Log-Linear Inference Models for Bayes Nets Applied to Relational Data

Oliver Schulte, Hassan Khosravi, Tiaxiang Gao, and Yuke Zhu

School of Computing Science, Simon Fraser University,
Burnaby, B.C., Canada V5A 1S6, oschulte@cs.sfu.ca,hkhosravi@cs.sfu.ca

Abstract. Log-linear models are widely used for relational data for both generative and discriminative models [1, 2]. They use a weighted sum of variables to define a log-likelihood function, and are usually derived from Markov net models. In this paper we describe log-linear relational models derived from Bayes nets, where the regression weights are log-conditional probabilities. Log-linear Bayes nets are desirable because conditional probabilities have a natural interpretation and a scalable maximum likelihood solution for learning. On the inference side, previous models used instance counts of relational patterns (features) as variables. A known problem for relational regression models is that instance counts can be on very different scales (ill-conditioning) [2]. We introduce a new log-linear inference model where all variables are scaled to the [0,1] range by using frequencies, not counts, of relational patterns. We carried out an empirical comparison on five benchmark databases with (i) weights as log-conditional probabilities vs. (ii) general weights learned with Markov net methods. The conditional probability parameters 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

Relational data are very common, from enterprise relational databases to network data arising from the world-wide web or social media. Log-linear models are a prominent model class in machine learning that has been widely used with relational data. In a log-linear graphical model, a joint generative distribution is defined via a product of local factors, which is not necessarily normalized. If the model is used discriminatively to make predictions about the value of a specific variable/node, the *regression equation* for the unnormalized likelihood $\tilde{P}$ is

$$\tilde{P}(y|\mathbf{x}) = exp(\sum_i w_i x_i) \tag{1}$$

where $\tilde{P}(y)$ is the unnormalized probability of a target variable or node, and the x_i are values of all relevant predictors in the Markov blanket of y. The w_i are weight parameters representing the (log)-factors associated with the x_i.

Log-linear models are usually associated with undirected models, such as Relational Markov networks [1] and Markov Logic Networks [2]. An MLN is a set of weighted first-order formulas that compactly defines a Markov network comprising ground instances of logical predicates. In this paper we study log-linear models that are derived from directed Bayes net models rather than from Markov net models. Our motivation for introducing this model class is that (1) maximum likelihood estimation provides a fast and scalable basis for parameter learning, and (2) the parameters have a natural interpretation as conditional probabilities. We summarize the main features of the log-linear Bayes net models.

Parameter Space $\{w_i\}$*.* The characteristic feature of a Bayes net model is that its parameters are conditional probabilities of a child node value given an assignment of values to its parents. In the Bayes net log-linear model, the weights w_i are logarithms of these conditional probabilities. Equivalently, the regression equation is a product of conditional probability factors. We refer to models with weights derived from conditional probabilities as Bayes net or *CP* log-linear models. We refer to models without constraints on the weights as Markov net or general log-linear models.

A standard approach to defining an inference model for a 1st-order Bayes net is to view it as a template for a ground (unrolled) Bayes net [4]. However, grounding a 1st-order Bayes net often leads to a graph with cycles, which arise from recursive dependencies or autocorrelations, where the value of an attribute for an individual depends on the value of the same attribute for related individuals [5, 6]. While we use the parameters of the 1st-order Bayes net to specify weights for a CP model, inference in a CP model is defined by a log-linear formalism, not with respect to a ground acyclic Bayes net. Recursive dependencies are therefore handled in the same way as with other log-linear formalisms. We discuss interpretations for the CP models in terms of ground graphs in Section 5.2.

Predictor Space $\{x_i\}$*.* In most relational log-linear models, the predictive features x_i are defined by the *counts* n_i of relevant relational patterns defined by the model [1, 2]. For instance, to predict the intelligence y of a student, the model may consider how many A grades she has received, how many B grades, etc. A problem with counts is that features with more instances have exponentially more influence. For example, if the model considers the ranking of a student as well as grades, the grade factors overwhelm the ranking, because a student has just one ranking but many grades. Since in a Bayes net model, the weights are on the same scale (log-probability), smaller weights cannot sufficiently scale down the impact of predictors with larger domains. Therefore we investigate using feature *frequencies* f_i as predictors x_i, whose scale is [0,1]. In the intelligence prediction example, we use the percentage of A grades among all grades the student has received, the percentage of B grades, etc. The use of frequency predictors is equivalent to using the *geometric mean* rather than the simple product to combine factors.

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 [7, 8].

Evaluation. Our experiments use five benchmark databases. For each database, we learn a Bayes net structure, and evaluate four different combinations of parameter/predictor spaces for this fixed structure. For conditional probability parameters we use the maximum likelihood parameter settings (observed frequencies). This is compared to general log-linear weight parameters computed by an optimization routine, the default routine in the Alchemy system. The Alchemy package is a state-of-the-art open-source software system for MLNs [3]. Using conditional probabilities brings substantial scalability improvements: parameter learning takes seconds, while the Alchemy system requires hours on 3 out of 5 benchmark databases. The predictive performance of conditional probabilities, using frequency predictors, is competitive with the general log-linear model with optimized weights, if not superior. The count model performs worse.

Paper Organization. We describe further related work. Then we present background: basic relational graphical models and connections between them. The next section defines the frequency and count regression models. We discuss parameter estimation with conditional probabilities, and the interpretation of the regression models in terms of ground graphical models. Empirical evaluation compares the frequency and count models with optimized log-linear weights on a number of 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 that uses feature frequencies rather than counts, and an empirical comparison of the frequency and count models.
2. A theoretical interpretation of the frequency model in terms of random instantiations of the Markov blanket of the target node.
3. Experiments that indicate that the frequency model with maximum likelihood estimates is competitive with a general log-linear model with optimized weights.

2 Related Work

Moralization Methods. Several researchers have examined converting a Bayes net relational model to a Markov net, which defines a log-linear model. Richardson and Domingos propose converting a Bayes net to a Markov Logic network using moralization to convert the structure and log-conditional probabilities as clique potentials [2]. This is also the standard Bayes net conversion recommended by the Alchemy system [9]. The moralization method is equivalent to our log-linear model with counts. Khosravi et al. [10] follow the moralization approach for the model structure, but do not use log-probabilities as parameters for inference. Instead, they use MLN parameter learning methods to obtain weights. To our knowledge, our experiments are the first that evaluate the moralized Bayes net structure with log-probability weights.

Natarajan et al. [11] consider 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. Natarajan et al. show that for decomposable combining rules, the combining rule can be implemented using additional unobserved random variables ("multiplexers") [11]. The entire Bayes net structure with observed plus unobserved variables can then be converted to an MLN, which would appear to define a log-linear model in the augmented variable space. Our log-linear model uses only observed features specified in the original Bayes net model. Another difference is that with combining rules, there is no closed form for parameter estimation, so gradient descent methods are applied.

Scaling Predictors. Scaling predictors to the [0,1] range is a familiar technique in log-linear regression model [12]. 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 [2] and the random selection pseudo-likelihood [13]. 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 features that depend on the individual target node (scaling factor). Combining rules like the arithmetic mean [11] 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 combining rule applies only to the parents of the target node.

3 Background: Relational Graphical Models

We denote random variables by upper case letters such as X_i, Y_j. With respect to a graphical model, we interchangeably refer to its nodes and its variables. We consider only graphical models with discrete random variables. We use vector notation for lists of random variables 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

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. The unnormalized **Markov blanket classification equation** is given by

$$\tilde{P}(Y_i = y | \mathbf{X} = \mathbf{x}) = P(Y_i = y | \mathbf{pa}_i) \cdot \prod_{X_j \in \text{children}_i} P(X_j = y | \mathbf{pa}_j). \qquad (2)$$

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. For an assignment of values to all nodes in the Markov net, the joint probability of the values is given by the product of the associated clique potentials, divided by a normalization constant.

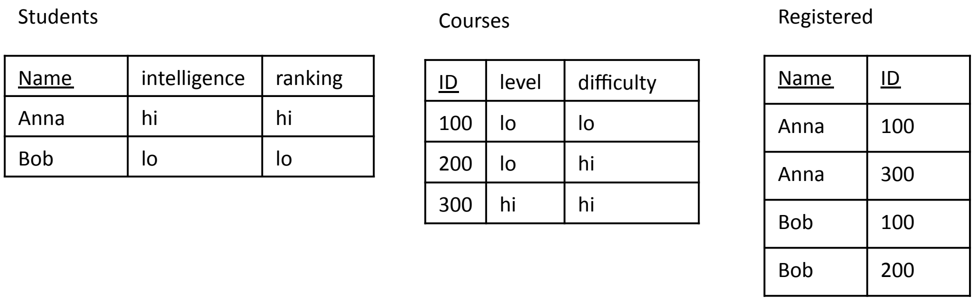
A **dependency network** structure is a directed graph; cycles are allowed [14, 15, 7]. The parameters are conditional probabilities of each node, given its *Markov blanket* (not just the parents). Dependency networks are like Markov networks in that conditional probabilistic independence corresponds 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 due to Poole [4]. 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 $f(\tau_1, \ldots, \tau_k) = f(\mathbf{X})$ where f is a functor and each τ_i is a first-order variable X_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 **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 atom**. A **population** is a set of individuals, corresponding to a domain or type in logic. Each first-order variable X is associated with a population. An **instantiation** or **grounding** for a set of variables $X_1, \ldots, X_k$ assigns to each variable X_i a constant from the population of X_i.

A **Parametrized** (Bayes, Markov, Dependency) Network is a (Bayes, Markov, Dependency) Network whose nodes are functor nodes. We usually omit the prefix "Parametrized". A **ground** graph is derived from a database and a network

[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.



Students

Name	intelligence	ranking
Anna	hi	hi
Bob	lo	lo

Courses

ID	level	difficulty
100	lo	lo
200	lo	hi
300	hi	hi

Registered

Name	ID
Anna	100
Anna	300
Bob	100
Bob	200

Fig. 1. A simple relational database instance.

by instantiating the adjacencies in the Parametrized network with all possible groundings. Figure 1 shows a simple database instance. A database instance specifies a unique value for each ground node; we denote such a joint assignment by $\mathbf{V} = \mathbf{v}$. 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).

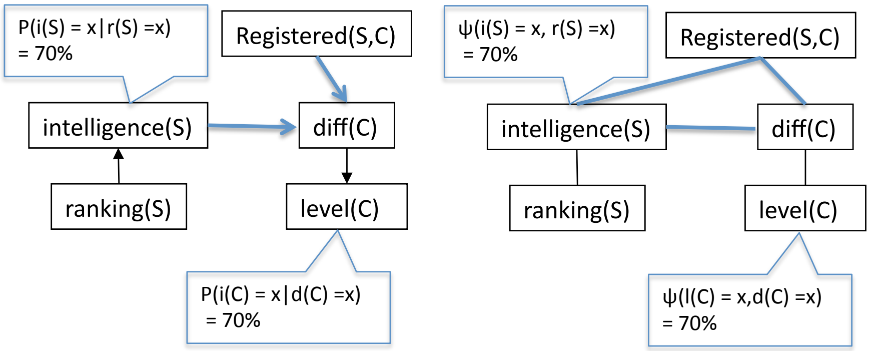


Fig. 2. Left: A Parametrized Bayes Net with some CP-table entries. $x = \{hi, lo\}$. Right: A Parametrized Markov Net, obtained by moralization.

Recursive dependencies (autocorrelations) are represented in a PBN by "copies" of the functors. Thus the structure $\mathit{gender}(X) \rightarrow \mathit{gender}(Y) \leftarrow \mathit{Friend}(X, Y)$

represents an association between the gender of a user and that of his/her friends. In this case we assume that the Bayes net is in main functor format [6]: for each functor f, there is a main functor node that is the only node with f with parents. In the example, $gender(Y)$ is the main functor, and $gender(X)$ 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 Parametrized BN B not in main functor format, there is an equivalent main functor Bayes net B' that has the same ground graph [6].

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 [2]. 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. If $M(B)$ is a Parametrized Markov net obtained from PBN B, the unnormalized likelihood function for the ground graph of $M(B)$ [2] is given by

$$P_{M(B)}(\mathbf{V} = \mathbf{v}) = exp\left(\sum_{ijk} n_{ijk}(\mathbf{V} = \mathbf{v}) \cdot ln(\theta_{ijk})\right). \tag{3}$$

In terms of the Bayes net parameters, Equation (3) is simply the product of all conditional probabilities defined by a ground child-parent instance. Bayes nets can also be converted to dependency nets [14]. First, for each node X_i, add a directed edge $X_j \rightarrow X_i$ from each node X_j in the Markov blanket of X_i. The resulting graph is the same as the moralized Bayes net structure but with bidrected rather than undirected adjacencies. The conditional probability parameters are given by the Markov blanket equation (2).

4 Parameter Space: the w_i

The characteristic feature of a Bayes net log-linear model is that a weight w_i is a log-conditional probability of a child node values given an assignment of values to its parents. While in relational data, the restriction to conditional probabilities incurs a loss of expressive power, we believe that conditional probability parameters are well motivated by the following advantages for interpretability and scalability.

Interpretability. The weight/clique potential parameters of undirected models are often difficult to interpret for users [16]. 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. Fast Bayes net parameter estimates can be obtained by using the observed conditional frequencies, which are defined by

$$\widehat{\theta}_{ijk} = \frac{n_{ijk}(\mathbf{V} = \mathbf{v})}{\sum_k n_{ijk}(\mathbf{V} = \mathbf{v})}. \tag{4}$$

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.

Fast conditional probability estimates can be combined with general weight learning in at least two important ways. (1) Conditional probabilities can be used to calculate *initial weights* for a local search procedure. (2) Several problems require *repeated parameter estimation*, for instance structure learning, or missing data imputation (e.g., via the EM algorithm). In this case fast parameter estimates can be used to quickly approach a solution, for instance a good structure.

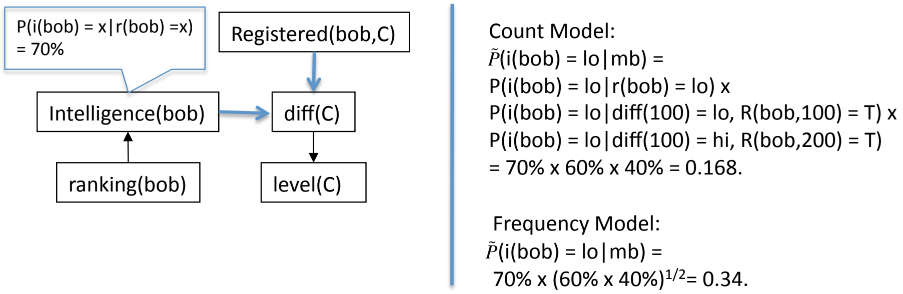
5 Predictor Space: the x_i

A standard way to define a log-linear model for a Bayes net is to convert it to a Markov network via moralization, and then use the log-linear model defined by the Markov network [2, 10]. The regression equation for this model is as follows. Let $Y = f(\mathbf{a})$ be a target ground node instantiating functor node $f(\mathbf{A})$. The **regression graph** for Y is the partially ground PBN B_Y that results by substituting a_i for A_i in functor node $f(\mathbf{A})$ and its Markov blanket. This is illustrated in Figure 3. If there is more than one functor node with f, we use the main functor node (Sec. 3.2). Using the notation from Section 3.2 *with the regression graph*, the regression equation for target node Y is given by

$$\tilde{P}(Y = y|\mathbf{X} = \mathbf{x}) = exp\left(\sum_{ijk} n_{ijk}(\mathbf{X} = \mathbf{x}, Y = y) \cdot ln(\theta_{ijk}).\right) \tag{5}$$

Here the summation is over the Markov blanket of the target node in the regression graph, that is, the index i ranges over the target node and its children. Including irrelevant predictors in the regression leads to bad predictions, so statistical-relational models restrict edges in the ground model to relevant predictors only [4]. In our examples and experiments below, we take the relevance conditions to be the existence of a link, such that for two entities a, b, there is a directed edge from a ground node representing an attribute of a to a ground attribute of b, only if a link exists between the two entities. Figure 3 illustrates the resulting computation for predicting the intelligence of Bob given the database instance of Figure 1.

A problem with the count regression equation (1) is that Markov blanket components with many groundings have exponentially more influence. To balance the scales of the predictor variables, we use the frequency of a factor rather



Count Model:
$\tilde{P}$(i(bob) = lo|mb) =
P(i(bob) = lo|r(bob) = lo) x
P(i(bob) = lo|diff(100) = lo, R(bob,100) = T) x
P(i(bob) = lo|diff(100) = hi, R(bob,200) = T)
= 70% x 60% x 40% = 0.168.

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

Fig. 3. Left: The regression graph for target $intelligence(bob)$. Right: The computation of the unnormalized Markov blanket probability, for the count model (top) and the frequency model (bottom). The example uses the conditional probability $P(d(C) = x|i(S) = x, Registered(S,C) = true) = 60\%$. The frequency model assigns higher weight to the ranking of *bob*.

than its count as a predictor. In terms of factor products, this corresponds to using the geometric mean rather than the simple product. The corresponding regression equation is as follows. Let m_i denote the number of possible groundings of family formula F_{ijk} in the regression graph. Note that m_i does not depend on j or k. The frequency regression equation is then given by

$$\tilde{P}(Y = y|\mathbf{X} = \mathbf{x}) = exp\left(\sum_{ijk} \frac{n_{ijk}(\mathbf{X} = \mathbf{x}, Y = y)}{m_i} \cdot ln(\theta_{ijk})\right). \quad (6)$$

Figure 3 illustrates the computation. We next establish a random selection interpretation of the frequency model.

5.1 Random Selection Interpretation.

The frequency regression value can be interpreted as an expectation over random instantiations of the Markov blanket as follows.

1. Let $X_1, \ldots, X_k$ be a list of *all* 1st-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 variable X_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 Markov blanket equation (2) 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 $ln(\tilde{P}^r(Y = y|\mathbf{X} = \mathbf{x}))$.

Table 1. Computing the random regression for target node *intelligence*(bob). Each course selection defines an instantiation of the Markov blanket of the target node with two associated factors. The random regression value -1.07 is the log-average. Notice that $-1.07 = ln(0.34)$, the log of frequency regression (Figure 3).

Grounding	Factor 1	Factor 2	Log-Product
$C = 100$	$P(i(bob) = lo \mid r(bob) = lo)$ $= .7$	$P(i(bob) = lo \mid diff(100) = lo,$ $R(bob, 100) = T) = .6$	$ln(.7 \times .6) =$ -0.87
C=200	$P(i(bob) = lo \mid r(bob) = lo)$ $= .7$	$P(i(bob) = lo \mid diff(200) = hi,$ $R(bob, 200) = T) = .4$	$ln(.7 * .4) = -1.27$
		Average	-1.07

Random regression can be modified for relevance restrictions by selecting only relevant predictors, as the example computation in Table 1 illustrates. In the example, frequency regression and random regression return the same value. The next proposition establishes this identity in general. Thus the frequency regression equation can be viewed as a closed form for computing the random regression value. We omit the proof due to space constraints. We remark that this result applies to any graphical model based on a 1st-order template, not only Bayes nets.

Proposition 1. *The frequency regression value for a target node (Equation* (6)*) equals the random regression value.*

5.2 Graphical Interpretations.

Because recursive dependencies lead to cycles in the grounding of the Parametrized Bayes net, the regression equations cannot be derived from a ground Bayes net in general. However, the conversion approach (Section 2) offers an interpretation in terms of a Markov net resp. dependency net derived from the Bayes net.

Count Model. The count regression formula (5) can be derived from the generative likelihood Equation 3 for Markov networks [2]. In terms of the ground moralized Markov net, Equation (5) is the product of all clique potentials for every clique containing the target node, hence the correct (unnormalized) regression equation for this model. The empirical frequencies θ_{ijk} are the maxima of the unnormalized generative likelihood equation for the moralized Bayes net. Normalization is necessary for inference, but not related to how well the model fits the fact in a database. We omit the proof due to space constraints.

Proposition 2. *Let B be a parametrized Bayes net and let $\mathbf{V} = \mathbf{v}$ denote a set of observations for a relational data structure that specify values for the nodes in the ground Bayes net. The empirical conditional frequencies (Equation 4) maximize the unnormalized likelihood (Equation 3) for the moralized Bayes net.*

Frequency Model. Although the frequency equation (6) is very similar to the count equation, it does not have an interpretation in terms of a Markov random field. The reason is that the scaling factor m_i depends on the target node. For instance, if the target node is the intelligence of a student, the scaling factor is the number of courses the student has taken. If the target node is the difficulty of the course, it is the number of students in the course. This means that the model does not use a single factor/potential for all queries, but instead scales the factor depending on the query. However, Equation 6 gives the Markov blanket conditional probabilities for all ground nodes, so it defines a dependency network whose structure is obtained by converting the Bayes net graph to a dependency graph (Sec. 3.2).

There does not seem to be a closed form solution for the frequency-model likelihood maxima. In the experiments below, we use the observed conditional frequencies as with count regression. One view of this choice is that the frequencies are a heuristic, as used by [17] for discriminative BN learning. A more sophisticated view is that since a Bayes net is a generative model, the generative likelihood equation (3) is the proper objective for parameter optimization. The scaling factors are not included in the objective function for *learning*, but are added to the *inference* model after parameters have been learned.

Inference. For the count model, Markov logic network inference algorithms can be used after moralization, as in [10, 18]. For the frequency model, Heckerman et al. [14] show that applying Gibbs sampling to a dependency network defines a stationary joint distribution, hence can be used to answer general queries. Their ordered pseudo-Gibbs sampler has been lifted to the relational setting [5]. A Gibbs sampler for the frequency model would be able to exploit the ordering constraints provided by the Bayes net structure. Since the form of the frequency model is very similar to that of the count model, an alternative is to adapt MLN inference methods developed for the count model.

6 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 [10] and on-line sources such as [19].

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 [10].

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 [20], 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).

6.1 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 [21, 10]. 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 [10]. 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.

6.2 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 [19]. We applied the learn-and-join algorithm to learn a Bayes net structure for each database, which is the start of the art structure learning algorithm for Parametrized Bayes Nets [10]. 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 [10, 6].

Parameter learning for general weights proceeds in two steps as in [10]: (1) Convert the Parametrized Bayes nets to Markov Logic Networks, using moralization, which adds a conjunctive clause for each family state F_{ijk}. We declared attribute predicates as functional as recommended by the Alchemy Group [9]. (2) The moralized BN Equation 3 is the likelihood function for the MLN, but with general weights w_{ijk} in place of $ln(\theta_{ijk})$. To learn the w_{ijk}, we applied the default weight training procedure [22] of the Alchemy package [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. We compared the following four approaches.

MBN+Count The Bayes net structure is converted to an MLN using moralization, weights learned using Alchemy [10]. Inference uses count regression.

MBN+Frequency Same as the previous method, but using frequency regression for inference.

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.

6.3 Results.

All results are averages from 5-fold cross validation, over all attributes in the database. Table 2 compares the accuracy score of the methods. Table 3 compares the log-likelihood score of the methods. We first discuss the comparison of the frequency vs. count models, and then the comparison of the CP models with the Markov net models.

Table 2. Accuracy score of the Bayes net parameters (cp+), which are conditional probabilities, with the Markov net parameters (mbn+), which are general weights. Accuracy is the percentage of correctly predicted values in the test data.

Method	UW	Mondial	MovieLens	Mutagenesis	Hepatitis
mbn+count	80.25% ± 0.05	43.81% ± 0.04	59.71% ± 0.02	61.49% ± 0.02	51.01% ± 0.02
mbn+freq	80.25% ± 0.05	43.81% ± 0.04	58.76% ± 0.02	60.89% ± 0.03	50.94% ± 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

Table 3. Conditional log-likelihood comparison of the Bayes net parameters (cp+) with the Markov net parameters (mbn+).

Method	UW	Mondial	MovieLens	Mutagenesis	Hepatitis
mbn+count	-0.44 ± 0.07	**-1.28** ± 0.07	-0.79 ± 0.03	-0.91 ± 0.09	-1.18 ± 0.26
mbn+freq	-0.43 ± 0.07	**-1.28** ± 0.07	-0.83 ± 0.03	-0.93 ± 0.13	-1.16 ± 0.21
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

Frequency vs. Count Model.

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.

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.

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. Figure 4 examines the scaling components of the weights directly. Every dataset shows scaling effects except for UW, where all methods achieve the same CLL score. The scaling effects are especially strong for MovieLens, where the Bayes net frequency model outperforms the count model the most.

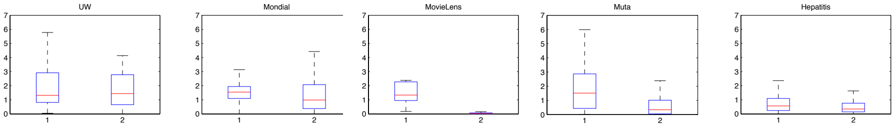


Fig. 4. Boxplots of the absolute weight sizes learned by the Markov Logic Network method. The median weight size is shown for the set of clauses with one 1st-order variable (left), and the set of clauses with two 1st-order variables (right). Smaller weights for 2-variable clauses balance their larger number of groundings against the smaller number for one-variable clauses.

Bayes net vs. Markov net parameters.

Accuracy. The conditional probability parameters have a slightly higher score than the Markov net methods, with the biggest differences on MovieLens (5%) and Hepatitis (4%).

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

Learning Times. Table 4 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).

Experimental Conclusions. Together with our theoretical analysis, the empirical findings make a strong case for recommending the frequency model over the count model when the CP parameters are used. The performance of the CP frequency model compared to general log-linear weight learning is quite impressive across databases and attributes. Therefore, log-linear models derived from Bayes nets appear to offer a good baseline model within the class of log-linear relational models. One of the reasons for the widespread popularity of Bayes nets for nonrelational data is that parameters have a natural interpretation and high-quality estimates can be obtained quickly. We believe that providing users with

Table 4. 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

a relational model class that has similar advantages will encourage applications of statistical-relational learning. Users have the option to carry out an initial data exploration and deploy more complex methods if the results are promising.

7 Conclusion and Future Work

This paper considered log-linear inference models for applying Bayes nets to relational data. The characteristic feature of such models is that the weight parameters are log-conditional probabilities of parents given children. The predictor variables in previous relational log-linear models are instance counts of relational patterns. We provided theoretical considerations and empirical evidence that for conditional probability parameters, it is important to rescale the predictors to be instance frequencies. The frequency model can be interpreted as the expected log-linear regression value from a random instantiation of a node's Markov blanket. Using the maximum likelihood values as Bayes net parameters is much faster than optimizing weights using standard Markov Logic methods, typically seconds vs. hours. The predictive performance of log-conditional probability weights is competitive, on several datasets it was in fact superior.

Parametrized Bayes nets have been extended with decision trees to obtain more compact models of the conditional distribution of a child node given its parents [23]. This is a natural extension for testing the frequency model; we hypothesize that frequency scaling is even more important, because decision tree pruning leads to more variation in clause length. The powerful and effective technique of functional gradient boosting [8] could be applied to learning tree models that augment a Bayes net; gradient boosting is well-suited to learning potential functions for log-linear models. In sum, among graphical relational models, log-linear models based on Bayes nets offer an attractive trade-off between expressiveness vs. interpretability and scalability.

References

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

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

[3] 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.

[4] Poole, D.: First-order probabilistic inference. In Gottlob, G., Walsh, T., eds.: IJCAI, Morgan Kaufmann (2003) 985–991

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

[6] Schulte, O., Khosravi, H.: Learning directed relational models with recursive dependencies. In: ILP. (2011)

[7] 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

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

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

[10] 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

[11] 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

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

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

[14] 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

[15] Neville, J., Jensen, D.: Relational dependency networks. [24] chapter 8

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

[17] Grossman, D., Domingos, P.: Learning Bayesian network classifiers by maximizing conditional likelihood. In: ICML, New York, NY, USA, ACM (2004) 46

[18] Schulte, O., Khosravi, H.: Learning directed relational models with recursive dependencies. Machine Learning (2012) Forthcoming. Extended Abstract.

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

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

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

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

[23] Khosravi, H., Schulte, O., Hu, J., Gao, T.: Learning compact markov logic networks with decision trees. In: ILP. (2011)

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