented in *log-linear form*: $P(\mathbf{X}=\mathbf{x}) = \frac{1}{Z}\exp\left(\sum_i w_i g_i(\mathbf{x})\right)$, where the *features* $g_i(\mathbf{x})$ are arbitrary functions of the state.

A key inference task in graphical models is computing the marginal probabilities of some variables (the query) given the values of some others (the evidence). This problem is #P-complete, but can be solved approximately using algorithms such as *loopy belief propagation* (BP).

An *influence diagram* or *decision network* is a graphical representation of a decision problem (Howard & Matheson, 2005). It consists of a Bayesian network augmented with two types of nodes: *decision* or *action* nodes and *utility* nodes. The action nodes represent the agent's choices; factors involving these nodes and *state* nodes in the Bayesian network represent the (probabilistic) effect of the actions on the world. *Utility* nodes represent the agent's utility function, and are connected to the state nodes that directly influence utility. We can also define a *Markov decision network* as a decision network with a Markov network instead of a Bayesian network.

The fundamental inference problem in decision networks is finding the assignment of values to the action nodes that maximizes the agent's expected utility, possibly conditioned on some evidence. If $\mathbf{a}$ is a choice of actions, $\mathbf{e}$ is the evidence, $\mathbf{x}$ is a state, and $U(\mathbf{x}|\mathbf{a},\mathbf{e})$ is the utility of $\mathbf{x}$ given $\mathbf{a}$ and $\mathbf{e}$, then the *MEU problem* is to compute $\text{argmax}_{\mathbf{a}} E[U(\mathbf{x}|\mathbf{a},\mathbf{e})] = \text{argmax}_{\mathbf{a}} \sum_{\mathbf{x}} P(\mathbf{x}|\mathbf{a},\mathbf{e})U(\mathbf{x}|\mathbf{a},\mathbf{e})$.

2.2. Markov Logic

Markov logic is a probabilistic extension of first-order logic. Formulas in first-order logic are constructed from logical connectives, predicates, constants, variables and functions. A *grounding* of a predicate (or *ground atom*) is a replacement of all its arguments by constants (or, more generally, ground terms). Similarly, a grounding of a formula is a replacement of all its variables by constants. A *possible world* is an assignment of truth values to all possible groundings of predicates.

A *Markov logic network (MLN)* is a set of weighted first-order formulas. Together with a set of constants, it defines a Markov network with one node per ground atom and one feature per ground formula. The weight of a feature is the weight of the first-order formula that originated it. The probability distribution over possible worlds $\mathbf{x}$ specified by the MLN and constants is thus $P(\mathbf{x}) = \frac{1}{Z}\exp(\sum_i w_i n_i(\mathbf{x}))$, where w_i is the weight of the ith formula and $n_i(\mathbf{x})$ its number of true groundings in $\mathbf{x}$.

3. Markov Logic Decision Networks

Decision theory can be incorporated into Markov logic simply by allowing formulas to have utilities as well as weights. This puts the expressiveness of first-order logic at our disposal for defining utility functions, at the cost of very little additional complexity in the language. Let an *action predicate* be a predicate whose groundings correspond to possible actions (choices, decisions) by the agent, and a *state predicate* be any predicate in a standard MLN. Formally:

Definition A *Markov logic decision network (MLDN)* L is a set of triples (F_i, w_i, u_i), where F_i is a formula in first-order logic and w_i and u_i are real numbers. Together with a finite set of constants $C = \{c_1, c_2, \ldots, c_{|C|}\}$, it defines a Markov decision network $M_{L,C}$ as follows:

1. $M_{L,C}$ contains one binary node for each possible grounding of each state and action predicate appearing in L. The value of the node is 1 if the ground atom is true, and 0 otherwise.
2. $M_{L,C}$ contains one feature for each possible grounding of each formula F_i in L for which $w_i \neq 0$. The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is the w_i associated with F_i in L.
3. $M_{L,C}$ contains one utility node for each possible grounding of each formula F_i in L for which $u_i \neq 0$. The value of the node is the utility u_i associated with F_i in L if F_i is true, and 0 otherwise.

We refer to groundings of action predicates as *action atoms*, and groundings of state predicates as *state atoms*. An assignment of truth values to all action atoms is an *action choice*. An assignment of truth values to all state atoms is a *state of the world* or *possible world*. The utility of world $\mathbf{x}$ given action choice $\mathbf{a}$ and evidence $\mathbf{e}$ is $U(\mathbf{x}|\mathbf{a},\mathbf{e}) = \sum_i u_i n_i(\mathbf{x},\mathbf{a},\mathbf{e})$, where n_i is the number of true groundings of F_i. The expected utility of action choice $\mathbf{a}$ given evidence $\mathbf{e}$ is:

$$E[U(\mathbf{x}|\mathbf{a},\mathbf{e})] = \sum_{\mathbf{x}} P(\mathbf{x}|\mathbf{a},\mathbf{e}) \sum_i u_i n_i(\mathbf{x},\mathbf{a},\mathbf{e}) = \sum_i u_i E[n_i]$$

The MEU problem in MLDNs is finding the action choice that maximizes expected utility, and is obviously intractable. We have developed *expanding frontier belief propagation* (Nath & Domingos, 2009), an efficient, approximate MEU algorithm for MLDNs. EFBP searches the space of action choices, computing the expected utility for each one using a variant of BP that only recomputes beliefs for portions of the network that have changed since the last search step. Experiments on multiple domains have shown that EFBP is much faster than running the same algorithm with standard BP at each search step, and produces results of very similar utility.

A wide range of decision problems can be elegantly formulated as MLDNs, including both classical planning and Markov decision processes (MDPs). To represent an MDP as an MLDN, we can define a constant for each state, action and time step, and the predicates $\texttt{State(s!, t)}$ and $\texttt{Action(a!, t)}$, with the obvious meaning. (The ! notation indicates that, for each $\texttt{t}$, exactly one grounding of $\texttt{State(s, t)}$ is true.) The transition function is then represented by the formula $\texttt{State(+s, t)} \wedge \texttt{Action(+a, t)} \Rightarrow \texttt{State(+s', t + 1)}$, with a separate weight for each $(\texttt{s}, \texttt{a}, \texttt{s'})$ triple. (Formulas with + signs before certain variables represent sets of identical formulas with separate weights, one for each combination of groundings of the variables with + signs.) The reward function is defined by the unit clause $\texttt{State}(*\texttt{s},\texttt{t})$, with a utility for each state (using $*$ to represent per-grounding utilities). Policies can be represented by formulas of the form $\texttt{State}(+\texttt{s},\texttt{t}) \Rightarrow \texttt{Action}(+\texttt{a},\texttt{t})$. Infinite-horizon MDPs can be represented using infinite MLNs (Singla & Domingos, 2007). Partially-observable MDPs are represented by adding the observation model: $\texttt{State}(+\texttt{s},\texttt{t}) \Rightarrow \texttt{Observation}(+\texttt{o},\texttt{t})$.

Since classical planning languages are variants of first-order logic, translating problems formulated in these languages into MLDNs is straightforward. For simplicity, suppose the problem has been expressed in satisfiability form (Kautz & Selman, 1992). It suffices then to translate the (first-order) CNF into a deterministic MLN by assigning infinite weight to all clauses, and to assign a positive utility to the formula defining the goal states. MLDNs now offer a path to extend classical planning with uncertain actions, complex utilities, etc., by assigning finite weights and utilities to formulas. (For example, an action with uncertain effects can be represented by assigning a finite weight to the axiom that defines them.) This can be used to represent first-order MDPs in a manner analogous to Boutilier et al. (2001).

4. Application to Viral Marketing

Viral marketing (Domingos & Richardson, 2001) is based on the premise that members of a social network influence each other's purchasing decisions. The goal is then to select the best set of people to market to, such that the overall profit is maximized by propagation of influence through the network. We modeled this problem using the state predicates $\texttt{Buys}(\texttt{x})$ and $\texttt{Trusts}(\texttt{x}_1,\texttt{x}_2)$, and the action predicate $\texttt{MarketTo}(\texttt{x})$. The utility function is represented by the unit clauses $\texttt{Buys}(\texttt{x})$ (with positive utility, representing profits from sales) and $\texttt{MarketTo}(\texttt{x})$ (with negative utility, representing the cost of marketing). The topology of the social network is specified by an evidence database of $\texttt{Trusts}(\texttt{x}_1,\texttt{x}_2)$ atoms.

The core of the model consists of two formulas:

$$\texttt{Buys}(+\texttt{x}_1) \wedge \texttt{Trusts}(+\texttt{x}_2,\texttt{x}_1) \Rightarrow \texttt{Buys}(\texttt{x}_2) \quad (1)$$

$$\texttt{MarketTo}(+\texttt{x}) \Rightarrow \texttt{Buys}(\texttt{x}) \quad (2)$$

In addition, the model includes the unit clause $\texttt{Buys}(\texttt{x})$ with a negative weight, representing the fact that most users do not buy most products. The weight of Formula 1 represents how strongly $\texttt{x}_1$ influences $\texttt{x}_2$, and the weight of Formula 2 represents how strongly users are influenced by marketing. In experiments with this model on a large (75,000 user) real-world dataset, EFBP performed competitively with previous domain-specific solutions using similar models.

5. Conclusion

Markov logic decision networks represent first-order probabilistic decision-theoretic problems by adding utility weights to Markov logic formulas.

Directions for future work include developing other algorithms for MEU inference, including algorithms for specific problems such as planning; using MLDNs for utility-guided learning, including relational reinforcement learning; etc.

Acknowledgements

This research was partly funded by ARO grant W911NF-08-1-0242, DARPA contracts FA8750-05-2-0283, FA8750-07-D-0185, HR0011-06-C-0025, HR0011-07-C-0060 and NBCH-D030010, NSF grants IIS-0534881 and IIS-0803481, and ONR grant N00014-08-1-0670. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, DARPA, NSF, ONR, or the United States Government.

References

Boutilier, C., Reiter, R., & Price, B. (2001). Symbolic dynamic programming for first-order MDPs. *Proc. of IJCAI-01*.

Domingos, P., & Richardson, M. (2001). Mining the network value of customers. *Proc. of KDD-01*.

Džeroski, S., & De Raedt, L. (1998). Relational reinforcement learning. *Proc. of ICML-98*.

Getoor, L., & Taskar, B. (Eds.). (2007). *Introduction to Statistical Relational Learning*. MIT Press.

Howard, R. A., & Matheson, J. E. (2005). Influence diagrams. *Decision Analysis*, *2*, 127–143.

Kautz, H., & Selman, B. (1992). Planning as satisfiability. *Proc. of ECAI-92*.

Nath, A., & Domingos, P. (2009). *Relational decision theory* (Technical Report). University of Washington.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*, 107–136.

Sanner, S. (2008). *First-order decision-theoretic planning in structured relational environments*. Doctoral dissertation, University of Toronto.

Singla, P., & Domingos, P. (2007). Markov logic in infinite domains. *Proc. of UAI-07*.

van Otterlo, M. (2005). *A survey of reinforcement learning in relational domains* (Technical Report). University of Twente.