Relational Random Regression for Bayes Nets

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

Abstract

Bayes nets 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. We describe 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 as the expected log-probability for a *random* instantiation of the node's Markov blanket. A provably equivalent closed form for random regression is a log-linear model, whose predictors are scaled to be instance *frequencies* of relational features, rather than instance *counts*. Instance counts were used in previous inference models based on Markov networks. We carried out an empirical comparison on five benchmark databases with (i) log-linear weights computed from Bayes net conditional probability parameters vs. (ii) general weights learned with Markov net methods. Bayes net maximum likelihood estimates took seconds to compute in comparison to hours for Markov net learning. With the frequency scaling, predictive accuracy for the Bayes net weights was competitive with the general weights.

1 Introduction

The most common approach to relational inference with graphical models is knowledge-based model construction (KBMC) [17]: A first-order or class-level model serves as a template, that is instantiated, or ground, with the individuals listed in the database. A major difficulty with KBMC for Bayes nets [18] is that the instantiated model may contain cycles, even if the class-level model does not [2, 24, 4]. 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.

Approach. Our inference model defines 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. This definition allows relational Bayes nets to be used discriminatively, for predicting the value of a target variable. For joint query prediction, Gibbs sampling can be used to the Markov blanket probabilities to a joint distribution. This extension is well studied in the theory of dependency networks [6, 16, 14, 12].

The key new idea in defining Markov blanket probabilities is to consider a *random instantiation* of the targets node's Markov blanket. 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 given a single random instantiation of the Markov blanket. We refer to this as the *random regression* probability of a target node value.

Theoretical Analysis: Closed Form and Log-linearity. We establish an efficient closed form for random regression that avoids constructing a ground network. This result also provides a comparison with standard Markov field log-linear models for relational data [24], [2]. Markov field models define a log-linear regression equation, where the predictors are feature counts 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 with feature *frequencies* instead of feature counts. This result shows how different relational graphical models, such as Bayesian, Markov, and dependency networks, can be compared in terms of their corresponding regression equations.

Parameter Learning. The parameters in a log-linear model are weights assigned to each predictor. An important observation for this paper is that the choice of parameter learning method interacts with the choice of features vs. counts as predictors. The scales of feature count predictors diverge, and features with more in-

stances have exponentially more influence. In contrast, feature frequency predictors are scaled to the common range [0,1]. We provide empirical evidence that, the scaling problem impairs predictive performance when the log-linear count model is combined with weights derived from Bayes net parameters.

Empirical Evaluation. For a given Bayes net structure, we compare two types of log-linear relational models: (1) Using counts as predictors vs. (2) frequencies as predictors (random regression). We compare two methods for parameter learning that employ weights derived from Bayes net conditional probabilities, and one that optimizes weights using Markov Logic Network learning. Deriving weights from Bayes net conditional probabilities makes for much faster weight learning (seconds vs. hours in our experiments). Using Bayes net parameters, the frequency/random regression model outperforms the count-based model. Our code and datasets are available on the world-wide web [8].

While this paper focuses on Bayes net models, the distinction between counts vs. frequencies as predictors applies also in other log-linear relational models [14, 10].

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

Contributions. The main contributions are 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.
3. Two different methods to compute log-linear weights from Bayes net conditional probability parameters.

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 [2]. This is also the standard Bayes net conversion recommended by the Alchemy system [1]. The moralization method is equivalent to our log-linear model with counts. Khosravi et al. [22] apply moralization approach to 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. [15] consider moralization with Bayes nets that have been augmented with combining rules for mapping probabilities obtained from multiple parent instances to a single one. We consider tabular Bayes nets whose parameters are CP-table entries only.

Scaling Predictors. Scaling predictors to the [0,1] range has been previously applied for learning with a log-linear classification model [20], and for learning with a generative model (e.g., the Weighted Pseudo Log-Likelihood [2] and the random selection pseudo-likelihood [21]). The key difference is that scaling is used only during *learning*, to ensure that the learning algorithm optimizes parameters sufficiently for features with low counts. In contrast, we use scaling during *inference*.

Combining Rules. 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 [15] 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. 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 of all nodes to define a *generative* pseudo-likelihood measure for first-order Bayes nets [21]. 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 θ_{ijk} values are specified in a **conditional probability table** or CP-table. The Markov blanket of a 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 conditional probability** is given by

$$\tilde{P}(Y_i = y|\mathbf{X} = \mathbf{x}) = P(Y_i = y|\mathbf{pa}_i) \cdot \prod_{X_j : Y_i \to X_j} P(X_j = y|\mathbf{pa}_j) \tag{3.1}$$

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 [6, 16, 14]. The parameters are conditional probabilities of each node, given its *Markov blanket*.

3.2 Graphical Models for Relational Data We follow the original presentation of Parametrized Bayes Nets (PBNs) due to Poole [19]. 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. 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**. 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 $A_1, \ldots, A_\ell$ 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 PBN for this database schema. The structure $gender(X) \rightarrow gender(Y) \leftarrow Friend(X, Y)$ in Figure 2 represents an association (autocorrelation) between the gender of a user and that of his/her friends [23].

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 assigns the k-th possible value to functor node f_i, and the j-th possible state to the parents.
- $n_{ijk}(\mathbf{V} = \mathbf{v})$ 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.

It is useful to reserve the parent index value $j = 0$ for the unconditional probability that node i takes on value k, that is, with no parent state specified. So F_{i0k} is equivalent to the assignment $f_i = k$. Similary we write $n_{i0k}(\mathbf{V} = \mathbf{v})$ for the number of groundings of functor node f_i such that the ground node is assigned value k, and $p_{i0k}(\mathbf{V} = \mathbf{v})$ for the frequency of such groundings.

Examples. The following examples refer to the DB instance of Figure 1, where $\mathbf{V} = \mathbf{v}$ denotes the assignment of values to ground nodes specified in the database. We use a Prolog-style list notation for a conjunction of literals. An example of a family formula F_{ijk} with child node $f_i = gender(X)$ is

$$gender(X) = M, gender(Y) = W, Friend(X, Y) = T.$$

From Figure 2, the associated conditional probability is $\theta_{ijk} = 40\%$. The number of true groundings is $n_{ijk}(\mathbf{V} = \mathbf{v}) = 2$, and the number of possible groundings is 2×2. Therefore the database frequency of this formula is

$$p_{ijk}(\mathbf{V} = \mathbf{v}) = \frac{n_{ijk}(\mathbf{V} = \mathbf{v})}{4} = 1/2.$$

For the unspecified parent state, we have

$$F_{i0k} \equiv gender(X) = M.$$

The number of true groundings is $n_{i0k}(\mathbf{V} = \mathbf{v}) = 2$, and the frequency is $p_{i0k}(\mathbf{V} = \mathbf{v}) = 2/3$. A family formula with child node $coffee\_dr(X)$ is

$$coffee\_dr(X) = T, gender(X) = W.$$

The associated conditional probability parameter is 70%. The number of true groundings is 1, and the number of possible groundings 3. Therefore the database frequency of this family formula is 1/3.

Figure 1: A simple relational database instance.

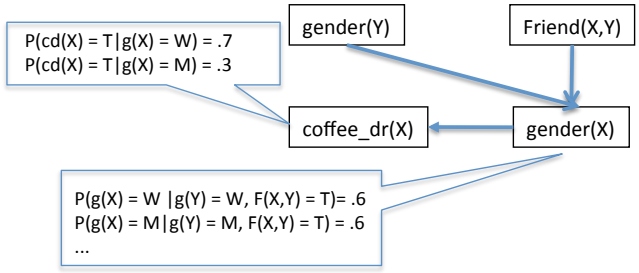

Figure 2: A Parametrized Bayes Net with some CP-table entries. CP-table entries are chosen for illustration and are not related to the data in Figure 1. For convenience, examples below use a uniform prior over the nodes $gender(Y)$ and $Friend(X, Y)$.

3.3 Model Conversions. A Bayes net can be converted to a Markov net through the standard **moralization** method: connect all co-parents, 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.

A PBN graph can be converted to a Markov Logic Network structure by moralization [2, 12.5.3], [22]: for each family state F_{ijk}, add a conjunction of literals that specifies the state.

Bayes nets can also be converted to dependency nets [6]. 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 (3.1).

4 Random Regression

Let $Y = f(a_1, \ldots, a_\ell)$ be a target ground node instantiating functor node $f(A_1, \ldots, A_\ell)$. The **regression graph** for Y is the partially ground PBN that results by substituting a_i for A_i in functor node Y *and* in its Markov blanket; see Figure 3. 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.

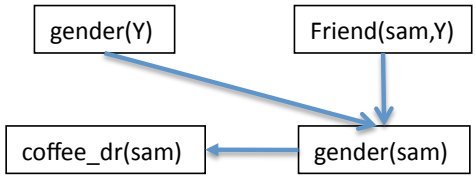

Figure 3: The regression graph for the target node $gender(sam)$ derived from the Bayes net of Figure 2 by substituting sam for X.

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 (3.1) 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 $ln(\tilde{P}^r(Y = y|\mathbf{X} = \mathbf{x}))$.

Including irrelevant predictors leads to bad predictions, so statistical-relational models restrict edges in the ground model to relevant predictors only [19]. In what follows, 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 provides a sample computation of a random regression for predicting the gender of Sam given the database instance of Figure 1. The next section relates random regression to log-linear models for relational prediction.

Table 1: Computing the random regression for target node value $gender(sam) = W$. 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
$Y = anna$	$P(cd(sam) = T\|g(sam) = W) = .7$	$P(g(sam) = W\|g(anna) = W, Fr(sam, anna) = T) = .6$	$ln(.7 \times .6) = -0.87$
Y=bob	$P(cd(sam) = T\|g(sam) = W) = .7$	$P(g(sam) = W\|g(bob) = M, Fr(sam, bob) = T) = .4$	$ln(.7 \times .4) = -1.27$
		Average	-1.07

5 Log-linear Relational Regression

The general form of a *log-linear regression equation* is

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

Frequency Model:
$\tilde{P}$(g(sam) = W|mb) =
70% x (60% x 40%)$^{1/2}$=
0.34 = exp(-1.07).

Figure 4: The computation of the unnormalized Markov blanket probability for the gender of Sam, for the count model (left) and the frequency model (right).

$$ln(\tilde{P}(Y = y|\mathbf{x})) = \sum_i w_i x_i.$$

In this section we consider how the predictor variables x_i may be derived from a Bayes net. In the next section we examine how the weight parameters w_i may be derived from a Bayes net. We consider two specific choices of predictor variables $\{x_i\}$.

The Frequency Model The predictor variables x_i are the *frequencies* p^Y_{ijk} of the Markov blanket states for the target node Y.

The Count Model The predictor variables are the *counts* n^Y_{ijk} of the Markov blanket states for the target node Y.

Figure 4 provides an example computation for frequency regression and for count regression.

5.1 Frequency Regression If the log-conditional probabilities from the Bayes net CP-table entries are used as weights, then the **frequency regression equation** is given by

(5.2)
$$ln(\tilde{P}(Y = y|\mathbf{X} = \mathbf{x})) = \sum_{ijk} p^Y_{ijk}(\mathbf{X} = \mathbf{x}, Y = y)\ ln(\theta_{ijk}).$$

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. For the unnormalized conditional log-likelihood of $gender(sam) = W$, random regression (Table 1) gives the same value $0.34 = e^{-1.07}$ as frequency regression. The next theorem shows that *the equivalence between frequency and random regression holds in all cases.* The proof is in the supplementary material.

THEOREM 5.1. *The frequency regression value for a target node (Equation (5.2)) equals the random regression value.*

5.2 Count Regression differs from frequency regression in that it uses the family counts n_{ijk} rather than the family frequencies p_{ijk}. The **count regression equation** is therefore given by

(5.3)
$$ln(\tilde{P}(Y = y|\mathbf{X} = \mathbf{x})) = \sum_{ijk} n^Y_{ijk}(\mathbf{X} = \mathbf{x}, Y = y)\ ln(\theta_{ijk}).$$

The count equation is closely related to Markov random fields, as follows. Consider the Parametrized Markov net M obtained by moralizing the PBN (Sec. 3.3). The count equation follows from applying the standard Markov field regression equation to the grounding of M [2]. In terms of the factor products defined by exponentiating the log-linear equations, the two models compare as follows. 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 4). Both regression models can be interpreted in terms of a dependency network that results from grounding a PBN model. Figure 5 summarizes the connections between graphical models and regression equations. In the next section we consider how to derive the weight parameters from given Bayes net parameters.

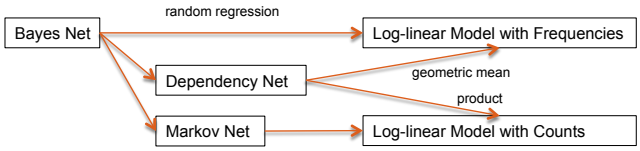

Figure 5: Connections between graphical models and log-linear regression equations.

6 Weight Learning With Bayes Net Parameters

For estimating the conditional probabilities, we use the empirical conditional frequencies observed in the database:

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

These estimates are well motivated by the following theoretical and practical considerations.

Maximum Likelihood. The random selection pseudo-likelihood for a Bayes net is the natural generative counterpart of random regression [21]. 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 [21, Prop.3.1].

Interpretability. The weight/clique potential parameters of undirected models are often difficult to interpret for users [18]. This is especially the case when weights are learned from data, with 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 of lifted learning scales well in both data size and the number of parameters in the model.

We investigate two methods for converting Bayes net conditional probabilities to log-linear weights w_{ijk}.

6.1 Log-conditional Probabilities as Weights. The first method uses the logarithm of the conditional probability of a child node value given values for its parents. In symbols:

$$w_{ijk} := ln(\theta_{ijk}).$$

Example. As in Section 3.2, consider the family state F_{ijk} with $\theta_{ijk} = 40\%$. In the Markov Logic network that results from converting the PBN, this parameter leads to the weighted conjunctive clause

$$g(X) = M, g(Y) = W, F(X,Y) = T; w = ln(40\%).$$

Other researchers have recommended log-conditional probabilities as weights [2], [1].

The Scale Problem. A problem with using log-conditional probabilities together with the count model is that features with more instances have exponentially more influence; see Figure 4. In a typical illustrative scenario, the regression target is the descriptive attribute of a single entity, (e.g., $gender(sam)$). One predictor feature represents an association between attributes of the same entity (e.g., $coffee\_dr(sam), gender(sam)$, whereas another represents an association between attributes of related entities and the single entity (e.g., $gender(sam), gender(Y), Friend(sam, Y) = T$). In this case the information from related entities has many instances, and tends to overwhelm that from the target entity, which has just one instance. In terms of a log-linear model, the count predictors are on different scales. *Frequency regression balances the impact of different features by scaling predictors to the common range [0,1].*

6.2 Log-differences as Weights. The second method uses the log-difference between the conditional probability of a child node value given values for its parents, and the prior unconditional probability of the child node value. A Bayes net implicitly defines a value θ_{i0k} for the marginal probability of node X_i taking on value k, which can be computed using a standard Bayes net query (or directly estimated from the data). The log-difference weight assignment is

$$w_{ijk} := ln(\theta_{ijk}) - ln(\theta_{i0k}).$$

Example. For the Bayes net of Figure 2, assume that conditional on X and Y *not* being friends, the distribution of $g(X)$ is uniform regardless of the value of $g(Y)$. For instance, $P(g(X) = W|F(X,Y) = F, g(Y) = M) = 1/2$. Then the marginal probability distribution that the Bayes net entails for $g(X)$ is also uniform, that is, $P(g(X) = M) = P(g(X) = W) = 50\%$. For each marginal probability assignment, the Markov Logic network that results from converting the PBN contains a unit clause with weight the log-marginal probability. In the example, there is a weighted clause

$$g(X) = M; w = ln(50\%).$$

For each family state F_{ijk}, the MLN contains a conjunctive clause, with the weight being the log-difference between the conditional and the marginal probabilities. In the example, there is a weighted clause

$$g(X) = M, g(Y) = W, F(X,Y) = T; w = ln(40\%) - ln(50\%).$$

Motivation and Discussion. The reason why log-difference addresses the scale problem is as follows. Compared to associations among the attributes of a single entity, we expect the associations with related entities to be relatively weak, because probabilistic dependencies in a network become weaker with distance. This means that the probability of a child value, given a parent condition specifying descriptive attributes of related entities, should be relatively close to the marginal probabilities of the child value. Therefore the log-difference should be relatively small, and so the log-difference method tends to assign smaller weights to formulas with many groundings. The supplementary material confirms this expectation with direct observations of the weight sizes assigned to different formulas. It also shows that weights found by Markov field optimization

methods are smaller for formulas with many groundings, which is further evidence that a scaling component is important for weights in the count model.

Our final section evaluates the different methods on learning time and predictive accuracy.

7 Performance Evaluation

We first discuss our comparison metrics, then the systems compared. 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 [8].

7.1 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 [2, 22]. 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 non-binary. We evaluate the learning methods using 5-fold cross-validation as follows. We formed 5 subdatabases for each database, by randomly selecting entities from each entity table, and restricting the relationship tuples in each subdatabase to those that involve only the selected entities (i.e., subgraph sampling [3, 22]). 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.

7.2 Comparison Methods.

7.2.1 Structure Learning. To obtain a Bayes net structure, we applied the learn-and-join algorithm to each database [22]. We then convert the PBN graph to a Markov Logic Network structure (see Section 3.3), declaring attribute predicates as functional, as recommended by the Alchemy Group [1]. 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 that target attributes, not links, following [9, 22].

7.2.2 Parameter Learning.

Bayes Net Methods. Pairing the two Bayes net weight types with the two predictor types yields 4 different log-linear regression models. Table 2 displays the pairings, with names for future reference.[1] The next proposition shows that *for the frequency model*, the log-conditional and the log-difference weights are equivalent. This result means that the extent to which the log-difference method addresses the scaling problem, is already built into the frequency model.

Table 2: Regression models + Bayes net weight conversion methods; the frequency models are mathematically equivalent to each other and to random regression.

Weights/Predictors	Count	Frequency
Log-conditional probabilities	log(cp)+count	log(cp)+freq
Log-differences	log-diff+count	—

PROPOSITION 7.1. *Frequency regression (Equation 5.2) returns the same result for log-conditional probability weights and for log-difference weights.*

Markov Net Methods. A Markov net model uses general weights w_{ijk}. To learn the w_{ijk} weights, we applied the default weight training procedure [13] of the Alchemy package [11]. (We added unit clauses for each node-value combination (cf. Section 6.2), as recommended by the Alchemy group.) We refer to this method as the **MBN** method, for "Moralized Bayes Net" [9]. MBN is the state-of-the-art method for log-linear prediction with Bayes nets [22].

7.2.3 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. For MBN we use the count inference model because Alchemy weight learning is optimized for this.

7.3 Databases. We used 5 benchmark real-world databases. For more details please see the references in [22] and on-line sources such as [8].

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

[1] Khosravi proposed subtracting the logarithm of the uniform distribution from the conditional probability [7]. This method is not theoretically equivalent to random regression. We found that using the marginal probabilities θ_{i0k} performs better than the uniform distribution on both accuracy and log-likelihood. Thus the marginal probabilities appear to be preferable both in terms of theoretical justification and in terms of predictive performance.

Table 3: A comparison of *runtime* (seconds) required for parameter learning with a fixed Bayes net structure. 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	709	2673	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

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 [22].

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.

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).

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

7.4.1 Learning Times. Table 3 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).

7.4.2 Predictive Accuracy. Table 4 compares the prediction scores of the methods. Figure 6 averages performance over all five databases to provide a simple visual summary of our findings. We first discuss the Bayes net methods, then compare them to Markov net weight learning.

Bayes Net Parameter Learning.

CLL. *Using frequencies rather than counts improves the conditional log-likelihood score for the log(cp) weights*, 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

Table 4: *Predictive accuracy* comparison of the Bayes net parameters (cp+) with the Markov net parameters (mbn), which are general weights. cnt/freq = count/frequency regression model.

CLL	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
log(cp)+cnt	-0.47 ± 0.10	-1.36 ± 0.11	-1.19 ± 0.07	-0.84 ± 0.03	-1.33 ± 0.07
log-diff+cnt	-0.42 ± 0.05	-1.36 ± 0.11	-1.10 ± 0.16	-0.77 ± 0.03	-1.20 ± 0.07
log(cp)+freq	**-0.41** ± 0.04	-1.34 ± 0.09	**-0.71** ± 0.01	**-0.73** ± 0.04	**-1.07** ± 0.10

Accuracy	UW	Mondial	MovieLens	Mutagenesis	Hepatitis
mbn	80.3% ± 0.05	43.8% ± 0.04	59.7% ± 0.02	61.5% ± 0.02	51.0% ± 0.02
log(cp)+cnt	78.3% ± 0.08	**44.7%** ± 0.04	64.3% ± 0.01	61.4% ± 0.05	49.2% ± 0.03
log-diff+cnt	80.9% ± 0.06	**44.7%** ± 0.04	61.9% ± 0.02	**67.0%** ± 0.03	**55.1%** ± 0.02
log(cp)+freq	**81.0%** ± 0.06	44.6% ± 0.04	**65.1%** ± 0.01	**67.0%** ± 0.03	54.8% ± 0.02

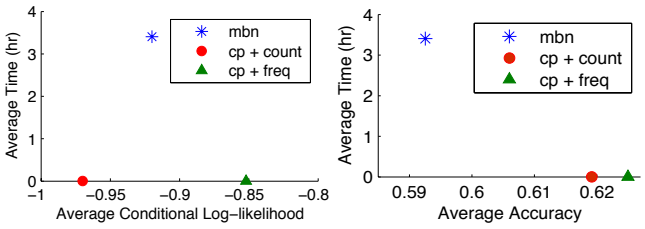

Figure 6: Overall predictive performance against weight learning time, averaged over all benchmark databases.

the balancing of factors to have more impact. The log-diff method too improves over the log(cp)+count model on all data sets, though not as much as frequency scaling. These findings support the importance of balancing predictor scales, either through rescaling (frequency) or adjusting weights (log-diff).

Accuracy. The Bayes net models have quite similar performance; the general ranking trend [log(cp) + freq] > [log-diff+ count] > [log(cp) + count] holds for this metric as well, with two exceptions: (1) On Hepatitis, the log-diff model has a slightly higher score than the frequency model (about 0.2%), and (2) on Movielens, the frequency model has a relatively large 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 parameters in combination with the frequency/random regression model are competitive with the optimized general weights.*

CLL. The CP+frequency model scores substantially better than the Markov net weights on Mutagen-

esis, 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%).

Together with our analysis of the scale balancing problem, the empirical findings make a good case for recommending the frequency model over the count model when the Bayes net parameters are used to derive log-linear weights.

8 Conclusion and Future Work

This paper considered an inference model for Bayes nets applied to relational 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 is a log-linear model, whose predictor variables are scaled to be frequencies in the range [0,1]. We compared random regression with standard log-linear models, using as weights both empirical conditional probabilities 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. Overall, random regression appears to be a principled, fast, and accurate model for relational prediction with Bayes nets.

References

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

[2] P. Domingos and M. Richardson, *Markov logic: A unifying framework for statistical relational learning*, in Introduction to Statistical Relational Learning [5].

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

[4] L. Getoor, N. Friedman, D. Koller, A. Pfeffer, and B. Taskar, *Probabilistic relational models*, in Introduction to Statistical Relational Learning [5], ch. 5, pp. 129–173.

[5] L. Getoor and B. Taskar, *Introduction to statistical relational learning*, MIT Press, 2007.

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

[7] H. Khosravi, *Fast parameter learning for Markov logic networks using bayes nets*, in ILP; Proceedings to appear, 2012.

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

[9] H. Khosravi, O. Schulte, T. Man, X. Xu, and B. Bina, *Structure learning for Markov logic networks with many descriptive attributes*, in AAAI, 2010, pp. 487–493.

[10] T. Khot, S. Natarajan, K. Kersting, and J. W. Shavlik, *Learning Markov logic networks via functional gradient boosting*, in ICDM, 2011, pp. 320–329.

[11] S. Kok, M. Summer, M. Richardson, P. Singla, H. Poon, D. Lowd, J. Wang, and P. Domingos, *The Alchemy system for statistical relational AI*, tech. rep., University of Washington., 2009. Version 30.

[12] D. Lowd, *Closed-form learning of Markov networks from dependency networks*, in UAI, 2012.

[13] D. Lowd and P. Domingos, *Efficient weight learning for Markov logic networks*, in PKDD, 2007, pp. 200–211.

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

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

[16] J. Neville and D. Jensen, *Relational dependency networks*, in An Introduction to Statistical Relational Learning [5], ch. 8.

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

[18] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, 1988.

[19] D. Poole, *First-order probabilistic inference*, in IJCAI, 2003, pp. 985–991.

[20] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum, *Classification with hybrid generative/discriminative models*, in NIPS, 2003.

[21] O. Schulte, *A tractable pseudo-likelihood function for Bayes nets applied to relational data*, in SIAM SDM, 2011, pp. 462–473.

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

[23] O. Schulte, H. Khosravi, and T. Man, *Learning directed relational models with recursive dependencies*, Machine Learning, (2012), p. Forthcoming.

[24] B. Taskar, P. Abbeel, and D. Koller, *Discriminative probabilistic models for relational data*, in UAI, 2002, pp. 485–492.