Learning Directed Relational Models With Recursive Dependencies

Oliver Schulte*, Hassan Khosravi, and Tong Man
oschulte@cs.sfu.ca, hkhosrav@cs.sfu.ca, mantong01@gmail.com

School of Computing Science
Simon Fraser University
Vancouver-Burnaby, B.C., Canada

Abstract. Recently, there has been an increasing interest in generative relational models that represent probabilistic patterns over both links and attributes. A key characteristic of relational data is that the value of a predicate often depends on values of the same predicate for related entities. In this paper we present a new approach to learning directed relational models which utilizes two key concepts: a pseudo likelihood measure that is well defined for recursive dependencies, and the notion of stratification from logic programming. An issue for modelling recursive dependencies with Bayes nets are redundant edges that increase the complexity of learning. We propose a new normal form for 1st-order Bayes nets that removes the redundancy, and prove that assuming stratification, the normal form constraints involve no loss of modelling power. We incorporate these constraints in the learn-and-join algorithm of Khosravi *et al.*, which is a state-of-the art structure learning algorithm that upgrades propositional Bayes net learners for relational data. Emprical evaluation compares our approach to learning recursive dependencies with undirected models (Markov Logic Networks). The Bayes net approach is orders of magnitude faster, and learns more recursive dependencies, which lead to more accurate predictions.

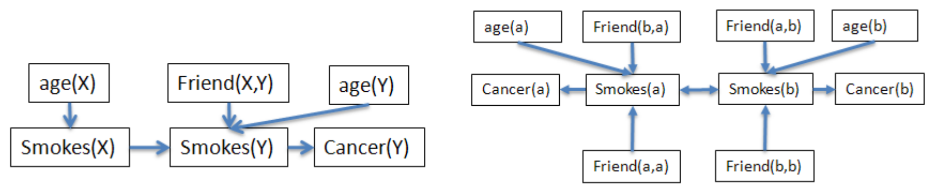
1 Introduction

Relational data are ubiquitous in real-world applications, ranging from social network analysis to enterprise databases. A key phenomenon that distinguishes relational data from single-population data is that the value of an attribute for an entity can be predicted by the value of the same attribute for related entities. For example, whether individual a smokes may be predicted by the smoking habits of a's friends. This pattern can be represented by clausal notation such as $Smokes(X) \leftarrow Smokes(Y), Friend(X, Y)$. In logic programming terminology, this is an example of a *recursive dependency*, where a predicate depends on itself.

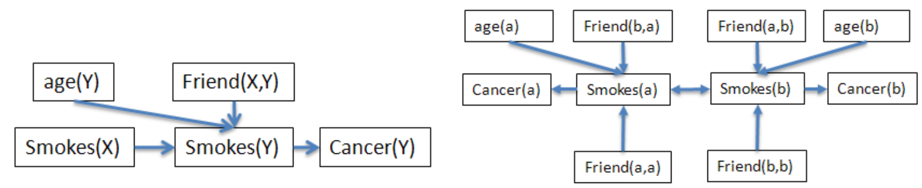
In this paper we investigate a new approach to learning recursive dependencies with Bayes nets, specifically Poole's Parametrized Bayes Nets (PBNs) [3]; however,

* Supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada. We are indebted to reviewers of the ILP conference for helpful comments.

our results apply to other directed relational models as well. The learn-and-join algorithm is a state-of-the art method for learning Bayes nets for relational data [1]. Its objective function is a pseudo-likelihood measure that is well defined for Bayes nets that include recursive dependencies [4]. A problem that we observed in research with datasets that feature recursive dependencies is that the repetition of predicates causes additional complexity in learning if each predicate instance is treated as a separate random variable. For example, suppose that the dependence of smoking on itself is represented in a Bayes net with a 3-node structure $\mathit{Smokes}(X) \rightarrow \mathit{Smokes}(Y) \leftarrow \mathit{Friend}(X, Y)$. Now suppose that we also include an attribute *age* for a person. Then a Bayes net learner would potentially consider two edges, $\mathit{age}(X) \rightarrow \mathit{Smokes}(X)$ and $\mathit{age}(Y) \rightarrow \mathit{Smokes}(Y)$, as shown in Figure 1(a). If there is in fact a statistical dependence of smoking on age, then each of these edges correctly represents this dependency, but one of them is redundant, as the logical variables X, Y are interchangeable placeholders for the same domain of entities.



(a) Left: A stratified Bayes net that is *not* in main functor format, because $\mathit{Smokes}(X)$ and $\mathit{Smokes}(Y)$ both have parents. Right: The ground graph for two individuals a and b.



(b) Left: A Bayes net in main functor format where $\mathit{Smokes}(Y)$ is the main node for the predicate *Smokes*. Right: The ground graph is the same as the ground graph for the Bayes net of Figure 1(a).

Approach. We propose a normal form for Parametrized Bayes nets that eliminates such redundancies: For each function/predicate symbol, designate one node as the *main node*. Then constrain the Bayes net such that only main nodes have edges pointing into them. In the example above, if $\mathit{Smokes}(Y)$ is the main functor for *Smokes*, the edge $\mathit{age}(X) \rightarrow \mathit{Smokes}(X)$ is forbidden, as shown in Figure 1(b). Notice that in terms of ground instances, the two Bayes nets have exactly the same ground graph. We prove that this observation holds in general, and therefore the main node constraint incurs no loss of expressive power: if a Bayes net B is stratified, then there is a Bayes net B' in main functor format such that B and B' induce the same ground graph for every relational database instance. A 1st-order Bayes net is stratified if there is an ordering of predicates, such that for each edge, either the predicate of the parent precedes that of the child in the ordering, or is the same.

Evaluation. We show how the learn-and-join algorithm can be extended to learn recursive dependencies using the main functor constraint. We use 3 performance metrics: Runtime, Accuracy (ACC), and Conditional log likelihood (CLL). The CLL of a ground atom in a database is its log-probability given the information in the database. Accuracy is evaluated using the most likely value for a ground atom. The measures we report are averages over all attribute predicates using cross validation. We compared the learn-and-join algorithm with two state-of-the-art Markov Logic Network (MLN) methods (LHL and LSM [2]) on two datasets. The MBN method of Khosravi *et al.* [1] applies MLN inference with the Bayes net model. Table 1 shows the result for a synthetic dataset and Table 2 shows the results for a real life dataset, Mondial. *Neither of the Markov Logic methods LHL nor LSM discovered any recursive dependencies.* In contrast, the learn-and-join algorithm discovered the dependencies displayed in Table 1 using clausal notation. The predictive accuracy of MBN using the recursive dependencies was much better (average accuracy improved by 25% or more).

	MBN	LSM	LHL
Time (seconds)	12	**1**	2941
Accuracy	**0.85**	0.44	0.47
CLL	**-0.8**	-2.21	-4.68

Table 1. Results on synthetic data.

	MBN	LSM	LHL
Time (seconds)	50	**2**	15323
Accuracy	**0.50**	0.26	26
CLL	**-1.05**	-1.43	-3.69

Table 2. Results on Mondial.

Database	Recursive Dependency Discovered
University	$gpa(X) \leftarrow Friend(X,Z), gpa(Z), ranking(X), grade(X,Y), registered(X,Y)$
University	$coffee(X) \leftarrow coffee(Y), Friend(X,Y)$
Mondial	$religion(X) \leftarrow Border(X,Y), religion(Y), continent(X)$
Mondial	$continent(X) \leftarrow Border(X,Y), continent(Y), religion(Y), gdp(X)$

Table 3. Recursive dependencies discovered by the learn-and-join algorithm.

References

1. Hassan Khosravi, Oliver Schulte andTong Man, Xiaoyuan Xu, and Bahareh Bina. Structure learning for Markov logic networks with many descriptive attributes. In *AAAI'10*, pages 487–493, 2010.
2. Stanley Kok and Pedro Domingos. Learning markov logic networks using structural motifs. In *ICML'10*, pages 551–558. 2010.
3. David Poole. First-order probabilistic inference. In Georg Gottlob and Toby Walsh, editors, *IJCAI'03*, pages 985–991. Morgan Kaufmann, 2003.
4. Oliver Schulte. A tractable pseudo-likelihood function for Bayes Nets applied to relational datasets. In *SIAM SDM'11*, pages 462–473, 2011.