Relational Random Regression for Bayes Nets (Poster Presentation at UAI-StarAI Workshop 2012)

Oliver Schulte and **Hassan Khosravi** and **Tianxiang Gao** and **Yuke Zhu**
School of Computing Science
Simon Fraser University
Vancouver-Burnaby, Canada

Abstract

Bayes nets (BNs) for relational databases are a major research topic in machine learning and artificial intelligence. When the database exhibits cyclic probabilistic dependencies, the usual Bayes net product formula does not define valid inferences. In this paper we describe and evaluate a new approach to defining Bayes net relational inference in the presence of cyclic dependencies. The key idea is to define the random regression log-probability of a target node value (unnormalized) as the expected log-probability (unnormalized) associated with a *random* instantiation of the node's Markov blanket. We provide a tractable closed form for random regression, which is equivalent to a log-linear model, but with the predictors scaled to be instance frequencies of relational patterns (features), rather than instance counts. Instance counts are used in previous inference models based on Markov networks. We carried out an empirical comparison on five benchmark databases with (i) weights as log-conditional probabilities using maximum likelihood estimates vs. (ii) general weights learned with Markov net methods. Maximum likelihood estimates took seconds to compute in comparison to hours for Markov net learning. With the frequency scaling, predictive accuracy for the conditional probability weights was competitive with the general weights.

1 Introduction

An important machine learning task is to use data to build a *generative statistical model* that represents the joint distribution of the random variables that describe the application domain [1]. One of the most widely used generative model classes are Bayes nets (BNs) [2]. The most common approach to relational inference with graphical models is knowledge-based model construction (KBMC) [3, 4]: A first-order or class-level model serves as a template, that is instantiated or ground with the complete information in the database. A major difficulty with KBMC for directed models is that the instantiated model may contain cycles even if the class-level model does not [5, 6, 7]. In the presence of cycles, the usual Bayes net product formula does not define valid probabilistic inferences. In this paper we propose a new relational log-linear inference model for Bayes nets that does not assume that the ground model is acyclic. The main idea is to consider a *random instantiation* of the class-level model, rather than a complete instantiation as in KBMC.

Approach. The first step in our inference model is to define the *Markov blanket probabilities*, which specify the conditional probability of a ground target node given an assignment of values to all other ground nodes, or equivalently, given an assignment of values to the Markov blanket of the target node. The second step extends the Markov blanket probabilities to a joint distribution for general inferences. It is well-known from the theory of dependency networks that Gibbs sampling can be used to define the extension [8, 9, 10], so this paper focuses on defining the Markov blanket distributions.

The key new idea in defining Markov blanket probabilities is to consider a *random instantiation* of the target's node Markov blanket. A single grounding of a class-level Bayes net contains no cycles, so the conditional probability of a target node value, given a single random instantiation of its Markov blanket, can be defined using the standard Bayes net formula. The unnormalized Markov blanket log-probability of a target node, given an assignment of values to all other ground nodes, is the expected value of the unnormalized log-conditional probability associated with a random instantiation of the Markov blanket. We refer to this as the *random regression* probability of a target node value. The random instantiation idea can be viewed as an application of Halpern's random selection probablistic semantics for first-order logic [11, 12].

Theoretical Analysis: Closed Form and Log-linearity. We establish a closed form for random regression that avoids constructing a ground network. This result also provides a simple comparison with standard Markov field log-linear models for relational data, such as relational Markov networks [6] and Markov Logic Networks [13]. Markov field models define a log-linear regression equation, where the features or predictors are derived from the Markov blanket of a target node. For instance, to predict the intelligence y of a student, the model may use as predictive features how many A grades she has received, how many B grades, etc. The closed-form equation for random regression is a log-linear equation, just as with Markov field models, but with feature *frequencies* replacing feature counts. Based on this result, we show how different relational graphical models, such as Bayesian, Markov, and dependency networks, can be compared in terms of their corresponding regression equations.

Parameter Learning. A straightforward approach to parameter learning for a log-linear model is to estimate the Bayes net conditional probability parameters and use the log-conditional probabilities as weights [5]. This approach is attractive because conditional probabilities have an intuitive interpretation, and because its computational cost scales well with both the size of the dataset and the number of model parameters. However, using the Bayes net parameters with the count model faces the *balancing problem*: Features with more instances have exponentially more influence. Since the Bayes net parameters are all on the same scale (log-probability), smaller weights cannot sufficiently scale down the impact of predictors with larger domains. In contrast, random regression uses feature frequencies, so the predictors are scaled to the common range [0,1].

Empirical Evaluation. The structure of a model is determined by a fixed Bayes net for a given database. We learn the Bayes net by applying the learn-and-join algorithm to five benchmark databases [14]. For each Bayes net structure, we compare two types of log-linear relational models: (1) Using counts as predictors vs. (2) frequencies as predictors. We compare two different methods for parameter learning: (1) Using the maximum likelihood estimates for the Bayes nets (empirical conditional frequencies in the data), with their logs as weights in the log-linear regression model, vs. (2) weights optimized using Markov Logic Network methods. Using log-conditional probabilities as parameters makes for much faster weight learning (seconds vs. hours in our experiments). With log-conditional probabilitiies, the frequency model outperforms the count model on all databases (on the log-likelihood metric). The predictive performance of log-conditional probabilities is competitive with optimized weights; the combination of log-conditional probabilities + frequency predictors outperforms optimized weights on all but one dataset.

While this paper focuses on Bayes net models, the distinction between counts vs. frequencies as predictors can be explored in other log-linear relational models, for example as a different form of potential function for the recent functional gradient boosting approach [10, 15].

Paper Organization. We describe further related work. Then we present background: basic relational graphical models and connections between them. The next section defines the random regression, and relates it to the frequency and count log-linear regression models. We discuss parameter estimation with observed conditional probabilities. Empirical evaluation compares the frequency and count models on five benchmark databases.

Contributions. The main contributions of this paper to relational learning may be summarized as follows.

1. A new log-linear regression model for Bayes nets defined in terms of random instantiations of the Markov blanket of the target node. This model is well-defined even in the presence of cyclic dependencies.
2. A closed form for the random regression model. This allows fast computation of the model's predictions, and provides a straightforward comparison with state-of-the-art log-linear relational models: Whereas previous log-linear models use feature counts as predictors, the random regression model scales counts to be frequencies.
3. Experiments indicate that the random regression model, with quickly computed maximum likelihood estimates, offers predictive performance that is competitive with a general log-linear model using optimized weights.

2 Related Work

Moralization Methods. Richardson and Domingos propose converting a Bayes net to a Markov Logic network using moralization, with log-conditional probabilities as weights [5]. This is also the standard Bayes net conversion recommended by the Alchemy system [16]. The moralization method is equivalent to our log-linear model with counts. Khosravi et al. [14] follow the moralization approach for the model structure, but do not use log-probabilities as parameters for inference. To our knowledge, our experiments are the first that evaluate the moralized Bayes net structure with log-probability weights.

Natarajan et al. [17] consider moralization with Bayes nets that have been augmented with combining rules for mapping probabilities obtained from multiple parent instances to a single one. In contrast, we consider tabular Bayes nets whose parameters are CP-table entries only. Combining rules do not generally lead to log-linear models.

Scaling Predictors. Scaling predictors to the [0,1] range has been previously applied in a log-linear classification model [18]. To our knowledge, scaling has not been applied for inference in the generative context. Variants of the Markov net pseudo-likelihood have been proposed that include scaling factors, such as the Weighted Pseudo Log-Likelihood [13] and the random selection pseudo-likelihood [11]. The key difference is that these scaling factors are used only during *learning* to ensure that the learning algorithm optimizes parameters sufficiently for features with low counts. In contrast, we use the scaling parameters during inference.

The frequency model uses both global shared parameters (conditional probabilities) and local scaling factors that depend on the individual target node. Combining rules like the arithmetic mean [17] similarly combine global parameters with a local scaling factor. Our frequency model uses the *geometric mean* rather than the arithmetic mean. To our knowledge, the geometric mean has not been used in Bayes net models with relational data. Another difference with combining rules is that we apply scaling to the entire Markov blanket of the target node, whereas a Bayes net combining rule applies only to the parents of the target node.

Random Selection Pseudo-Likelihood. Schulte uses the expected log-likelihood associated with a random grounding to define a *generative* pseudo-likelihood measure for a first-order Bayes net and given input database [11]. In this work we use the random grounding idea *discriminatively* to define a regression equation for Markov blanket probabilities.

3 Background: Relational Graphical Models

With respect to a graphical model, we interchangeably refer to its nodes and its variables. We use vector notation for lists of variables/nodes and for lists of values assigned to them, e.g., $P(X_1 = x_1, \ldots, X_n = x_n) \equiv P(\mathbf{X} = \mathbf{x})$.

3.1 Graphical Models

We consider graphical models with discrete random variables only. A Bayes net (BN) is a pair $\langle G, \boldsymbol{\theta}_G \rangle$ where $\boldsymbol{\theta}_G$ is a set of parameter values that specify the probability distributions of children conditional on instantiations of their parents, i.e. all conditional probabilities of the form

$$\theta_{ijk} \equiv P(v_i = a_{ik} | \mathbf{PA}_i = \mathbf{pa}_{ij}),$$

where a_{ik} is the k-th possible value of node i and $\mathbf{pa}_{ij}$ is the j-th possible configuration of the parents of v_i. The conditional probabilities are specified in a **conditional probability table** for variable v_i or CP-table. The Markov blanket of a BN node Y_i comprises the set of children$_i$, parents$_i$ and co-parents$_i$ that share a child with node Y_i. The unnormalized **Markov blanket classification equation** [19, Ch.14.5.2] is given by

$$\tilde{P}(Y_i = y | \mathbf{X} = \mathbf{x}) = P(Y_i = y | \mathbf{pa}_i) \cdot \prod_{X_j \in \text{children}_i} P(X_j = y | \mathbf{pa}_j) \quad (1)$$

where $\mathbf{X}$ is the set of all nodes other than Y_i.

A **Markov network** structure is an undirected graph. For each clique C in the graph, a **clique potential function** Ψ_C specifies a nonnegative real number for each possible assignment of values to the clique.

A **dependency network** structure is a directed graph; cycles are allowed [8, 9, 10]. The parameters are conditional probabilities of each node, given its *Markov blanket*. Dependency networks are like Markov networks in that conditional probabilistic independence corresponds to graph separation. They are like Bayes nets in that the parameters are conditional probabilities.

3.2 Graphical Models for Relational Data

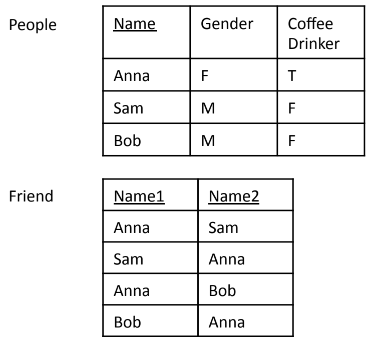
We follow the original presentation of Parametrized Bayes Nets (PBNs) due to Poole [20]. A **functor** is a function symbol or a predicate symbol. In this paper we discuss only functors with a finite range of possible values. A **parametrized random variable** or **functor node** is of the form $f(\tau_1, \ldots, \tau_k) = f(\mathbf{A})$ where f is a functor and each τ_i is a first-order variable A_i or a constant a_i of the appropriate type for the functor.[1] If a functor node $f(\boldsymbol{\tau})$ contains no variable, it is a **ground node**. An assignment to a ground node of the form $f(\boldsymbol{\tau}) = a$, where a is a constant in the range of f, is a **ground literal** [21]. A **population** is a set of individuals, corresponding to a domain or type in logic. Each first-order variable A is associated with a population. An **instantiation** or **grounding** for a set of variables $A_1, \ldots, A_k$ assigns to each variable A_i a constant from the population of A_i.

A **Parametrized** (Bayes, Markov, Dependency) Network is a (Bayes, Markov, Dependency) Network whose nodes are functor nodes. We usually omit the prefix "Parametrized". Figure 1 shows a simple relational database and Figure 2 shows a Parametrized Bayes net for this relational schema. A database instance specifies a unique value for each ground node; we denote such a joint assignment by $\mathbf{V} = \mathbf{v}$. For instance, the database in Figure 1 specifies the value M for the ground node $gender(sam)$, and the value T for the ground node $Friend(anna, sam)$. We use the following notation.

- F_{ijk} is the **family state** that expresses that functor node f_i is assigned its k-th value, and the state of its parents is assigned its j-th value.
- $n_{ijk}(\mathbf{V} = \mathbf{v})$ is the number of groundings of F_{ijk} that evaluate as true for a given complete assignment of values (= database instance).
- $p_{ijk}(\mathbf{V} = \mathbf{v})$ is the frequency of the family state in the database, that is, the number of groundings that evaluate as true, over the number of possible groundings.

To illustrate, let $\mathbf{V} = \mathbf{v}$ be the ground node assignment corresponding to the database instance in Figure 1. Also, choose F_{ijk} to be the assignment $gender(X) = M, gender(Y) = F, Friend(X, Y) = T$. Then $n_{ijk}(\mathbf{V} = \mathbf{v}) = 2$, and $p_{ijk}(\mathbf{V} = \mathbf{v}) = 2/9 = 1/3$.

Recursive dependencies (autocorrelations) are represented in a PBN by "copies" of the functors. Thus the structure $gender(X) \rightarrow gender(Y) \leftarrow Friend(X, Y)$ in Figure 2 represents an association between the gender of a user and that of his/her friends. We assume that the Bayes net is in main functor format [22]: for each functor f, there is a main functor node that is the only f-node with parents. In the example, $gender(X)$ is the main functor, and $gender(Y)$ is an auxilliary functor used only for representing the recursive dependency. While the existence of a main functor may seem like a strong assumption, Schulte *et al.* show that under a mild ordering condition on the BN structure, for every PBN B not in main functor format, there is an equivalent main functor Bayes net B' that has the same ground graph [22].



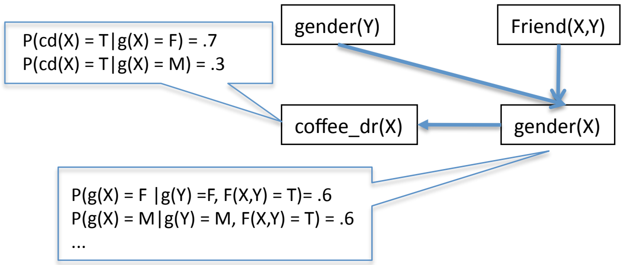
People

Name	Gender	Coffee Drinker
Anna	F	T
Sam	M	F
Bob	M	F

Friend

Name1	Name2
Anna	Sam
Sam	Anna
Anna	Bob
Bob	Anna

Fig. 1. A simple relational database instance.



P(cd(X) = T|g(X) = F) = .7
P(cd(X) = T|g(X) = M) = .3

P(g(X) = F |g(Y) =F, F(X,Y) = T)= .6
P(g(X) = M|g(Y) = M, F(X,Y) = T) = .6
...

Fig. 2. A Parametrized Bayes Net with some CP-table entries. CP-table entries are chosen for illustration and are not related to the data in Figure 1.

Model Conversions. Bayes nets can be converted to Markov nets through the standard **moralization** method: connect all spouses that share a common child, and make all edges in the resulting graph undirected. Thus each family in the Bayes net becomes a clique in the moralized structure. For each state of each family clique, we define the clique potential in the Markov net to be the conditional probability of the child given its parents.

Bayes nets can also be converted to dependency nets [8]. For each node X_i, and each node X_j in the Markov blanket of X_i, add a directed edge $X_j \rightarrow X_i$. The conditional probability parameters are given by the Markov blanket equation (1).

4 Random Regression

Let $Y = f(a_1, \ldots, a_k)$ be a target ground node instantiating functor node $f(A_1, \ldots, A_k)$. The **regression graph** for Y is the partially ground PBN B_Y that results by substituting a_i for A_i in functor node Y and in its Markov blanket. This is illustrated in Fig-

[1] We use the term "functor node", for brevity and to avoid confusion with the statistical sense of "parametrized", meaning that values have been assigned to parameters.

ure 3. If there is more than one functor node with f, we use the main functor node (Sec. 3.2). Given a target node value y and an assignment $\mathbf{X} = \mathbf{x}$ of values to all ground nodes other than Y, random regression is defined by the following steps.

1. Let $A_1, \ldots, A_k$ be a list of *all* first-order variables that occur in the Markov blanket of target node Y in the regression graph for Y.
2. Select an instance (constant) a_i from the population of A_i, for each $i = 1, \ldots, k$; the selections are random, independent, and uniform. Replace each node in the Markov blanket with the corresponding ground node.
3. Using the values assigned to the ground nodes in the database, apply the Bayes net Markov blanket equation (1) to compute the factor product for Y; this defines a log-sum for the random instantiation $\mathbf{a_i}$. The expected value of this log-sum is the **random regression** value $ln(\tilde{P}^r(Y = y|\mathbf{X} = \mathbf{x}))$.

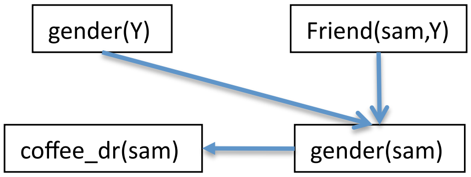


Fig. 3. The regression graph for the target node *gender*(*sam*) derived from the Bayes net of Figure 2 by substituting *sam* for X.

Including irrelevant predictors leads to bad predictions, so statistical-relational models restrict edges in the ground model to relevant predictors only [20]. In our examples and experiments below, we take the relevance conditions to be the existence of a link, *so we only consider instances of the Markov blanket that are related to the target node.* Table 1 illustrates a sample computation of a random regression for predicting the gender of Sam given the database instance of Figure 1.

Table 1. Computing the random regression for target node node value *gender*(*sam*) = F. We use obvious abbreviations for functors. Each friend selection defines an instantiation of the Markov blanket of the target node with two associated factors.

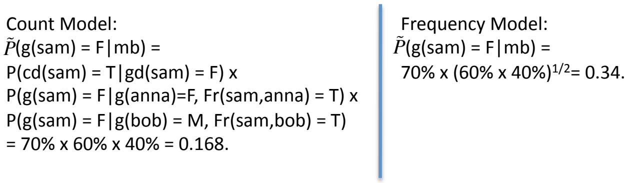
Grounding	Factor 1	Factor 2	Log-Product
$Y = anna$	$P(cd(sam) = T\|g(sam) = F)$ $= .7$	$P(g(sam) = F\|g(anna) = F,$ $Fr(sam, anna) = T) = .6$	$ln(.7 \times .6)$ = -0.87
Y=bob	$P(cd(sam) = T\|g(sam) = F)$ $= .7$	$P(g(sam) = F\|g(bob) = M,$ $Fr(sam, bob) = T) = .4$	$ln(.7 \times .4)$ =-1.27
		Average	-1.07

5 Random regression and Log-linear Regression: The Frequency Model

Random regression sums over the set of all ground functor nodes in the Markov blanket, which increases exponentially with the number of first-order variables in the Markov blanket. Frequency regression provides an equivalent formula that sums over all *non-ground* functor nodes in the Markov blanket. The **frequency regression equation** is given by

$$ln(\tilde{P}(Y = y|\mathbf{X} = \mathbf{x})) = \sum_{ijk} p^Y_{ijk}(\mathbf{X} = \mathbf{x}, Y = y)\ ln(\theta_{ijk}). \tag{2}$$

Here and elsewhere the superscript Y indicates that the notation is used with reference to the regression graph for target node Y. The summation is over Y's Markov blanket in the regression graph, so the index i ranges over the target node and its children. Figure 4 provides an example computation of frequency regression. Random regression (Table 1) gives the same value $-1.07 = ln(0.34)$ as frequency regression for the unnormalized conditional log-likelihood of *gender*(*sam*) = F. The next proposition shows that this equivalence holds in all cases. We omit the proof due to space constraints.



Count Model:
$\tilde{P}$(g(sam) = F|mb) =
P(cd(sam) = T|gd(sam) = F) x
P(g(sam) = F|g(anna)=F, Fr(sam,anna) = T) x
P(g(sam) = F|g(bob) = M, Fr(sam,bob) = T)
= 70% x 60% x 40% = 0.168.

Frequency Model:
$\tilde{P}$(g(sam) = F|mb) =
70% x (60% x 40%)$^{1/2}$= 0.34.

Fig. 4. The computation of the unnormalized Markov blanket probability for the gender of Sam, for the count model (left) and the frequency model (right). The frequency model assigns higher weight to the factor that represents the coffee drinking of Sam. Notice that $ln(0.34) = -1.07$, so the log-probability defined by the frequency models agrees with the log-probability defined by random regression.

Proposition 1. *The frequency regression value for a target node (Equation (2)) equals the random regression value.*

We remark that this result applies to random regression with any graphical model based on a first-order template, not only Bayes nets. The random regression inference model can be interpreted in terms of a ground dependency network, whose graph structure is given by converting the Bayes net to a dependency network (Sec. 3.2), and whose conditional probability parameters are given by random regression.

6 Log-linear Regression: The Count Model

Frequency regression has a simple relationship to the standard log-linear regression equation associated with Markov random fields. This regression equation is given by

$$ln(\tilde{P}(Y = y|\mathbf{X} = \mathbf{x})) = \sum_{ijk} n^{Y}_{ijk}(\mathbf{X} = \mathbf{x}, Y = y)\; ln(\theta_{ijk}). \tag{3}$$

Since Equation 3 uses the family counts n_{ijk} rather than the family frequencies p_{ijk}, we refer to it as the **count equation**. To see the relationship with Markov random fields, consider the Parametrized Markov net M obtained by moralizing the Parametrized Bayes net (Sec. 3.2). The count equation is obtained by applying the standard Markov field regression equation to the grounding of M [13]. In graphical terms, the equation is the product of all clique potentials in which the target node participates; see Figure 4. Count regression is a natural comparison point to random regression due to their similarity.

In the count regression equation (3), Markov blanket components with many groundings have exponentially more influence. The frequency model balances the scales of the predictor variables, whose common range is [0,1]. In terms of the factor products defined by exponentiating the log-linear equations, the count equation multiplies together all ground Markov blanket factors, whereas the frequency equation first computes the *geometric mean* of the ground factors associated with each functor node in the Markov blanket, then multiplies these geometric means. Figure 5 summarizes the theoretical connections between graphical models and regression equations.

Inference. Assuming complete data, the regression equations can be evaluated in closed-form for conditional inference. We outline how the regression models can be extended to general joint inferences. For the count model, Markov logic network inference algorithms can be used after moralization, as in [23, 14]. Heckerman et al. [8] show that applying Gibbs sampling to a dependency network defines a stationary joint distribution, hence can be used to answer general queries based on random/frequency regression. Their ordered pseudo-Gibbs sampler has been lifted to the relational setting [24]. Since the form of the frequency equation is very similar to that of the count model, an alternative is to adapt MLN inference methods developed for the count model.

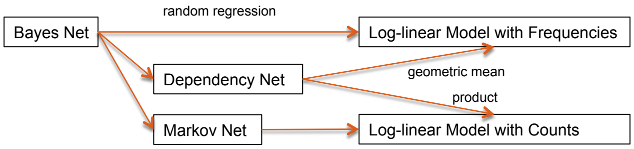


Fig. 5. Connections between graphical models and log-linear regression equations. Applying random regression directly to a Parametrized Bayes net leads to the frequency model Proposition 1. Converting the Bayes net to a Markov net leads to the count model. Converting the Bayes net to a dependency network leads to the frequency model if the geometric mean is used to combine Markov blanket conditional probabilities (Eq. 1). Using the product of conditional probabilities instead leads to the count model.

7 Parameter Learning With Bayes net Maximum Likelihood Values

Our experiments evaluate using the empirical conditional frequencies as parameters in the Bayes net model:

$$\widehat{\theta}_{ijk} = \frac{n_{ijk}(\mathbf{V} = \mathbf{v})}{\sum_k n_{ijk}(\mathbf{V} = \mathbf{v})}.$$

We believe that these estimates are well motivated by the following theoretical and practical considerations.

Maximum Likelihood Solution. The random selection pseudo-likelihood for a Bayes net is the natural generative counterpart of random regression [11]. This measure is the expected log-likelihood of a random instantiation of all first-order variables in the Bayes net. The pseudo-likelihood is maximized by the conditional frequencies $\widehat{\theta}_{ijk}$ in the database [11, Prop.3.1]. This result is a counterpart to the standard maximum likelihood solution for i.i.d. propositional data.

Interpretability. The weight/clique potential parameters of undirected models are often difficult to interpret for users [2]. This is especially the case when weights are learned from data, which can reflect complex interactions between weights assigned to different local cliques. In contrast, a Bayes net parameter can be interpreted as a conditional probability, and reflects local statistics restricted to a parent-child constellation.

Scalability. Using frequency estimates can be viewed as a type of *lifted learning*, by which we mean using only the sufficient statistics in a relational database rather than an iteration over ground facts. The computational cost scales well in both the size of the data and the number of parameters in the model.

8 Evaluation

We first discuss the datasets used, then the systems compared, finally the comparison metrics. We used 5 benchmark real-world databases. For more details please see the references in [14] and on-line sources such as [25].

MovieLens Database. This is a standard dataset from the UC Irvine machine learning repository.

Mutagenesis Database. This dataset is widely used in ILP research. It contains information on Atoms, Molecules, and Bonds between them. We use the discretization of [14].

Hepatitis Database. This data is a modified version of the PKDD02 Discovery Challenge database. The database contains information on the laboratory examinations of hepatitis B and C infected patients.

Mondial Database. This dataset contains data from multiple geographical web data sources. We followed the modification of [26], and used a subset of the tables and features for fast inference.

UW-CSE database. This dataset lists facts about the Department of Computer Science and Engineering at the University of Washington (UW-CSE), such as entities (e.g., Student, Professor) and their relationships (i.e. AdvisedBy, Publication). The dataset was obtained by crawling pages in the department's Web site (www.cs.washington.edu).

8.1 Performance Metrics.

We use 3 performance metrics: Learning Time, Accuracy (ACC), and Conditional log likelihood (CLL). ACC and CLL have been used in previous studies of MLN learning [27, 14]. The CLL of a ground atom in a database is given by the log of the regression equation; for a database we report the average CLL over all atoms in the test set. To define accuracy, we apply inference to predict the probability of an attribute value, and score the prediction as correct if the most probable value is the true one. For ACC and CLL the values we report are averages over all predicates that represent descriptive attributes. We do not use Area under Curve, as it mainly applies to binary values, and most of the attributes in our dataset are nonbinary. We evaluate the learning methods using 5-fold cross-validation as follows. We formed 5 subdatabases for each by randomly selecting entities from each entity table and restricting the relationship tuples in each subdatabase to those that involve only the selected entities (subgraph sampling [28, 14]). The models were trained on 4 of the 5 subdatabases, then tested on the remaining fold. We report the average score over the 5 runs, one for each fold.

8.2 Comparison Systems.

All experiments were done on a QUAD CPU Q6700 with a 2.66GHz CPU and 8GB of RAM. Our code and datasets are available on the world-wide web [25]. We applied the learn-and-join algorithm to learn a Bayes net structure for each database [14]. A limitation of the current learn-and-join algorithm is that it learns a generative model over attributes given link structure, so our evaluation considers only queries whose target are attributes, not links [29, 14].

Parameter learning for general weights proceeds in two steps as in [14]: (1) Convert the Parametrized Bayes nets to Markov Logic Networks, using moralization, which adds a conjunctive clause for each family state F_{ijk} [5]. We declared attribute predicates as functional as recommended by the Alchemy Group [16]. (2) A Markov net model uses a general weight w_{ijk} in place of $ln(\theta_{ijk})$ derived from a conditional probability. To learn the w_{ijk} weights, we applied the default weight training procedure [30] of the Alchemy package [31].

Inference is performed by evaluating the count resp. frequency regression equation. We employ exact inference rather than approximate inference (e.g., MC-SAT) to avoid conflating the impact of the inference model with the impact of the inference implementation. We conducted experiments with MC-SAT and the results were similar. We compared the following approaches.

MBN As described above, the Bayes net structure is converted to an MLN using moralization, weights learned using Alchemy [14]. Inference uses count regression. This is the state-of-the-art method for log-linear prediction with Bayes nets, and therefore our baseline comparison.

CP+Count Parametrizes the Bayes net with the empirical conditional probabilities and uses count regression.

CP+Frequency Parametrizes the Bayes net with the empirical conditional probabilities and uses frequency regression.

8.3 Results.

All results are averages from 5-fold cross validation, over all attributes in the database.

Learning Times. Table 2 shows runtime results for parameter learning. We see *clear scalability advantages* *for the maximum likelihood conditional probability estimates*: they take seconds to compute, whereas the local search method requires as much as 10 hours in the worst case (Hepatitis).

Table 2. A comparison of runtime (seconds) required for parameter learning with a fixed Bayes net structure. The Bayes net methods use the observed conditional frequencies. The Markov net methods use Alchemy's default weight learning. Database sizes are specified by the number of tuples and the number of ground atoms.

Dataset	Bayes Net (s)	Markov Net (s)	#tuples	#Ground atoms	#Parameters
UW	**2**	5	2099	3380	125
Mondial	**3**	90	814	3366	575
MovieLens	**8**	10800	82623	170143	327
Mutagenesis	**3**	14400	15218	35973	880
Hepatitis	**3**	36000	12447	71597	793

Predictive Accuracy. Table 3 compares the log-likelihood score of the methods, and Table 4 their accuracy score. Figure 6 averages performance over all five databases to provide a simple visual summary of our findings. We first discuss the frequency vs. count models, and then compare CP weights with the Markov net weights.

Table 3. Conditional log-likelihood comparison of the Bayes net parameters (cp+) with the Markov net parameters (mbn), which are general weights. MBN is the previous state-of-the-art baseline method.

Method	UW	Mondial	MovieLens	Mutagenesis	Hepatitis
mbn	-0.44 ± 0.07	**-1.28** ± 0.07	-0.79 ± 0.03	-0.91 ± 0.09	-1.18 ± 0.26
cp+count	-0.42 ± 0.05	-1.36 ± 0.11	-1.10 ± 0.16	-0.77 ± 0.03	-1.20 ± 0.07
cp+freq	**-0.41** ± 0.04	-1.34 ± 0.09	**-0.71** ± 0.01	**-0.73** ± 0.04	**-1.07** ± 0.10

Table 4. Accuracy score of the Bayes net parameters (cp+), which are conditional probabilities, with the Markov net parameters (mbn). Accuracy is the percentage of correctly predicted values in the test data.

Method	UW	Mondial	MovieLens	Mutagenesis	Hepatitis
mbn	80.25% ± 0.05	43.81% ± 0.04	59.71% ± 0.02	61.49% ± 0.02	51.01% ± 0.02
cp+count	80.89% ± 0.06	**44.70%** ± 0.04	61.93% ± 0.02	66.95% ± 0.03	**55.12%** ± 0.02
cp+freq	**81.01%** ± 0.06	44.59% ± 0.04	**65.14%** ± 0.01	**66.96%** ± 0.03	54.79% ± 0.02

Frequency vs. Count Model. *CLL. Using frequencies rather than counts improves the conditional log-likelihood score for the CP model*, substantially on MovieLens and Hepatitis (by 0.4 resp. 0.13 log-likelihood units). Whereas accuracy is a 0-1 loss function, CLL is continuous, so we expect the balancing of factors to have more impact.

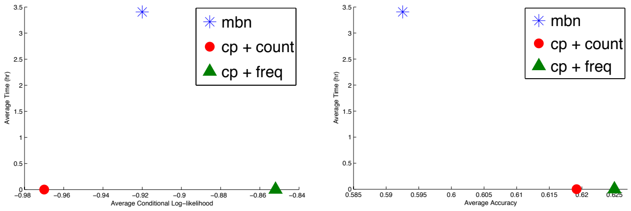


Fig. 6. Performance on conditional log-likelihood against time, averaged over all five benchmark databases. Compared to the Markov Logic methods, Bayes net parameter learning takes essentially no time.

Accuracy. The count and frequency models are close, except for MovieLens, where the frequency method has a 3% advantage. MovieLens is an especially unbalanced set because the number of ratings varies from movie to movie and user to user. Also, there are generally many more users rating a given movie than movies rated by a given user.

Bayes net vs. Markov net parameters. Bayes net weights are competitive with the optimized weights with both regression equations.

CLL. The CP+frequency model scores substantially better than the Markov net weights on Mutagenesis, Hepatitis and MovieLens (by 0.18, 0.11, 0.08 log-likelihood units) but worse on Mondial (0.06 difference).

Accuracy. The CP+frequency models have a slightly higher score than the Markov net weights, with the biggest differences on MovieLens (5%) and Hepatitis (4%).

We also performed experiments using the Markov net weights together with the frequency model. There is little difference between the Markov model with counts and frequencies. We hypothesize that this is because the optimized Markov model weights include a scaling component. This hypothesis is confirmed by the scaling components of the weights directly; we omit the details due to space constraints.

Experimental Conclusions. Compared to Markov Logic parameter learning, Bayes net parameter learning is very fast. The Bayes net parameters were competive with the Markov Logic parameters in terms of predictive performance. Using Bayes net parameters with random/frequency regression outperformed the Markov parameters on all but one dataset on our main metric (CLL). Comparing Bayes net parameters using the frequency vs. count regression, the frequency model has better performance on all datasets on CLL. Together with our analysis of the balancing problem, the empirical findings make a good case for recommending the frequency model over the count model when the CP parameters are used.

9 Conclusion and Future Work

This paper considered an inference model for Bayes nets applied to linear data, that is well defined in the presence of cyclic dependencies. The key idea is to consider the expected log-linear regression value from a *random* instantiation of a node's Markov blanket. We provided an equivalent closed form definition that shows that random regression is equivalent to a log-linear model, whose predictors are scaled to be frequencies in the range [0,1]. We compared random regression with standard log-linear models, using both the empirical conditional frequencies and weights learned by local optimization. The log-conditional probabilities are much faster to compute, typically seconds vs. hours. The predictive performance of log-conditional probability weights was competitive with optimized regression weights, in fact superior on all but one dataset.

References

[1] Getoor, L., Taskar, B.: Introduction. [32] 1–8

[2] Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann (1988)

[3] Ngo, L., Haddawy, P.: Answering queries from context-sensitive probabilistic knowledge bases. Theor. Comput. Sci. **171**(1-2) (1997) 147–177

[4] Wellman, M., Breese, J., Goldman, R.: From knowledge bases to decision models. Knowledge Engineering Review **7** (1992) 35–53

[5] Domingos, P., Richardson, M.: Markov logic: A unifying framework for statistical relational learning. [32]

[6] Taskar, B., Abbeel, P., Koller, D.: Discriminative probabilistic models for relational data. In: UAI. (2002) 485–492

[7] Getoor, L., Friedman, N., Koller, D., Pfeffer, A., Taskar, B.: Probabilistic relational models. [32] chapter 5 129–173

[8] Heckerman, D., Chickering, D.M., Meek, C., Rounthwaite, R., Kadie, C., Kaelbling, P.: Dependency networks for inference, collaborative filtering, and data visualization. Journal of Machine Learning Research **1** (2000) 49–75

[9] Neville, J., Jensen, D.: Relational dependency networks. [32] chapter 8

[10] Natarajan, S., Khot, T., Kersting, K., Gutmann, B., Shavlik, J.W.: Gradient-based boosting for statistical relational learning: The relational dependency network case. Machine Learning **86**(1) (2012) 25–56

[11] Schulte, O.: A tractable pseudo-likelihood function for Bayes nets applied to relational data. In: SIAM SDM. (2011) 462–473

[12] Halpern, J.Y.: An analysis of first-order logics of probability. Artificial Intelligence **46**(3) (1990) 311–350

[13] Domingos, P., Lowd, D.: Markov Logic: An Interface Layer for Artificial Intelligence. Morgan and Claypool Publishers (2009)

[14] Schulte, O., Khosravi, H.: Learning graphical models for relational data via lattice search. Machine Learning **88:3** (2012) 331–368

[15] Khot, T., Natarajan, S., Kersting, K., Shavlik, J.W.: Learning markov logic networks via functional gradient boosting. In: ICDM. (2011) 320–329

[16] Alchemy Group: Frequently asked questions URL = `http://alchemy.cs.washington.edu/`.

[17] Natarajan, S., Khot, T., Lowd, D., Tadepalli, P., Kersting, K., Shavlik, J.W.: Exploiting causal independence in markov logic networks: Combining undirected and directed models. In: ECML/PKDD (2). (2010) 434–450

[18] Raina, R., Shen, Y., Ng, A.Y., McCallum, A.: Classification with hybrid generative/discriminative models. In: NIPS. (2003)

[19] Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall (2010)

[20] Poole, D.: First-order probabilistic inference. In: IJCAI. (2003) 985–991

[21] Chiang, M., Poole, D.: Reference classes and relational learning. Int. J. Approx. Reasoning **53**(3) (2012) 326–346

[22] Schulte, O., Khosravi, H., Man, T.: Learning directed relational models with recursive dependencies. Machine Learning (2012) Forthcoming.

[23] Khosravi, H., Schulte, O., Hu, J., Gao, T.: Learning compact markov logic networks with decision trees. Machine Learning (2012) Forthcoming.

[24] Neville, J., Jensen, D.: Relational dependency networks. Journal of Machine Learning Research **8** (2007) 653–692

[25] Khosravi, H., Man, T., Hu, J., Gao, E., Schulte, O.: Learn and join algorithm code. URL = `http://www.cs.sfu.ca/~oschulte/jbn/`.

[26] She, R., Wang, K., Xu, Y.: Pushing feature selection ahead of join. In: SIAM SDM. (2005)

[27] Kok, S., Domingos, P.: Learning markov logic network structure via hypergraph lifting. In: ICML. (2009) 64–71

[28] Frank, O.: Estimation of graph totals. Scandinavian Journal of Statistics **4:2** (1977) 81–89

[29] Khosravi, H., Schulte, O., Man, T., Xu, X., Bina, B.: Structure learning for Markov logic networks with many descriptive attributes. In: AAAI. (2010) 487–493

[30] Lowd, D., Domingos, P.: Efficient weight learning for Markov logic networks. In: PKDD. (2007) 200–211

[31] Kok, S., Summer, M., Richardson, M., Singla, P., Poon, H., Lowd, D., Wang, J., Domingos, P.: The Alchemy system for statistical relational AI. Technical report, University of Washington. (2009) Version 30.

[32] Getoor, L., Tasker, B.: Introduction to statistical relational learning. MIT Press (2007)