Modelling Relational Statistics With Bayes Nets

Oliver Schulte, Hassan Khosravi, Arthur Kirkpatrick, Tong Man, Tianxiang Gao, and Yuke Zhu

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

Abstract. Learning statistical patterns based on relational frequencies is a machine learning task that supports applications like strategic planning and query optimization. Parametrized Bayes nets (PBNs) are a 1st-order logic extension of Bayes nets for representing statistical patterns in relational data. The standard grounding semantics for PBNs is not appropriate for answering frequency queries, because such queries concern generic events, not individual ground facts. We propose a new frequency semantics for PBNs, that is based on Halpern's classic domain frequency semantics for probabilistic 1st-order logic [**?**]. A suitable objective function for learning parameters for answering frequency queries is the recent relational BN pseudo-likelihood measure [**?**], which is based on random instantiations of 1st-order variables, like Halpern's semantics. The pseudo-likelihood maxima are the observed empirical frequencies in the relational data structure. A naive computation of database frequencies is intractable due to the complexity imposed by negated relational links. We render this computation tractable by using the fast Mobius transform. Evaluation on four benchmark datasets shows that maximum pseudo-likelihood provides accurate estimates at different sample sizes.

1 Introduction

Many real-world applications store data in relational format, with different tables for entities and their links. One of the machine learning problems that arise from relational data is *frequency estimation*: building a model that can answer queries about the rates of generic events in the database [**?**]. For example, a frequency query for a movie customer database may be "what is the percentage of friends who are both women"? A model of relational frequencies can be used for several applications.

Statistical 1st-order Patterns. AI research into combining 1st-order logic and probability investigated in depth the representation of statistical patterns in relational structures, based on relational frequencies [**?**, **?**]. Often such patterns can be expressed as generic statements, like "intelligent students tend to take difficult courses".

Policy making and strategic planning. A university administrator may wish to know which program characteristics attract high-ranking students in general, rather than predict the rank of a specific student in a specific program.

Maier *et al.* [**?**] describe several applications of causal-relational knowledge for decision making. These causal relations reflect correlations defined by relational frequencies.

Query optimization is an application where a statistical model predicts a probability for given table join conditions that can be used to infer the size of the join result [**?**]. Estimating join sizes (selectivity estimation) is used to minimize the size of intermediate join tables [**?**].

Approach We focus on building a Bayes net model for relational frequency, utilizing Poole's Parametrized Bayes nets (PBNs) [**?**]. The nodes in a PBN are constructed with functors and 1st-order variables (e.g., $genre(M)$ may be a node). The PBN semantics proposed by Poole is a grounding semantics where the 1st-order Bayes net is instantiated with all possible groundings to obtain a directed graph whose nodes are functor with constants (e.g., $genre(scream)$). The ground graph can be used to answer queries *about individuals*, such as "if user 10 has 30 friends, of whom 10 are women, what is the probability that user 10 is a woman"? However, the ground graph is not appropriate for answering queries *about frequencies* because these are about generic rates and percentages, not about any particular individuals. We propose a new semantics for Parametrized Bayes nets that supports frequency queries. The semantics is based on Halpern's classic domain frequency semantics for probabilistic 1st-order logic. This semantics views statements with 1st-order variables as expressing a frequency statement. For instance, the claim ""the percentage of friends who are both women is 40%" could be expressed by the 1st-order formula

$$P(Gender(X) = female, Gender(Y) = female | Friend(X, Y)) = 40\%.$$

Formally, the domain frequency semantics views a 1st-order variable as randomly selecting a member of its domain [**?**, **?**]. The probability of a 1st-order statement is the probability that it is satisfied by independent random instantiations of its variables. We show that the random selection idea can be applied to assign a frequency semantics to 1st-order queries over the nodes in a Parametrized Bayes net. While we focus on PBNs, the random selection semantics can be applied to any statistical-relational model whose syntax is based on 1st-order logic.

Getoor introduced statistical-relational models (SRM) as a Bayes net type representation of relational frequencies [**?**]. To our knowledge, ours is the first work that evaluates a relational model other than SRMs for frequency estimation.

Learning The state-of-the-art structure learning method for PBNs is the learn-and-join algorithm of Khosravi *et al.* [**?**]. In this paper we study parameter learning for frequency modelling. A standard Bayes net parameter learning method is maximum likelihood estimation. A relational likelihood function for Bayes nets is difficult to define because of cyclic data dependencies. We propose to utilize a recent relational pseudo-likelihood measure for Bayes nets [**?**] that is well defined even in the presence of cyclic dependencies. This measure matches the frequency semantics well because it is also based on the concept of random instantiations: The pseudo log-likelihood is the expected log-likelihood of a random instantiation of the 1st-order variables in the PBN. The pseudo-likelihood can be used as an objective function for estimating Bayes net parameters from relational data. In this paper we consider the parameters that maximize the pseudo likelihood, abbreviated as MPLE. The maximization problem has a closed-form solution: the MPLE parameters are the empirical frequencies, as with classical i.i.d. maximum likelihood estimation. The paper examines the accuracy and tractability of MPLE. The main computational challenge is to compute database statistics that involve negated relationships. Enumerating the complement (negation) of a relationship table is computationally infeasible. We show that the fast Möbius transform makes MPLE tractable, even in the case of negated relationships. The Möbius transform has been applied for computing Dempster-Schaeffer belief updates [**?**]. Ours is the first application in relational learning.

Results We evaluate MPLE on four benchmark real-world datasets. On complete-population samples MPLE achieves near perfect accuracy in parameter estimates, and excellent performance on Bayes net queries. The accuracy of MPLE parameter values is high even on medium-size samples.

Contributions Our main contributions for frequency modelling in relational data are the following:

1. A new frequency semantics for graphical models based on 1st-order logic, derived from Halpern's random selection semantics for probabilistic 1st-order logic.
2. Making the computation of frequency estimates tractable by computing database statistics using the fast Möbius transform.
3. Evaluating the empirical accuracy of the Bayes net frequency models at medium to large sample sizes.

While this paper focuses on Bayes net frequency models, the fast Möbius transform is a general procedure for computing relational statistics that involve negated links. This problem arises also in learning Probabilistic Relational Models with (link) existence uncertainty [**?**, Sec.5.8.4.2]. Getoor *et al.* use a "1-minus" technique that avoids enumerating the complement of a *single* relation. The FMT achieves this for an arbitrary number of relations, which has been an open problem in statistical-relational learning. Other application areas include multi-relational data mining and ILP models with clauses contain negated relationships.

Paper Organization We review background and notation in the next section. Section 3 presents s theoretical results on the consistency and asymptotic efficiency. Section 4 presents the fast Möbius transform for relational data. Simulation results are presented in Section 5, showing the runtime cost of estimating parameters, and evaluations of their quality by (a) inference on random queries, and (b) comparison with the true population parameter values.

2 Related Work

Type 1 and Type 2 Relational Probabilities. Classic AI research established a fundamental distinction between two types of probabilities associated with a relational structure [**?**, **?**]. Type 1 probabilities are assigned to the rates, statistics, or frequencies of events in a database. Type 2 probabilities are assigned to specific, non-repeatable events or the properties of specific entities; we may refer to them as *instance probabilities*. Syntactically, type 1 probabilities are assigned to formulas that contain 1st-order variables (e.g., $P(\mathit{Flies}(X)|\mathit{Bird}(X) = \mathit{90}\%$, or "birds fly" with probability 90%), whereas instance probabilities are assigned to formulas that contain constants only (e.g., $P(\mathit{Flies}(\mathit{tweety}) = \mathit{90}\%)$. There has been much AI research on using Bayes nets for representing and reasoning both with class frequencies [**?**] and instance probabilities [**?**]. Most statistical-relational learning has been concerned with type 2 instance probabilities; models like Probabilistic Relational Models (PRMs) [**?**] and Markov Logic Networks (MLNs) [**?**] define probabilities for ground instances using a grounding semantics. To our knowledge, Statistical Relational Models (SRMs) [**?**] are the only statistical model class with an explicit semantics for representing database frequencies, prior to our proposed semantics for PBNs.

SRMs and PBNs. SRMs differ from PBNs and other statistical-relational models in several respects. (1) The SRM syntax is not that of first-order logic, but is derived from a tuple semantics [**?**, Def.6.3]. The semantics of SRMs is that tuples are sampled independently from different tables, and a Boolean join indicator variable takes on the value true if the tuples join (i.e., agree on the primary key fields) [**?**, Def.6.3]. This is different from the random selection semantics we propose for PBNs. (2) The expressive power of SRMs is less than that of PBNs. A relevant difference for our work is that SRMs cannot express general combinations of positive and negative relationships [**?**, Def.6.11]. While this avoids the computational difficulties associated with negated relationships, it limits the expressive power of SRMs. For discussion of further limitations see [**?**, Ch.6]. (3) With respect to learning, the published learning algorithm for SRMs uses information gain as a model selection score, not a pseudo-likelihood [**?**]. Conditional database frequencies were used by analogy with the relational case, but there was no theorem establishing these as the score maxima. A direct empirical comparison is difficult as code has not been released (Getoor, personal communication). The study by [**?**] used all the data for both testing and training, rather than considering increasing sample sizes. In this setting, SRMs achieved good average accuracy on the task of join selectivity estimation for random queries; the accuracy of parameter estimates was not considered.

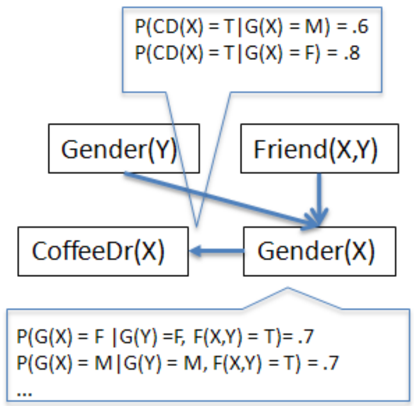
Unified Learning For Type 1 and Type 2 Probabilities. For *inference*, connections between relational frequencies and instance probabilities have been a major subject in AI research. For example, Halpern [**?**] showed that any type 2 inference model can be used for type 1 as follows. Introduce a new individual constant for each 1st-order variable in the relational model (e.g., random-student, random-course, and random-prof). Applying instance probability inference to these new individuals provides a query answer that can be interpreted as a generic frequency. To illustrate, if the only thing we know about Tweety is that Tweety is a bird, then the probability that Tweety flies should be the frequency of flyers in the bird population. In statistical terminology, marginal probabilities assigned to a ground atom should reflect population probabilities. We apply Halpern's method to perform frequency inference with Markov Logic Networks and compare it with Bayes net inference; we leave for future work comparisons with other instance probability models, such as PRMs.

We believe that a unified approach to learning for both relational probability types is an exciting research direction for statistical-relational learning. This paper contributes to unification in two ways. (1) We show how one and the same 1st-order model can be used for both type 1 and type 2 inferences. Depending on the type of query asked, one needs to use a grounding semantics for type 2 queries about ground instances, and a frequency semantics for type 1 queries about frequencies. (2) Previous work has used the random selection pseudo-likelihood to learn structures that perform well for type 2 queries [**?**, **?**]. In this paper we provide evidence that the same likelihood measure learns structures and parameter values that make accurate predictions for type 1 queries. Thus the same objective function is suitable for learning models for both types of queries.

3 Background: Parametrized Bayes Nets

Our work combines concepts from relational databases and graphical models. As much as possible, we use standard notation in these different areas. Parametrized Bayes nets are a basic learning model for relations; we follow the original presentation of Poole [**?**]. We assume familiarity with Bayes nets and concepts such as CP-table and I-map [**?**]. A **functor** is a function symbol or a predicate symbol. Each functor has a set of values (constants) called the **range** of the functor. To conform to statistical terminology, Poole refers to 1st-order variables as population variables. A **population variable** X is associated with a population, which is a set of individuals, corresponding to a type or domain in logic. A **functor random variable** or **functor node** is of the form $f(X_1, \ldots, X_k)$. A **Parametrized Bayes Net** is a Bayes net whose nodes are functor random variables. Figure **??** shows a PBN. The syntax of PBNs is similar to that of other directed relational graphical models (cf. [**?**]).

The functor formalism is rich enough to represent an entity-relationship schema via the following translation: Entity sets correspond to populations, descriptive attributes to functors, relationship tables to Boolean functors, and foreign key constraints to type constraints on the arguments of relationship predicates. Figure **??** shows a simple relational database instance [**?**].



People

Name	Gender	Coffee Drinker
Anna	F	T
Bob	M	F

Friend

Name1	Name2
Anna	Bob
Bob	Anna

Fig. 1. An illustrative Parametrized Bayes Net. $Friend(X, Y)$ is a relationship node, the other three nodes are attribute nodes.

People

Name	Gender	Coffee Drinker
Anna	F	T
Bob	M	F

Friend

Name1	Name2
Anna	Bob
Bob	Anna

Fig. 2. A simple relational database instance.

An **instantiation** or **grounding** for a set of variables $X_1, \ldots, X_k$ assigns a constant c_i from the population of X_i to each variable X_i.

4 Frequency Semantics for Parametrized Bayes Nets

For a single population, a distribution over population members induces a joint distribution over their attributes (e.g., age, height, gender). Classic AI research generalized the concept of single population frequencies to 1st-order logic using the idea of a *random selection* [**?**, **?**]. We provide a brief review in the context of a functor language. For example, consider a probabilistic 1st-order statement using the obvious abbreviations for the BN of Figure **??**:

$$P(Friend(X, Y) = T, Gender(X) = M, Gender(Y) = F) = 1/4. \qquad (1)$$

which assigns a probability to a sentence with free 1st-order variables.[1] To evaluate whether the sentence is true in a given database/interpretation $\mathcal{D}$, the

[1] The full syntax distinguishes between free variables and variables with a probabilistic interpretation.

random selection semantics assumes a distribution over the population/domain associated with each free 1st-order variable. Assuming the independence of these distributions, we obtain a joint distribution over the values of population variables $X_1, X_2, \dots, X_k$; that is, a joint distribution over tuples of individuals. The domain frequency of a 1st-order statement is then the sum over all tuples that satisfy the statement, weighted by the probability of each tuple. The statement is true in a database if it assigns the domain frequency correct for the database.

In learning, an observed database instance $\mathcal{D}$ provides data only for a subpopulation. We define the observed **database frequency**, denoted by $P_{\mathcal{D}}$, of a functor node assignment in a relational database to be the number of instantiations of the population variables in the functor nodes that satisfy the assignment in the database, divided by the number of all possible instantiations. The database frequency is the special case of Halpern's domain frequency with a uniform distribution over all observed population members. For example, the probability statement (**??**) is true in the database of Figure **??** given a uniform distribution over users.

The random selection concept provides a type 1 semantics for Parametrized Bayes nets: if we view 1st-order variables $X_1, X_2, \dots, X_k$ as independent random variables that each sample an individual, then a functor of the form $f(X_1, X_2, \dots, X_k)$ represents a funcion of a random k-tuple. Since a function of a random variable is itself a random variable, this shows how we can view functor nodes containing 1st-order variables as random variables in their own right, without grounding the variables first. For example, using the obvious abbreviations for the BN of Figure **??**, the semantics of a joint assignment like

$$P(F(X,Y) = T, G(X) = M, G(Y) = M, CD(X) = T) = 10\%$$

is "if we randomly select two users X and Y, there is a 10% chance that they are friends, both are men, and one is a coffee drinker".

5 Review: Pseudo-Likelihood for Parametrized Bayes Nets

Schulte [**?**, **?**] proposed a way to measure the fit of a Bayes net model to relational data that matches the random selection semantics: the idea is to consider a *random* grounding of the 1st-order variables in the Parametrized Bayes net, rather than a complete grounding. The pseudo log-likelihood is defined as follows.

1. Randomly select a grounding for *all* 1st-order variables that occur in the Bayes Net. The result is a ground graph with as many nodes as the original Bayes net.
2. Look up the value assigned to each ground node in the database. Compute the log-likelihood of this joint assignment using the usual product formula; this defines a log-likelihood for the random instantiation.
3. The expected value of this log-likelihood is the *pseudo log-likelihood* of the database given the Bayes net.

For illustration, Table **??** provides a sample computation of the random selection log-likelihood, where we have used the obvious abbreviations for functors.

As the example illustrates, the pseudo-likelihood is well-defined even in the presence of an autocorrelation dependency of Gender on itself.

X	Y	F(X,Y)	G(X)	CD(X)	G(Y)	BN probability	BN log-p
Anna	Bob	T	F	T	M	$0.5^2 \cdot 0.3 \cdot 0.8 = 0.06$	-2.81
Bob	Anna	T	M	F	F	$0.5^2 \cdot 0.3 \cdot 0.6 = 0.24$	-3.10
Anna	Anna	F	F	T	F	$0.5^2 \cdot 0.5 \cdot 0.8 = 0.26$	-2.30
Bob	Bob	F	M	F	M	$0.5^2 \cdot 0.5 \cdot 0.6 = 0.11$	-2.59

Table 1. An example computation of the pseudo-likelihood for the database of Figure **??** and the Bayes net of Figure **??**. (Unspecified BN parameters are chosen as uniform solely for computational convenience.) The pseudo log-likelihood is -2.7, the average of the log-probabilities (rightmost column).

Schulte [**?**] proves the following result, which shows that the pseudo-likelihood maxima (MPLE) are the observed empirical frequencies, analogous to the maximum likelihood estimates for i.i.d. data.

Proposition 1. *For a Bayes net structure and database $\mathcal{D}$, the parameter values that maximize the pseudo-likelihood are the conditional empirical frequencies defined by database distribution $P_{\mathcal{D}}$.*

In the remainder of the paper we consider the properties of MPLE parameter estimates. We begin with a procedure for computing them.

6 Computing Relational Frequencies

Initial work in SRL modelled the distribution of descriptive attributes given knowledge of existing links. Database statistics conditional on the *presence* of one or more relationships can be computed by table joins with SQL. More recent models represent *uncertainty about relationships* with link indicator variables. For instance, a Parametrized Bayes net includes relationship indicator variables such as $Reg(S, C)$. Learning with link uncertainty requires computing sufficient statistics that involve the *absence* of relationships. A naive approach would explicitly construct new data tables that enumerate tuples of objects that are *not* related. However, the number of unrelated tupes is too large to make this scalable (think about the number of user pairs who are *not* friends on Facebook). Can we instead reduce the computation of sufficient statistics that involve negated relationships to the computation of sufficient statistics that involve existing (positive) relationships only? The classic Möbius parametrization for binary variables provides an affirmative answer [**?**, p.239]. Consider a set $b_1, \ldots, b_m$ of binary variables, where all marginal probabilities are available that involve only positive values. Thus we have available probabilities such as $P(b_1 = 1)$; $P(b_1 = 1, b_2 = 1)$; $P(b_1 = 1, b_3 = 1, b_k = 1)$; etc. These joint probabilities are the **Möbius parameters** of the joint distribution. The Möbius inversion theorem entails that *all* joint probabilities, involving any number of 0 values, can be computed as an alternating sum of the Möbius parameters. We can apply this result for MPLE as follows. Consider a PBN family containing m relationship nodes. We wish to compute frequencies of the joint family assignments, from which conditional probabilities are easily derived. The Möbius inversion theorem entails that each joint frequency *can be computed from joint frequencies that involve existing relationships only.*

The **fast Möbius transform** (FMT) is an optimal algorithm for converting the Möbius parameters to a complete set of joint probabilities [**?**]. The FMT was originally described using category theory with lattice structures. Our version is adapted for **joint probability tables** (JP-tables). A JP-table is just like a CP-table whose rows correspond to joint probabilities rather than conditional probabilities. To represent a Möbius parameter, we allow relationship nodes to take on the value $*$ for "unspecified". For instance, suppose that the family nodes are $\mathit{Int}(S)$, $\mathit{Reg}(S, C)$, $\mathit{RA}(S, P)$. Then the Möbius parameter $P(\mathit{Int}(S) = \mathit{1})$ is stored in the row where $\mathit{Int}(S) = \mathit{1}, \mathit{Registered}(S, C) = *, \mathit{RA}(S, P) = *$. The FMT uses a local update operation corresponding to the simple probabilistic identity

$$P(\sigma, \mathbf{R}, R = F) := P(\sigma, \mathbf{R}) - P(\sigma, \mathbf{R}, R = T) \tag{2}$$

where σ is an attribute condition that does not involve relationships and $\mathbf{R}$ specifies values for a list of relationship nodes. This shows how a probability that involves $k + 1$ false relationships can be computed from two probabilities that each involve only k false relationships, for $k \geq 0$. The FMT initializes the JP-table with the Möbius parameters without negated relationships, that is, all relationship nodes have the value T or $*$. It then goes through the relationship nodes $R_1, \ldots, R_m$ in order, replaces at stage i all occurrences of $R_i = *$ with $R_i = F$, and applies the local update equation for the probability value for the modified row. At termination, all $*$ values have been replaced by F and the JP-table specifies all joint frequencies. Algorithm **??** gives pseudocode and Figure **??** illustrates the FMT in a schematic example with two relationship nodes.

Complexity Analysis. Kennes and Smets provide a thorough theoretical analysis of the FMT. We summarize the main points, for more details see [**?**]. (1) The key property of the FMT is that *it accesses data only about existing links*, never about nonexisting links. The number of additions performed by the FMT is $m2^{m-1}$. A lower bound argument shows that this is optimal [**?**, Cor.1]. (2) Without a bound on m, computing sufficient statistics in a relational structure is #P-complete [**?**, Prop.12.4]. In practice, the number m of relationship nodes is small, 4 or less in a typical relational model. (3) Kennes and Smets describe an "obvious algorithm" that applies the local update to each row in the JP-table. The obvious algorithm also uses only existing links, but requires $O(3^m)$ additions. [2]

[2] The obvious algorithm, but not the FMT, was rediscovered by Khosravi *et al.* and presented as a conference poster at ILP 2009. This work was not included in the proceedings or any other archival publication.

Algorithm 1 The fast Möbuis transform for parameter estimation in a Parametrized Bayes Net.

Input: database $\mathcal{D}$; a set of functor nodes divided into attribute nodes $A_1, \ldots, A_j$ and relationship nodes $R_1, \ldots, R_m$.
Output: Joint Probability specifying the data frequencies for each joint assignment to the input functor nodes.

```
for all attribute value assignments A_1 := a_1, ..., A_j := a_j do
    initialize the JP-table with the Möbius parameters: set all relationship nodes to
    either T or *; find joint frequencies with data queries.
    for i = 1 to m do
        Change all occurrences of R_i = * to R_i = F.
        Update the joint frequencies using (??).
    end for
end for
```

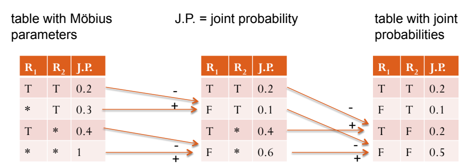


Fig. 3. The fast Möbius transform with $m = 2$ relationship nodes. For simplicity we omit attribute conditions.

7 Evaluation

All experiments were done on a QUAD CPU Q6700 with a 2.66GHz CPU and 8GB of RAM. We evaluated the algorithm on real-world datasets. The datasets and our code are available on the Web [pointer omitted for blind review].

7.1 Datasets

We used four benchmark real-world databases, with the modifications by [**?**], which contains details and references.

Dataset	#tuples
Mondial	814
Hepatitis	12447
Financial	17912
Movielens	82623

Table 2. Size of datasets in number of table tuples.

MovieLens. A dataset from the UC Irvine machine learning repository.
Financial A dataset from the PKDD 1999 cup.

Hepatitis Database. A modified version of the PKDD'02 Discovery Challenge database.

Mondial Database. A geography database. Mondial features a self-relationship, *Borders*, that indicates which countries border each other.

To obtain a Bayes net structure for each dataset, we applied the learn-and-join algorithm to each database [**?**]. This is the state-of-the-art structure learning algorithm for PBNs; for an objective function, it uses the pseudo-likelihood described in this paper. We also conducted experiments with synthetic graphs and datasets. The results are similar to those on real-life datasets. We omit details for lack of space.

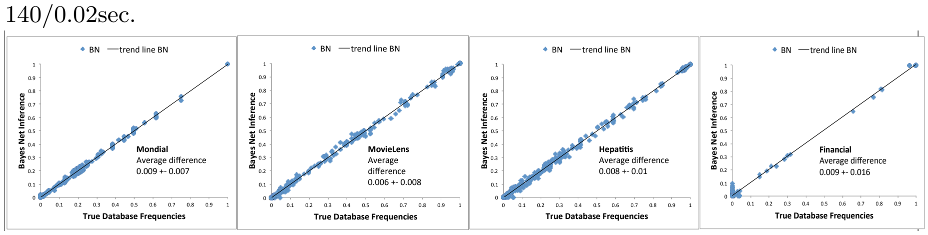
7.2 Learning Times

Figure **??** shows the runtimes for computing parameter values. The first method uses the fast Möbius transform to compute the conditional probabilities, the second Complement method uses SQL queries that explicitly construct tables for the complement of relationships, i.e., tables that contain tuples of unrelated entities.

Table 3. Learning time results (sec) for the fast Möbius transform vs. constructing complement tables.

Database	Parameters	Complement	FMT	C/FMT
Mondial	1618	157	7	22
Hepatitis	1987	18,246	77	237
Financial	10926	228,114	14,821	15
MovieLens	326	2,070	50	41

Fig. 4. Query Performance: Estimated vs. true probability. The average error and standard deviation are shown as well. Number of queries/average inference time per query: Mondial, 506/0.08sec; MovieLens, 546/0.05sec; Hepatitis, 489/0.1sec; Financial, 140/0.02sec.



7.3 Inference

The basic inference task for Bayes nets is answering probabilistic queries. If the given Bayes net structure is an I-map of the true distribution, then correct parameter values lead to correct predictions. Thus the performance on queries has been used to evaluate parameter learning in several studies [**?**]. We randomly generate queries for each dataset according to the following procedure. First, randomly choose a target node V 100 times, and go through each possible value a of V such that $P(V = a)$ is the probability to be predicted. For each value a, choose randomly the number k of conditioning variables, ranging from 1 to 3. Make a random selection of k variables $V_1, \ldots, V_k$ and corresponding values $a_1, \ldots, a_k$. The query to be answered is then $P(V = a|V_1 = a_1, \ldots, V_k = a_k)$.

As in [**?**], we evaluate queries after learning parameter values on the entire database. Thus the BN is viewed as a statistical summary of the data rather than generalizing from a sample. BN inference is carried out using the Approximate Updater in CMU's Tetrad program. Figure **??** shows the query performance for each database. A point (x, y) on a curve indicates that there is a query such that the true probability value in the database is x and the probability value estimated by the model is y.

The Bayes net inference is close to the ideal identity line, with an average error of less than 1%.

Comparison With Markov Logic Networks Although most statistical-relational models were designed for instance-level probabilities rather than frequency queries, Halpern's method described in Section **??** allows us to apply an instance-level inference model for frequency estimates. To benchmark our results, we compare PBN inferences with Markov Logic Network (MLN) frequency estimates. In graphical terms, MLNs can be viewed as defining an undirected relational model. They are a good comparison point because (1) they are currently one of the most active areas of SRL research; the Alchemy system provides open-source, state-of-the-art learning and inference software [**?**]. (2) They do not require the specification of further components (e.g., a combining rule or aggregation function). (3) An undirected model can accommodate recursive relationships (cyclic dependencies). We compare the following learning algorithms [**?**]. We use the MC-SAT algorithm for MLN inference as implemented in Alchemy.

PBN Bayes net parametrized with maximum pseudo likelihood estimates.

MBN+Neg The Bayes net structure is converted to an MLN using the standard moralization procedure. The weights of clauses are learned using Alchemy's default weight learning procedure [**?**].

LHL The LHL algorithm is a structure learning algorithm that produces a parametrized MLN.

LSM The state-of-the-art LSM structure learning algorithm that produces a parametrized MLN. In experiments by Kok and Domingos, LSM outperformed other MLN learners.

The MBN method (for "Moralized Bayes Net"), was used by Khosravi *et al.* [**?**, **?**, **?**], but *without clauses involving negated links.* To define a complete joint distribution for frequency querying, it is necessary to add clauses with negated links. Unfortunately, Alchemy fails to terminate on any of our datasets because of the computational challenges that arise from many clauses with negated links. This is evidence for the usefulness of the Fast Möbius Transform. To obtain comparison results, we used state-of-the art MLN structure learning algorithms as well as the moralized structure. An advantage of this approach is that the structures are learned with Alchemy clause weight estimation as a subroutine, to they are optimized for Alchemy parameter learning.

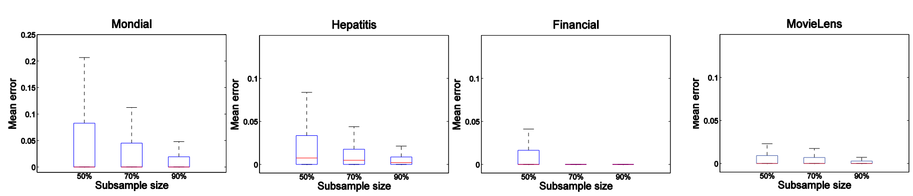
The Bayes net models provide much more accurate frequency estimates than the MLN models, with an average improvement of 10% or more. This shows that even with complete data, learning an accurate model of relational frequencies is not a trival task.

	Average error			
Dataset	JBN	MLN(LSM)	MLN(LHL)	MBN+Neg
Mondial	**0.9**%	8.6%	10.5%	NT
Hepatitis	**0.8**%	11.2%	13.2%	NT
Financial	**0.9**%	9.1%	NT	NT
Movielens	**0.6**%	14.2%	NT	NT

Table 4. The frequency query performance of Bayes nets vs. Markov Logic Networks. We show the average absolute error over all random queries between the predicted frequency and the true database frequencies. NT denotes non-termination within the system resources.

7.4 Conditional Probabilities

Fig. 5. Error (absolute difference) in conditional probability estimates. Median (red center line) and spread of error in the estimates of conditional probability parameters, averaged over 10 random subdatabases and all parameters in a given BN.



To study parameter estimation directly at different sample sizes, we performed a set of experiments to train the model on $N\%$ of the data and test on the other $(100 - N)\%$ of the data. Conceptually, we treated each benchmark database as specifying an entire population, and then estimated the complete-population frequencies from partial-population data. A fractional sample size parameter is uniform across tables and databases. We employed standard subgraph subsampling [**?**, **?**], which selects entities from each entity table uniformly at random and restricts the relationship tuples in each subdatabase to those that involve only the selected entities. Subgraph sampling matches the random selection semantics which is based on random draws from a population. It is applicable when the observations include positive and negative link information (e.g., not listing two countries as neighbors implies that they are not neighbors). The subgraph method satisfies an ergodic law of large numbers in the sense that as the subsample size increases, the subsample relational frequencies approach the population relational frequencies.

With increasing sample size, MPLE estimates approach the true value in all cases. Even for the smaller sample sizes, the median error is close to 0, confirming that most estimates are very close to correct. As the box plots show, the 3rd error quartile of estimates is bound within 10% on Mondial, the worst case, and within less than 5% on the other datasets.

8 Conclusion

This paper considered parameter learning for Parametrized Bayes nets that model database statistics. Our approach was to use the maxima of a recent pseudo-likelihood function as estimates, which are the empirical frequencies. The fast Möbius transform makes the computation of database frequencies feasible even when the frequencies involve negated links. Theoretically, the maximum pseudo-likelihood estimates approach the true conditional probabilities as observations increase. Experimentally, the fit is good even for medium data sizes. Overall, our results indicate that Parametrized Bayes nets together with maximum pseudo-likelihood estimates provide an accurate tractable model of the frequency of events in a relational structure.

A direction for future work is to adapt more techniques from i.i.d. Bayes net parameter learning, such as smoothing frequencies and incorporating uncertainty in parameter estimates [**?**]. A theoretical understanding of estimator variance would be desirable: we may adapt the asymptotic approximations of [**?**], or apply graph estimator theory [**?**]. A plausible hypothesis is that recursive dependencies of an attribute on itself lower the effective sample size and hence increase the parameter variance.