

Generalized Ordering-search for Learning Directed Probabilistic Logical Models

Jan Ramon, Tom Croonenborghs, Daan Fierens, Hendrik Blockeel, and Maurice Bruynooghe

K.U.Leuven, Dept. of Computer Science, Celestijnenlaan 200A, 3001 Heverlee, Belgium,
{Jan.Ramon, Tom.Croonenborghs, Daan.Fierens, Hendrik.Blockeel,
Maurice.Bruynooghe}@cs.kuleuven.be

Abstract. Recently, there has been an increasing interest in directed probabilistic logical models and a variety of languages for describing such models has been proposed. Although many authors provide high-level arguments to show that in principle models in their language can be learned from data, most of the proposed learning algorithms have not yet been studied in detail. We introduce an algorithm, generalized ordering-search, to learn both structure and conditional probability distributions (CPDs) of directed probabilistic logical models. The algorithm upgrades the ordering-search algorithm for Bayesian networks. We use relational probability trees as a representation for the CPDs. We present experiments on blocks world domains, a gene domain and the Cora dataset.

1 Introduction

An important class of probabilistic logical models are directed models that are relational extensions of Bayesian networks. A variety of languages for describing such models has been proposed: Probabilistic Relational Models (PRMs) [4], Bayesian Logic Programs (BLPs) [5], Logical Bayesian Networks [2] and many others. Although most authors describe high-level algorithms to learn models in their language or provide arguments to show that such algorithms can be developed, there are still many problems that have not been studied in detail. One such problem is how to deal with recursive dependencies. Consider for example the blocks world [1]: we have a set of blocks that can be stacked on top of each other, the *on/2* predicate is used to represent a certain state. Obviously some (recursive) dependencies hold between all the different *on/2* facts. In undirected models one could state that each *on/2* fact depends on all other *on/2* facts, but in directed models this is not allowed since it would lead to cycles. Hence, learning a directed probabilistic logical model of the blocks world is a challenging problem.

In this paper we introduce an algorithm, generalized ordering-search, to learn both structure and conditional probability distributions of directed probabilistic logical models. Our algorithm is based on the ordering-search algorithm of Teyssier and Koller [8] for learning propositional Bayesian networks. Our contribution is that we upgrade this algorithm to the relational case and investigate the use of relational probability trees [3, 6] as compact and interpretable models of conditional probability distributions (CPDs). We will use the terminology of Logical Bayesian Networks (LBNs) [2] but our discussion applies equally to other directed probabilistic logical models such as PRMs and BLPs. More details can be found in the full paper [7].

2 Generalized Ordering-search

When learning directed models one of the main problems is to avoid cycles. For propositional Bayesian networks (BNs) this can be done by assuming a causal ordering on the set of random variables (RVs). Given such an ordering, we can learn for each RV X a probability tree [3] that identifies which of the preceding RVs are most relevant for X (these will be X 's parents in the BN), and specifies how they influence X (the CPD). However, we usually do not know the optimal ordering. One solution is to do hill-climbing through the space of all orderings and in each step apply the above procedure to learn a BN, until an accurate BN is encountered. This is known as *ordering-search* [8].

Ordering-search cannot be applied directly to the relational case for two reasons. First, we cannot simply learn a separate CPD for each ground RV because we want our model to generalize over the domain (set of constants), i.e. the model should be on the predicate-level instead of on the ground-level. Second, we do not want to learn the optimal ordering on the set of ground RVs but rather a model of this ordering that generalizes over the domain.

To deal with the above two problems, we adapt the original ordering-search algorithm in two ways. First, instead of learning a separate CPD for each RV, we learn for each probabilistic predicate a single so-called *generalized conditional probability function (GCPF)*. Basically, a GCPF for a predicate p is a function that takes as input any set of RVs (the possible parents) plus a target RV T built from p and that returns a set of parents for T as well as a CPD for T given these parents. Second, instead of searching through the space of orderings until a good ordering is found, we now search through the space of orderings only to collect information about the likelihood of different orderings (using the learned GCPFs) and afterwards we use this information to learn a model of the optimal ordering. Note that an ordering on the RVs together with an appropriate set of GCPFs fully specifies an LBN. We now briefly explain the two steps of our algorithm in more detail.

In the first step we learn for each probabilistic predicate a GCPF under the form of a logical probability tree [3, 6]. Let \mathcal{R} denote the set of all ground RVs. More precisely, for each predicate p we learn a tree \mathcal{T}_{gcpf} such that for any target $T \in \mathcal{R}$ built from p , any assignment v_T to T , any set $\mathcal{E} \subseteq \mathcal{R}$ of known evidence RVs and any assignment $v_{\mathcal{E}}$ to the RVs in \mathcal{E} , $\mathcal{T}_{gcpf}(T, v_T, \mathcal{E}, v_{\mathcal{E}}, \mathcal{R}) = \hat{P}(T = v_T \mid \mathcal{E} = v_{\mathcal{E}})$ expresses the probability that T has value v_T given the values of all RVs in \mathcal{E} . To learn such a tree we first generate training data by sampling data from random orderings (i.e. randomly selecting T and \mathcal{E}) and then apply the TILDE system adapted to learn probability trees [3] on this dataset. The final result of this step is a set of GCPFs that can be used together with any ordering to determine an LBN.

In the second step we learn a model of the optimal ordering. Concretely, we use TILDE to learn a logical decision tree that specifies for any pair of RVs their best ordering. The training data for this regression tree is generated by searching through the space of orderings and computing for each considered ordering the difference in likelihood when swapping two adjacent RVs in this ordering (this difference in likelihood is the target of the regression). To compute the likelihood of an ordering we use the GCPFs of the previous step. Using the obtained regression tree we can determine the

optimal ordering for any pair of RVs and hence indirectly also for any number of RVs. This optimal ordering together with the GCPFs fully determines an LBN.

3 Experiments

We evaluated our approach on four datasets: a simple artificial dataset about the inheritance of genes, two datasets derived from relational reinforcement learning experiments on the blocks world [1] and the Cora dataset. We compared the results of generalized ordering search with the results obtained when learning CPDs for a given expert ordering and for a random ordering.

The main conclusion from our experiments is that generalized ordering-search succeeds in learning an ordering that performs approximately equally well (in terms of likelihood) as the expert orderings and considerably better than random orderings. Also, the complexity (size) of the learned CPDs for generalized ordering search is similar to that for the expert orderings and smaller than for random orderings.

Acknowledgements

TC and DF are supported by the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT Vlaanderen). JR and HB are post-doctoral fellows of the Fund for Scientific Research (FWO) of Flanders.

References

1. T. Croonenborghs, J. Ramon, H. Blockeel, and M. Bruynooghe. Online learning and exploiting relational models in reinforcement learning. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007.
2. D. Fierens, H. Blockeel, M. Bruynooghe, and J. Ramon. Logical Bayesian networks and their relation to other probabilistic logical models. In *Proceedings of the 15th International Conference on Inductive Logic Programming*, pages 121–135. Springer, 2005.
3. D. Fierens, J. Ramon, H. Blockeel, and M. Bruynooghe. A comparison of approaches for learning probability trees. In *Proceedings of 16th European Conference on Machine Learning, Porto, Portugal*, pages 556–563, 2005.
4. L. Getoor, N. Friedman, D. Koller, and A. Pfeffer. Learning Probabilistic Relational Models. In S. Dzeroski and N. Lavrac, editors, *Relational Data Mining*, pages 307–334. Springer-Verlag, 2001.
5. K. Kersting and L. De Raedt. Towards combining inductive logic programming and Bayesian networks. *Proceedings of the 11th International Conference on Inductive Logic Programming*, pages 118–131. Springer-Verlag, 2001.
6. J. Neville and D. Jensen. Dependency networks for relational data. In *Proceedings of the 4th IEEE International Conference on Data Mining*, 2004.
7. J. Ramon, T. Croonenborghs, D. Fierens, H. Blockeel, and M. Bruynooghe. Generalized ordering-search for learning directed probabilistic logical models. *Machine Learning (special issue ILP'06)*, 2007. Conditionally accepted.
8. M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI)*, pages 584–590, 2005.