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 models that represent probabilistic patterns over both links and attributes. An effective structure learning technique has been to upgrade propositional Bayes net learning for generative relational models. A problem with applying Bayes net methods to learn recursive rules is that they lead to cyclic dependencies, which violates the acyclicity constraint of Bayes nets. 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. Emprirical evaluation compares our approach to learning recursive dependencies with undirected models (Markov Logic Networks).

1 Introduction

An important research topic for learning with relational data is the discovery of generative models that represent probabilistic patterns over both links and attributes. An effective structure learning technique has been to upgrade propositional Bayes net learning for generative relational models [8, 4, 14, 5, 7]. In a first-order model, the same predicate may have different instances with different logical variables. This expressive power allows the model to elegantly represent *recursive dependencies* using Horn clauses. For example, whether person a smokes may be predicted by the smoking habits of a's friends, represented by the rule $Smokes(X) \leftarrow Smokes(Y), Friend(X, Y)$, which expresses a recursive dependency of the *Smokes* predicate on itself. We address three key difficulties for learning recursive dependencies. (1) The repetition of predicates causes additional complexity in learning if each predicate instance is treated as a separate random variable since they behave the same statistically (e.g., $Smokes(X)$ and $Smokes(Y)$). (2) A well-known difficulty for Bayes net learners is that recursive dependencies lead to cyclic dependencies among ground facts [14, 2, 17]. The cycles make it difficult to define a model likelihood function for observed ground facts, which is an essential component of model selection techniques. (3) A related problem is that defining valid probabilistic inferences in cyclic models is difficult. This paper describes a new approach to learning directed graphical generative models of relational data that include recursive dependencies.

Approach. We employ Parametrized Bayes nets (PBN) [13] because they are a relatively straightforward extension of Bayes nets for relational data. Our approach and results apply to other directed relational models as well. (1) To address duplicate predicates, we propose a *new normal form* for stratified PBNs that adds

constraints on edges for different instances of the same predicate. A Parametrized Bayes Net is stratified if there is some ordering of the predicates such that the predicate that appears in a parent node is the same as that in the child node or comes before it in the ordering. We provide a theorem that for stratified PBNs, the normal form restriction involves no loss of expressive power.

(2) To define a model likelihood function for Bayes net search, we utilize the recent relational Bayes net pseudo likelihood measure [15]. The recent efficient learn-and-join algorithm [8] searches for models that maximize the pseudo likelihood. We show how the learn-and-join algorithm can be adapted to exploit the normal form constraints for efficient structure learning.

(3) For inference, to avoid cycles in the ground model, we convert our learned Bayes nets to an undirected model using the standard moralization procedure. Markov Logic Networks (MLNs) are a prominent relational model class that can learn and reason with recursive dependencies [2]. The moralization approach approach combines the efficiency and scalability of Bayes net learning with the high-quality inference procedures of MLNs.

In our test datasets, our structure learning method is orders of magnitude faster than state-of-the art MLN learners, and the learned models provide substantially more accurate predictions.

Paper Organization. We review the relevant background and define our notation. We describe the normal form extension of the learn-and-join algorithm. We then evaluate the ability of the extended algorithm to learn autocorrelations, compared to Markov Logic Network learner.

Contributions. The main contributions may be summarized as follows. (1) A new formal form theorem for Parametrized Bayes nets that addresses redundancies in modelling autocorrelations. (2) An extension of the learn-and-join algorithm for learning Bayes nets that include autocorrelations.

2 Related Work.

Adaptations of Bayes net learning methods for relational data are presented in [8, 4, 14, 5, 7]. Our algorithm is an extension of the learn-and-join method [8], which is a lattice-search that maximizes the relational Bayes net pseudo-likelihood score [15]. Issues connected to learning Bayes nets with recursive dependencies are discussed in detail in [14]. Early work on this topic required additional constraints to ensure the acyclicity of the ground Bayes net model [5, 7]. The generalized order-search of Ramon *et al.* instead resolves cycles by learning an ordering of ground atoms. A basic difference between this paper and generalized order search is that we focus on issues at the *predicate level*. Our algorithm can be combined with generalized order-search as follows: First use our algorithm to learn a Bayes net structure at the predicate/class level. Second carry out a search for a good ordering of the ground atoms. We leave this option for future work. The moralization approach of performing inference by converting a directed model to an undirected one is due to [8].

Stratification is a widely imposed condition on logic programs, because it increases the tractability of reasoning in an LP with a relatively small loss of expressive power. Our definition is essentially the same as the definition of local stratification in logic programming [1]. The difference is that that levels are assigned to predicates/functions rather than ground literals, so the definition does not need to distinguish positive from negative literals. In the case where every

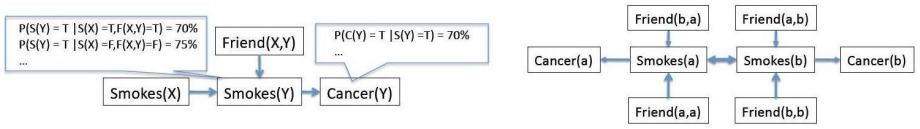

Fig. 1. A Parametrized Bayes Net and its grounding for two individuals a and b. The double arrow $\leftrightarrow$ is equivalent to two directed edges.

predicate appears in only one node in the PBN graph, stratification is equivalent to disallowing cycles in the graph. Related ordering constraints appear in the statistical-relational literature [3, 5].

3 Background and Notation

Parametrized Bayes nets are a basic statistical-relational model; we follow the original presentation of Poole [13]. A **functor** is a function symbol or a predicate symbol. In this paper we discuss only functors with a finite domain of possible values. A **parametrized random variable** or **functor node** is of the form $f(X_1, \ldots, X_k) = f(\mathbf{X})$ where f is a functor and each first-order variable X_i is of the appropriate type for the functor. We assume that the variables X_i are distinct. A **Bayes net structure** is a directed acyclic graph (DAG) G, whose nodes comprise a set of random variables. A Bayes net (BN) is a BN structure with conditional probability parameters. A **Parametrized Bayes Net** is a Bayes net whose nodes are functor nodes. A **ground** PBN $\overline{B}$ is derived from B by instantiating the variables in the functor nodes in B with all possible constants. Figure 1 shows a PBN and its grounding. A **level mapping** assigns to each functor f in a PBN B a nonnegative integer $level(f)$. B is **stratified** if there is a level mapping such that for every edge $f(\mathbf{X}) \rightarrow g(\mathbf{Y})$, we have $level(f) \leq level(g)$.

The (natural) join of two tables is a new table that contains the rows in the Cartesian products of the tables whose values match on common fields. A table join corresponds to logical conjunction.

Markov Logic Networks are presented in detail by Domingos and Richardson [2]. An MLN is a set of weighted first-order formulas. Bayes net DAGs can be converted into MLN structures through the standard **moralization** method [2, 12.5.3]: connect all spouses that share a common child, and make all edges in the resulting graph undirected. For each assignment of values to a child and its parents, add a formula to the MLN.

4 Stratification and the Main Functor Node Format

The functor concept allows different nodes in a PBN to be associated with the same attribute or relationship, where the difference between the nodes is in their variable arguments only. This expressive power is essential to represent recursive dependencies where instances of an attribute/relationship depend on other instances of the same attribute/relationship. However, it causes additional complexity in learning if each functor is treated as a separate random variables. Consider for example the PBN shown in Figure 2 left. If we treat $Smokes(X)$ and $Smokes(Y)$

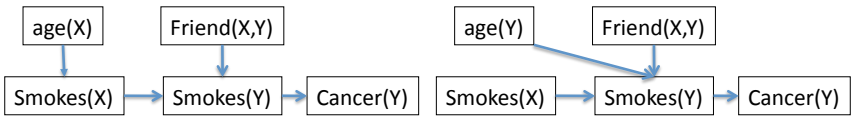

Fig. 2. Two Bayes nets with different predictors for $Smokes(X)$ and $Smokes(Y)$.

as entirely separate variables, BN learning needs to consider additional edges like $Smokes(X) \rightarrow Cancer(X)$. However, this edge is redundant because the 1st-order variables X and Y are interchangeable as they refer to the same entity set. Redundant edges can be avoided if we restrict the model class to the main functor format, where for each function symbol f (including relationships), there is a *main functor node* $f(\mathbf{X})$ such that all other functor nodes $f(\mathbf{Y})$ associated with the same functor are sources in the graph, that is, they have no parents. The intuition for this restriction is that statistically, two functors with the same function symbol are equivalent, so it suffices to model the distribution of these functors conditional on a set of parents just once. This leads to the following formal definition.

Definition 1. *A PBN B is in* ***main functor node form*** *if for every functor f of B, there is a distinguished functor node $f(\mathbf{X})$, called the* ***main functor node*** *for f, such that every other functor node $f(\mathbf{Y})$, where $\mathbf{X} \neq \mathbf{Y}$, has no parents in B.*

The PBN of Figure 1 is in main functor form. The main functors are $Friend(X, Y)$ for the relationship predicate $Friend$, and $Smokes(Y)$ for the function symbol $Smokes$, and $Cancer(Y)$ for the function symbol $Cancer$. The PBN of Figure 2 (left) is not in main functor form because we have two functor nodes for $Smokes$ with nonzero indegree. Our main theorem states that a stratified PBN B can be transformed into an equivalent PBN B' that is main functor format. For instance, in the PBN of Figure 2 left we can first substitute the edge $age(X) \rightarrow Smokes(X)$ for the edge $age(Y) \rightarrow Smokes(Y)$ as in Figure 2 right. In terms of ground instances, the two PBNs have exactly the same ground graph. We also require that the result of this transformation must be an acyclic PBN; this will be the case with stratified PBNs, which are defined as follows.

Proposition 1. *Let B be a stratified PBN. Then there is a PBN B' in main functor form such that for every database $\mathcal{D}$, the ground graph $\overline{B}$ is the same as the ground graph $\overline{B'}$.*

We extend the learn-and-join algorithm [8] with the main functor restriction to accommodate autocorrelations. The learn-and-join algorithm upgrades a single-table BN learner for relational learning. The basic idea of the learn-and-join algorithm is that join tables should inherit edges from their subjoins. To illustrate in the PBN of Figure 1, applying the single-table BN learner to the *People* table may produce a single-edge graph $Smokes \rightarrow Cancer$. Then we apply the Bayes net learner to the self-join table $J = People \bowtie Friend \bowtie Friend$, with the constraint that the edge $Smokes \rightarrow Cancer$ must be included.

A theoretical foundation for the algorithm is the recent relational Bayes net pseudo likelihood measure [15].The pseudo log-likelihood of a PBN is the expected

	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.

log-likelihood of a *randomly selected grounding*. This score is well-defined even with recursive dependencies, because it invokes a *random* instantiation of the PBN rather than a *complete* instantiation with all known individuals. Schulte provides a closed form for computing the pseudo log-likelihood znc shows that the learn-and-join algorithm (implicitly) maximizes the pseudo-likelihood [15]. Given a specification of main functor nodes, it is straightforward to extend the learn-and-join algorithm by adding as a constraint that only edges pointing into main functor nodes are allowed. The full paper will contain a complete description with pseudo code.

5 Evaluation

Systems Compared and Metrics. To perform inference, we convert the learned PBN structure to a Markov Logic structure via moralization, and then apply MLN parameter learning [8]; this is the MBN method. For each dataset, we evaluated the learning methods with a 5-fold cross-validation scheme. We compare our method to the two most recent MLN structure learning algorithms, LHL [9] and LSM [10]. We use 4 performance metrics: Number of Clauses or Parameters, Runtime, Accuracy (ACC), and Conditional log likelihood (CLL) [12, 9]. ACC and CLL have been used in previous studies of MLN learning. The CLL of a ground atom in a database given an MLN is its log-probability given the MLN and the information in the database. Accuracy is evaluated using the most likely value for a ground atom. For ACC and CLL the values we report are averages over all predicates.

Experiments on synthetic data. We manually created a small dataset (about 1000 tuples) for a University domain [5], including a Friendship relation among students. The dataset features a strong autocorrelation for the ranking of friends and for the coffee habits of friends. Table 1 shows the results.

Experiments on real world data. We use the *Mondial Database.* This dataset contains data from multiple geographical web data sources [11]. We followed the modification of [16], and used a subset of the tables and features. Our dataset contains 4 entity tables, *Country*, *Continent*, *Economy*, *Government*, where the latter three are related to Country by many-one relationships, and one relationship table *Borders* that relates two countries. Table 2 shows the results.

Both MBN and LSM are fast. The speed of LSM is due to the fact that its rules are mostly just the unit clauses that model marginal probabilities (e.g., $intelligence(S, I)$) [2]. *Neither LHL nor LSM discovered recursive dependencies.* In contrast, the learn-and-join algorithm discovered the following dependencies which we display using clausal notation (like Bayesian clauses [7]) in Table 5.

6 Conclusion

An effective structure learning approach has been to upgrade propositional Bayes net learning for generative relational models. We presented a new method for applying

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

Table 3. Dependencies discovered by the extension of the learn-and-join algorithm.

Bayes net learning for *recursive dependencies* based on a recent pseudo-likelihood score and a new normal form theorem. The pseudo-likelihood score quantifies the fit of a recursive dependency model to relational data, and allows us to efficiently apply model search algorithms. The proposed normal form eliminates potential redundancies that arise when predicates are duplicated to capture recursive relationships. In evaluations our structure learning method was very efficient and found recursive dependencies that were missed by Markov logic structure learning methods.

References

1. Krzysztof R. Apt and Marc Bezem. Acyclic programs. *New Generation Comput.*, 9(3/4):335–364, 1991.
2. Pedro Domingos and Matthew Richardson. Markov logic: A unifying framework for statistical relational learning. In *Introduction to Statistical Relational Learning* [6].
3. Daan Fierens. On the relationship between logical bayesian networks and probabilistic logic programming based on the distribution semantics. In *ILP*, 2009.
4. Daan Fierens and et al. Learning directed probabilistic logical models: Ordering-search versus structure-search. In *ECML*, 2007.
5. Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *In IJCAI*, pages 1300–1309. Springer-Verlag, 1999.
6. Lise Getoor and Ben Tasker. *Introduction to statistical relational learning.* MIT Press, 2007.
7. Kristian Kersting and Luc de Raedt. Bayesian logic programming: Theory and tool. In *Introduction to Statistical Relational Learning* [6], chapter 10, pages 291–318.
8. Hassan Khosravi and et al. Structure learning for Markov logic networks with many descriptive attributes. In *Conference on Artificial Intelligence (AAAI).*, 2010.
9. Stanley Kok and Pedro Domingos. Learning markov logic network structure via hypergraph lifting. In *ICML*, pages 64–71, 2009.
10. Stanley Kok and Pedro Domingos. Learning markov logic networks using structural motifs. In *ICML'10*, pages 551–558. 2010.
11. Wolfgang May. Information extraction and integration: The mondial case study. Technical report, Universit?at Freiburg, Institut f?ur Informatik, 1999.
12. Lilyana Mihalkova and Raymond J. Mooney. Bottom-up learning of Markov logic network structure. In *ICML*, pages 625–632. ACM, 2007.
13. David Poole. First-order probabilistic inference. In Georg Gottlob and Toby Walsh, editors, *IJCAI*, pages 985–991. Morgan Kaufmann, 2003.
14. Jan Ramon and et al. Generalized ordering-search for learning directed probabilistic logical models. *Machine Learning*, 70(2-3):169–188, 2008.
15. Oliver Schulte. A tractable pseudo-likelihood function for Bayes Nets applied to relational data. In *SIAM SDM*, pages 462–473, 2011.
16. Rong She, Ke Wang, and Yabo Xu. Pushing feature selection ahead of join. 2005.
17. Benjamin Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *UAI*, 2002.