Learning Statistical Models of Time-Varying Relational Data

Sumit Sanghai, Pedro Domingos and Daniel Weld
Department of Computer Science and Engineering
University of Washington, Seattle, WA 98195

1 Introduction

Formalisms that can represent objects and relations, as opposed to just variables, have a long history in AI. Recently, significant progress has been made in combining them with a principled treatment of uncertainty. In particular, probabilistic relational models or PRMs [4] are an extension of Bayesian networks that allows reasoning with classes, objects and relations. Although PRMs have been successfully applied to a lot of different domains, they lack the temporal dynamics of the real world. In most real world systems, objects get created, modified and even deleted over time. Similarly, the relationships between objects change as time progresses. For example, consider the problem of predicting the set of research topics that become "hot" (e.g., as measured by the number of papers published about them) over time, the changing distribution of these topics among conferences, and the interests and collaborations between authors. It would be difficult to learn a PRM that modeled this time-varying behavior.

Currently the most powerful representation available for capturing sequential phenomena is dynamic Bayesian networks (DBNs) [1], but DBNs are unable to compactly represent many real-world domains that contain multiple objects and classes of objects, as well as multiple kinds of relations among them. DBNs are even more awkward if one wishes to model objects and relations that appear and disappear over time. Thus, our research has focused on a new representation, *dynamic probabilistic relational models* (DPRMs) which combines PRMs with DBNs. Previously, we have explored the problem of efficient inference [8]; this paper outlines our thoughts on learning DPRMs.

2 Dynamic Probabilistic Relational Models

We start by briefly summarizing the definition of PRMs and DPRMs, adapted from [4; 8]. A PRM encodes a probability distribution over the set of all possible instantiations *I* of a schema. In the simplest case, the relational attributes of all objects are assumed to be known, and the PRM specifies a probability distribution for each propositional attribute A of each class C. The parents of each attribute (i.e., the variables it depends on) can be other attributes of C, or attributes of classes that are related to C by some slot chain. Thus, by knowing the relational attributes one can get the joint probability distribution by computing the set of parents for each object and its attributes and calculating the probability through the distribution specified. More generally, only the object skeleton might be known, in which case the PRM also needs to specify a distribution over the relational attributes [5].

Now, we extend PRMs to handle the time domain in the same way that DBNs extend Bayesian networks. Given a relational schema $\mathcal{S}$, we first extend each class C with the relational attribute C.*previous*, with domain C. As before, we initially assume that the relational skeleton at each time slice is known.

Definition 1 A *two-time-slice PRM (2TPRM)* for a relational schema $\mathcal{S}$ is defined as follows. For each class C and each propositional attribute $A \in \mathcal{A}(\mathcal{C})$, we have:

- A *set of parents* $Pa(C.A) = \{Pa_1, Pa_2, ..., Pa_l\}$, where each Pa_i has the form $C.B$ or $f(C.\tau.B)$, where τ is a slot chain containing the attribute *previous* at most once, and $f()$ is an aggregation function.
- A *conditional probability model* for $P(C.A|Pa(C.A))$. $\Box$

Definition 2 A *dynamic probabilistic relational model (DPRM)* for a relational schema $\mathcal{S}$ is a pair $(M_0, M_\rightarrow)$, where M_0 is a PRM over I_0, representing the distribution P_0 over the initial instantiation of $\mathcal{S}$, and $M_\rightarrow$ is a 2TPRM representing the transition distribution $P(I_t|I_{t-1})$ connecting successive instantiations of $\mathcal{S}$. $\Box$

DPRMs are extended to the case where only the object skeleton for each time slice is known in the same way that PRMs are, by adding to Definition 1 a set of parents and conditional probability model for each relational attribute, where the parents can be in the same or the previous time slice. When the object skeleton is not known (e.g., if objects can appear and disappear over time), the 2TPRM includes in addition a Boolean existence variable for each possible object, again with parents from the same or the previous time slice.

3 Inference in DPRMs

Just as a PRM can be expanded into a Bayesian network, so can a DPRM be unrolled into a DBN. In principle, we can then perform inference using particle filtering [2], the most widely used approximate inference algorithm for DBNs. Particle filtering maintains a set of samples (particles) to approximate the distribution of any state; the distribution for next state is achieved by importance sampling and resampling. Unfortunately, for DPRMs, particle filtering is likely to perform poorly, because the state space will be huge. We overcome this by adapting Rao-Blackwellisation [7] to the relational setting. Rao-Blackwellisation divides the state variables into two sets — one in which values are inferred using a particle filter and the other in which values are calculated analytically from the values of the variables in the first set. We make the major assumption that relational attributes do not appear anywhere in the DPRM as parents of unobserved attributes, and that each reference slot can be occupied by at most one object. Then, a Rao-Blackwellised particle is composed of sampled values for all propositional attributes of all objects, plus a probability vector for each relational attribute of each object which is inferred exactly.

While this technique can vastly reduce the size of the state space which particle filtering needs to sample, storing and updating all the requisite probabilities can still become quite expensive. This expense can be ameliorated if context-specific independences exist. We can then replace the vector of probabilities with a novel tree structure whose leaves represent probabilities for entire sets of objects [8].

Our experiments evaluated the efficiency of several inference schemes applied to an assembly-plan execution monitoring task in a simplified manufacturing domain. Even with hundreds of thousands of particles, standard particle filtering failed (i.e. terminated due to inconsistent observations which could not be explained) on datasets with around 100 objects and 500 time steps. In contrast, our inference algorithm yielded accurate predictions on similar problems with only 5000 particles, and ran more quickly and with less storage [8].

Much work remains to improve inference. For example, we will endeavor to lift the assumptions mentioned above and more effectively use a DPRM's structure during inference.

4 Learning in DPRMs

When a DPRM consists of only a single time slice it becomes equivalent to a PRM, and when the DPRM is devoid of relations it is a DBN. Thus we look to combine the learning algorithms already developed for PRMs and DBNs. The first step, parameter learning, appears to be relatively straightforward when no data is missing, since the parameters associated with different types of nodes can be estimated individually. However, there is a subtlety which makes the problem more complex than in a DBN:

- A DPRM can generate a unique state in multiple ways, and each way must be considered during parameter estimation.

For example, if in the new state objects get created, the order of creation can affect the likelihood of the data, as the newly-created objects can interact with each other. There may be a combinatorial number of ways in which a DPRM may generate each state, so we are developing methods to do parameter estimation efficiently. One possibility is to impose a canonical ordering, and another is to greedily compute the most likely order(s) in which the data could have been generated.

In order to learn the DPRM structure, we have to take care of several more issues:

- Defining constraints to eliminate illegal DPRMs is essential when navigating the space of structures. A cycle in a PRM is illegal, and this constraint extends to the two parts of a DPRM. There are additional constraints on a 2TPRM; specifying these in a way that allows creation of an unbounded number of dynamic objects is challenging.
- There are several strategies for searching the space of DPRM structures. The simplest idea is to add and delete edges in the two components, PRM and 2-TPRM, to generate candidate DPRMs. One could do the search by first learning a PRM which gives a good intra-time-slice connectivity, before learning the inter-time-slice connectivity.
- An important task is scoring a DPRM, e.g. with a likelihood-based measure. To compute the likelihood of the data given a candidate DPRM, fast DPRM inference is required. While our particle filtering algorithm is quite fast, we wish to extend it so that we can efficiently explore the space of DPRMS, incrementally updating the likelihood scores. We believe the two-phase search strategy suggested previously will simplify this task.
- Since the space of candidate DPRM models is huge, we are considering pruning mechanisms. Note that some of the methods stated above actually prune the space (e.g. learning the PRM first, followed by time dependencies). One may also impose priors on the models to bias towards simplicity by limiting the number of edges. We plan to design priors over DPRM structures by extending the approach of Heckerman et al.[6] who exponentially penalize arc differences from a ”best” prior structure. We will compare the relative benefits of doing this at the class vs. instance level.
- We plan to extend the learning algorithm to work in the presence of missing values and hidden variables. EM is easiest to apply when the observations are relational but the hidden state is not. Solving this problem with full generality would require an extension of structural EM [3], but this needs to be done for PRMs first.

Acknowledgements

This work was partly supported by an NSF CAREER Award to the second author, by ONR grant N00014-02-1-0932, and by NASA grant NAG 2-1538.

References

[1] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 1989.

[2] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[3] N. Friedman. The Bayesian structural EM algorithm. UAI-98.

[4] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. IJCAI-99.

[5] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. ICML-01.

[6] D. Heckerman, D. Geiger, and D. Chickering. Learning bayesian networks: the combination of knowledge and statistical data. Machine Learning, 1995.

[7] K. Murphy and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In A. Doucet, *et al.*, editors, *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[8] S. Sanghai, P. Domingos, and D. Weld. Dynamic probabilistic relational models. Submitted to IJCAI-03.