A Log-Linear Model for Bayes Nets Applied to Relational Data

Abstract

We describe a new log-linear multi-relational model for directed graphs (Bayes nets) that provides fast parameter learning and accurate predictions. Log-linear models are widely used for multi-relational data. They are usually associated with undirected graphical models (Markov nets) [Taskar *et al.*, 2002; Domingos and Lowd, 2009]. The key new feature of our model is that it uses the frequencies of multi-relational features as predictors; previous models use feature counts. Frequencies scale counts by dividing by the size of the relevant relational neighborhood. Our experiments show that frequencies provide substantially more accurate predictions than counts. We provide a novel sampling semantics for the frequency model, in terms of a random instantiation of a local conditional probability model. We compared Bayes net learning with our model on five benchmark databases with state-of-the-art Markov net methods (Alchemy weight learning, MLN-Boost). Bayes net learning is fast; parameter learning took seconds vs. hours. The predictive accuracy of the Bayes net log-linear models was competitive, in most cases superior.

1 Introduction

Multi-relational data are very common, from enterprise databases to network data from the world-wide web and social media. This paper presents a new log-linear multi-relational model based on Bayes nets, that has good predictive performance and permits highly scalable parameter learning. Log-linear models are a prominent model class that has been widely used in statistical-relational learning [Sutton and McCallum, 2007]. They are usually associated with undirected models, such as Relational Markov networks [Taskar *et al.*, 2002] and Markov Logic Networks [Domingos and Lowd, 2009].

In this paper we focus on log-linear models for the conditional probability of a target node, given an assignment of values to the Markov blanket of the target node. In the terminology of dependency networks [Heckerman *et al.*, 2000], such models are referred to as *local probability distributions*. Local distributions are well-defined even with cyclic dependencies, which are common in relational data [Neville and Jensen, 2007]. Gibbs sampling can be used to extend the local distributions to a joint distribution [Heckerman *et al.*, 2000; Natarajan *et al.*, 2012; Lowd, 2012]. A log-linear local distribution takes the form

$$P(y|\mathbf{x}) = \frac{1}{Z}\exp(\sum_i w_i x_i).$$

A propositional Bayes net (BN) has a log-linear local distribution [Russell and Norvig, 2010, Ch.14.5.2]: (1) The weights $\{w_i\}$ are logarithms of the Bayes net parameters, the conditional probabilities of a child value given an assignment of parent values. (2) The predictors $\{x_i\}$ indicate which joint state of a parent-child family holds, for the target node and each of its children. We lift the log-linear propositional BN model to relational data in a new way by using the *frequencies* of family states. For instance, to predict whether a social network user drinks coffee, the predictors may include what percentage of the user's friends drink coffee. Previous log-linear models for multi-relational data have used feature instantiation counts rather than frequencies. The major claim of this paper is that, when used with Bayes net parameters, feature frequencies produce substantially more accurate predictions than feature counts. We describe theoretical considerations and experimental results to support this claim. We also provide a novel semantics for the frequency model: it is equivalent to the expected value of a random instantiation of the propositional BN local distribution model.

Motivation. *Predictor Variables.* Frequencies work better than counts because they address the *scale balance problem*: In count models, features with more instances have exponentially more influence. For example, if the model considers the gender of a user as well as features of her friends, the information from the friends overwhelms the gender, because the user has many friends but only one gender. Using feature frequencies as predictors x_i puts all predictors on the common scale [0,1]. While changing from counts to frequencies is a small step in the form of the regression equation, it leads to a big jump in predictive performance because of the balance problem. *Weight Parameters.* Parametrizing the model in terms of Bayes net conditional probabilities has important advantages for scalability (closed-form parameter learning) and interpretability.

Contributions and Significance. The main contributions are as follows.

1. A new log-linear regression model for Bayes nets that uses feature frequencies rather than counts, and log-conditional probabilities as regression weights.
2. A theoretical interpretation of the frequency model in terms of random instantiations of the Markov blanket of the target node.

We view deriving weights from Bayes net parameters as a strong baseline method that quickly produces an accurate model. Benefits to statistical-relational learning include the following. (1) Researchers can use the baseline to evaluate how computational cost trades off against improved performance. (2) Users can easily interpret the model for initial data exploration, and apply more complex methods if the initial results are promising. (3) Current relational regression models have difficulty scaling to medium-sized datasets, especially those with many descriptive attributes. Our work extends the practical applicability of relational learning to such datasets. (4) While we focus on Bayes nets, the scale balance problem arises also for other relational models. Our solution of changing the predictor space from counts to frequencies applies to log-linear models in general.

Paper Organization. We describe further related work. Then we present background on relational Bayes net models. The next section defines the frequency and count local probability models. Then we discuss their semantics. Empirical evaluation on five benchmark databases compares the frequency and count models, and the state-of-the-art MLN-Boost learner [Khot *et al.*, 2011].

2 Related Work

Parameter Space. To our knowledge, ours is the first implemented system that uses Bayes net parameters in a log-linear relational model. Schulte and Khosravi (2012) convert the Bayes net structure to a Markov Logic network (MLN) using moralization, which defines a log-linear model. They do not apply the BN parameters, but use MLN methods to learn general regression weights; we include this method in our experiments. Khosravi (2013) proposed using the BN parameters scaled by the uniform distribution over prior node values; this performs worse than our model. Both of these approaches use feature counts not frequencies. Natarajan *et al.* (2010) consider moralization with Bayes nets that have been augmented with combining rules (for mapping probabilities obtained from multiple parent instances to a single one). They show that for decomposable combining rules, the combining rule can be implemented using additional unobserved random variables ("multiplexers"). We consider tabular Bayes nets whose parameters are CP-table entries only.

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 [Natarajan *et al.*, 2010] similarly combine global parameters with a local scaling factor. Our frequency model applies the *geometric mean* rather than the arithmetic mean. To our knowledge, the geometric mean has not been used before in relational prediction models. Another difference with combining rules is that we apply the geometric mean in the entire Markov blanket of the target node, whereas usually a combining rule applies only to the parents of the target node. It is important to apply combining to the entire Markov blanket to balance the scale of regressors: if the children of the target node have many instantiations, and the parents have few, information from the children tends to overwhelm that from the parents, regardless of what rule is specified for combining/aggregating the parent instances. Relational Dependency Networks apply aggregation functions to the entire Markov blanket of a target node [Neville and Jensen, 2007].

Regressor Scaling. Variants of the Markov net pseudo-likelihood have been proposed that include scaling factors [Domingos and Lowd, 2009; Schulte, 2011]. 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*.

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., $P(X_1 = x_1, \ldots, X_n = x_n) \equiv P(\mathbf{X} = \mathbf{x})$. We consider graphical models with discrete random variables only. A Bayes net (BN) is a pair $\langle G, \boldsymbol{\theta}_G \rangle$ where (1) G is a directed acyclic graph, (2) $\boldsymbol{\theta}_G$ 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

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

Here 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 $MB(Y)$ of a node Y comprises the set of Y's children, parents and co-parents (nodes that share a child with node Y). The standard Bayes net formula for the local conditional distribution $P(Y = y|\mathbf{X} = \mathbf{x})$ of a target node Y is the product of: the probability that $Y = y$ given the state of its parents, and the probability of each child node value given the state of its parents, where Markov blanket values and states are specified in the assignment $\mathbf{X} = \mathbf{x}$. Written on a log-scale, this leads to the **Bayes net regression equation**

$$ln(P(Y = y|\mathbf{X} = \mathbf{x})) = \sum_{ijk} x_{ijk}\ ln(\theta_{ijk}) - ln(Z) \quad (3.1)$$

where the index i ranges over the target node and its children. The indicator variable $x_{ijk} \in \{0, 1\}$ takes on the value 1 iff the joint assignment $Y = y, \mathbf{X} = \mathbf{x}$ specifies value k for node i and state j for the parents of i.

To adapt Bayes nets for relational data, we follow the original presentation of Parametrized Bayes Nets (PBNs) due to Poole 2003. 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 $f(\tau_1, \ldots, \tau_\ell)$ 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 contains no variable, it is a **ground node**. 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 net** is a Bayes net 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 their friends [Schulte *et al.*, 2012].

A database instance specifies a unique value for each ground node. 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} is the number of groundings of F_{ijk} that evaluate as true for a given database instance.
- p_{ijk} is the frequency of the family state in the database, that is, the number of true groundings n_{ijk}, over the number of possible groundings.

While the quantities n_{ijk} and p_{ijk} depend on the data, we simplify notation by not showing this dependence symbolically.

Examples. The following examples refer to the DB instance of Figure 1. 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 $n_{ijk} = 2$, and the number of possible groundings is 2×2. Therefore the formula's database frequency is $p_{ijk} = n_{ijk}/4 = 1/2$.

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 is 1/3.

4 Log-linear Relational Regression

Given a PBN, 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 4. A key difference between propositional and relational data is that a Markov blanket state can be instantiated more than once in the relational neighborhood of the target node. Accordingly, we consider two

People

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

Friend

Name1	Name2
Anna	Sam
Sam	Anna
Sam	Bob
Bob	Sam

Figure 1: A simple relational database instance.

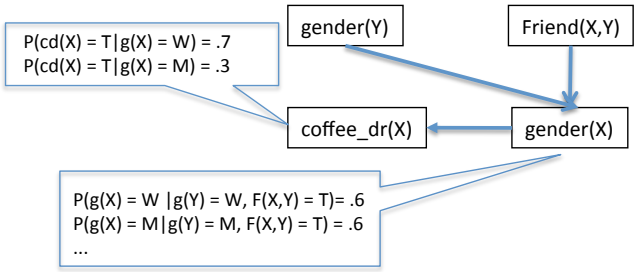

Figure 2: A Parametrized Bayes Net with some CP-table entries. CP-table entries are chosen for illustration and are not related to the data in Figure 1. For convenience, examples below use a uniform prior over the nodes $gender(Y)$ and $Friend(X, Y)$.

choices of predictor variables $\{x_i\}$ for lifting the propositional BN local model (3.1) to the relational case: In the **frequency model**, the predictor variables x_i are the *frequencies* p^Y_{ijk}. In the **count model**, the predictor variables are the *counts* n^Y_{ijk}. Here and elsewhere the superscript Y indicates that the notation is used with reference to the regression graph for target node Y.

If the BN log-conditional probabilities are used as weights, then the **count regression equation** is given by

$$ln(P(Y = y|\mathbf{X} = \mathbf{x})) = \sum_{ijk} n^Y_{ijk}\, ln(\theta_{ijk}) - ln(Z). \tag{4.2}$$

Equation (4.2) has a straightforward semantics as a grounding of the propositional BN regression equation (3.1): After instantiating the regression graph with each possible grounding, apply the BN formula to each ground instance of a conditional probability term. The **frequency regression equation** is given by

$$ln(P(Y = y|\mathbf{X} = \mathbf{x})) = \sum_{ijk} p^Y_{ijk}\, ln(\theta_{ijk}) - ln(Z). \tag{4.3}$$

Including irrelevant predictors leads to bad predictions, so like other statistical-relational models, we only consider instances that are linked to the target node by some relationship type (i.e., we do not consider negated links). Figure 3 provides example computations. The example illus-

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

Figure 3: The computation of the local regression probability for the gender of Sam.

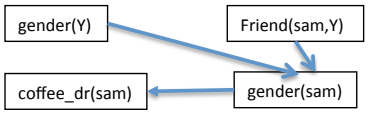

Figure 4: The regression graph for the target node $gender(sam)$ derived from the Bayes net of Figure 2 by substituting sam for X.

trates how frequency regression addresses the scale imbalance problem.

The frequency equation sums the log-averages of the groundings of each conditional probability term. We introduce a novel semantics to relate this to the propositional BN local distribution model: Whereas the count equation results from applying the propositional formula (3.1) to a *complete* grounding of the regression graph, the frequency model results from applying the propositional formula to a *random* grounding of the regression graph.

5 Random Selection Interpretation.

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 (3.1) to compute a log-sum for the random instantiation $\mathbf{a_i}$. The *expected value* of this log-sum is the **random regression** value $ln(P^r(Y = y|\mathbf{X} = \mathbf{x}))$.

Table 1 provides a sample computation of a random regression for predicting the gender of Sam given the database instance of Figure 1. Applying random regression to $gender(sam) = W$ (Table 1) gives the same value as frequency regression, namely $-1.07 = ln(0.34)$. The next theorem shows that *the equivalence between frequency and random regression holds in all cases.* The proof is omitted due to space constraints.

THEOREM 5.1. *The frequency regression value for a target node (Equation* (4.3)*) equals the random regression value.*

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

Our experiments evaluate the different methods on learning time and predictive accuracy.

6 Empirical Evaluation

We performed extensive experiments; due to space constraints, we describe some key findings and summarize others. 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 (reference omitted for blind review).

6.1 Comparison Methods.

Structure Learning. The learn-and-join algorithm is the state-of-the-art Bayes net structure learning algorithm for relational data [Schulte and Khosravi, 2012]. To obtain a Bayes net structure, we applied the learn-and-join algorithm to each database.

Parameter Learning. For estimating BN conditional probabilities, we use the empirical conditional frequencies observed in the database:

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

A general log-linear model uses weights w_{ijk} in place of $ln(\theta_{ijk})$. To learn the w_{ijk} weights, we applied the default weight training procedure of the Alchemy package [Kok *et al.*, 2009] using the same moralization procedure as in [Domingos and Richardson, 2007, 12.5.3], [Schulte and Khosravi, 2012]: convert a parametrized BN to an MLN structure that contains, for each family state F_{ijk} in the BN, a conjunction of literals that specifies the state. It is readily seen that the regression equation for the resulting MLN is the count equation [Schulte, 2011]. Thus weights learned by Alchemy can be used for count regression. Our comparison methods are the following.

MBN Converts the Bayes net structure to an MLN using moralization. Learns weights using Markov net methods [Lowd and Domingos, 2007]. Uses count regression for inference.

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

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

We employ exact inference rather than approximate inference (e.g., MC-SAT) to avoid conflating the impact of the model choice with the impact of the inference implementation. We conducted experiments with MC-SAT and the results were similar.

6.2 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 [Domingos and Richardson, 2007; Schulte and Khosravi, 2012]. 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, and the standard deviation. 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 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 [Frank, 1977; Schulte and Khosravi, 2012]). The models were trained on 4 of the 5 subdatabases, then tested on the remaining fold.

6.3 Databases. We used 5 benchmark real-world databases. For more details please see the references in [Schulte and Khosravi, 2012].

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 [Schulte and Khosravi, 2012].

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).

Table 2: A comparison of structure + parameter learning time (seconds). Database sizes are specified by the number of tuples.

Dataset	Bayes Net (s)	Markov Net (s)	#Ground atoms (s)	#tuples	#Parameters
UW	**36+2**	36+5	2673	709	125
Mondial	**12+3**	12+90	2234	870	575
MovieLens	**72+8**	72+10800	170143	82402	327
Mutagenesis	**30+3**	30+14400	35035	15218	880
Hepatitis	**24+3**	24+36000	71008	14774	793

Table 3: *Predictive accuracy* comparison of the Bayes net parameters (cp+) with learned Markov net weights (mbn). cnt/freq = count/frequency regression model.

CLL	UW	Mondial	MovieLens	Mutagenesis	Hepatitis
mbn	-0.44 $\pm$ 0.07	**-1.25** $\pm$ 0.04	-0.79 $\pm$ 0.03	-0.91 $\pm$ 0.09	-1.18 $\pm$ 0.26
log(cp)+cnt	-0.47 $\pm$ 0.10	-1.39 $\pm$ 0.19	-1.19 $\pm$ 0.07	-0.84 $\pm$ 0.03	-1.33 $\pm$ 0.07
log(cp)+freq	**-0.41** $\pm$ 0.04	-1.36 $\pm$ 0.17	**-0.71** $\pm$ 0.01	**-0.73** $\pm$ 0.04	**-1.07** $\pm$ 0.10

Accuracy	UW	Mondial	MovieLens	Mutagenesis	Hepatitis
mbn	80.3% $\pm$ 0.05	47.8% $\pm$ 0.03	59.7% $\pm$ 0.02	61.5% $\pm$ 0.02	51.0% $\pm$ 0.02
log(cp)+cnt	78.3% $\pm$ 0.08	48.4% $\pm$ 0.02	64.3% $\pm$ 0.01	61.4% $\pm$ 0.05	49.2% $\pm$ 0.03
log(cp)+freq	**81.0%** $\pm$ 0.06	**48.5%** $\pm$ 0.02	**65.1%** $\pm$ 0.01	**67.0%** $\pm$ 0.03	**54.8%** $\pm$ 0.02

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

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

Predictive Accuracy. Table 3 compares the prediction scores of the methods. We first discuss the Bayes net parametrization, then compare it to Markov net weight learning.

Bayes Net Parameter Learning. *Frequency regression outperforms count regression on every dataset, on both metrics.* The CLL score improves 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.

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 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 model scores slightly higher than the Markov net weights on every dataset, with the biggest differences on MovieLens (5%) and Hepatitis (4%).

Conclusion. The findings support the main claim of our paper, that balancing predictor scales is important for

using Bayes net parameters as factors in a log-linear model. Compared to optimized general weights, the BN log-linear model performs very well, which supports its usefulness as a baseline relational model.

We also performed experiments using the Markov net weights together with the frequency model. There is little difference between using these weights with counts and frequencies. We hypothesized that the optimized weights already include a scaling component, which explains the equivalence. This hypothesis is confirmed by inspecting the weight magnitudes directly: formulas with many groundings tend to be assigned weights of smaller absolute size. We omit the details due to space constraints. The finding that weights optimized for prediction include a scaling component is further support for the importance of balancing the scales of predictors.

7 MLN-Boost Experiments

MLN-Boost is a state-of-the-art learner for undirected relational models [Khot *et al.*, 2011]. To our knowledge, MLN-Boost has not previously been tested on datasets with many descriptive attributes. MLN-Boost simultaneously learns the model structure and parameters, using functional gradient boosting. The models are based on ensembles of regression trees rather than Bayes nets, so the model structures are not comparable. Nonetheless, MLN-Boost sets a benchmark for predictive accuracy. We used the Boostr implementation by the inventors of MLN-Boost [Khot *et al.*, 2013]. The current version of MLN-Boost is restricted to binary predicates as input. We followed the method of [Khot *et al.*, 2011] and converted attributes with k possible values to k binary predicates (value 1 vs. not 1, value 2 vs. not 2,...). For the predictive accuracy of an attribute with k values, we set the acceptance threshold to $1/k$ and check if the model accepts the true attribute value.[1] We used the inference routine of Boostr. Since inference over the complete dataset is slow, experiments with Boostr challenged our system resources; we tried various settings and report those that yielded the most meaningful results within reasonable computation time ($<$ 3 days for a single setting). Including all attributes in the dataset led to an execution error, so we used, for each dataset, two subsets: (1) a *maximal* multi-class attribute set with no execution error, and (2) a subset comprising all *binary predicates* (e.g., gender). Instead of cross-validation, we use a single random 80-20 training-test split.[2] Boostr was applied to learn a joint model for the selected attributes as targets, with clause-based MLN setting, and default values for options, except for regression trees, where we used as many as possible while obtaining results. Boostr did not terminate on the large dataset MovieLens.

Table 4 shows the results. MLN-Boost is much more scalable than previous MLN learning methods, but still slower than Bayes net learning, by a factor of 10-1,000 depending on the learning task. The number of regression trees has a big impact on learning time. Predictive accuracy for binary attributes (dataset-2) is higher than for multi-class attributes (dataset-2+). The CLL score for BN learning is much better on 2 out of 3 multi-class tasks, and worse on 2 out of 3 binary-class tasks. The scores are comparable only to a limited degree because the BN scores were derived from a joint model of all predicates with cross-validation. Nonetheless, we conclude that our frequency model is very competitive with MLN-Boost, especially for nonbinary attributes. A multi-class extension of MLN-Boost would permit a more direct comparison; we leave this for future work.

Table 4: Performance Metrics for MLN-Boost. NT = inference did not terminate.

Task	#Attributes/ #Predicates	Learning/ #Trees	ACC	CLL	Bayes Net: log(cp)+freq
UW-2+	4/11	1018.25 s/10	27.68%	-1.74	**-0.41**
Mondial-2+	7/31	314 s/10	34.77%	-2.72	**-1.36**
Mondial-2	1	260.2 s/20	80.77%	**-0.24**	-1.36
Muta-2+	3/7	1589 s/5	46.19%	**-0.63**	-0.73
Muta-2	4	94389.2 s/20	68.98%	**-0.53**	-0.73
Hepatitis-2+	13/37	9791s/5	NT	NT	-1.07
Hepatitis-2	5	19921 s/20	30.08%	-1.53	**-1.07**

8 Conclusion and Future Work

This paper presented a new log-linear inference model for Bayes nets applied to relational data. In a Bayes net model, the weight parameters are log-conditional probabilities of parents given children. An innovation of our model is to use the frequencies of relational patterns as predictor variables, rather than their counts as in previous log-linear models. Using frequencies rather than counts addresses the imbalance problem, that counts of different features may be on very different scales. In empirical tests of predictive performance, frequencies outperformed counts on all datasets. A novel sampling semantics for the frequency model shows that it can be interpreted as the expected value of applying Bayes net prediction to a random instantiation of a target node's Markov blanket. Thus while the change from counts to frequencies is a small alteration in the form of the predictive equations, it leads to a different semantics, and it has a big impact on predictive performance. Using the maximum likelihood values as Bayes net parameters is much faster than optimizing weights using standard Markov Logic methods, typically seconds vs. hours.

A topic for future work is to investigate the extension of the local probability models to joint inferences. Local probability models may be inconsistent in the sense that there is no joint distribution that agrees with the local conditional probabilities [Heckerman *et al.*, 2000]. An open theoretical question is whether frequency regression models for different target nodes are guaranteed to be mutually consistent. If they are inconsistent, a possible approach to is to apply the recent averaging methods for dependency networks [Lowd, 2012].

[1] We experimented with different standard methods for extending binary decisions to k classes [Bishop, 2006, Ch.7.1.3], and found that this method gives the best results for MLN-Boost.

[2] Except for UW where cross-validation is feasible for the maximal set, so we did not use a binary subset.

References

[Bishop, 2006] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[Domingos and Lowd, 2009] Pedro Domingos and Daniel Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan and Claypool Publishers, 2009.

[Domingos and Richardson, 2007] Pedro Domingos and Matthew Richardson. Markov logic: A unifying framework for statistical relational learning. In *Introduction to Statistical Relational Learning* [2007].

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

[Getoor and Taskar, 2007] Lise Getoor and Ben Taskar. *Introduction to statistical relational learning*. MIT Press, 2007.

[Heckerman *et al.*, 2000] David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, Carl Kadie, and Pack Kaelbling. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, 2000.

[Khosravi, 2013] Hassan Khosravi. Fast parameter learning for markov logic networks using bayes nets. In *ILP'12 (to appear)*, 2013.

[Khot *et al.*, 2011] Tushar Khot, Sriraam Natarajan, Kristian Kersting, and Jude W. Shavlik. Learning markov logic networks via functional gradient boosting. In *ICDM*, pages 320–329. IEEE Computer Society, 2011.

[Khot *et al.*, 2013] Tushar Khot, Jude Shavlik, and Sriraam Natarajan. Boostr, 2013. URL = `http://pages.cs.wisc.edu/~tushar/Boostr/`.

[Kok *et al.*, 2009] Stanley Kok, M. Summer, Matthew Richardson, Parag Singla, H. Poon, D. Lowd, J. Wang, and Pedro Domingos. The Alchemy system for statistical relational AI. Technical report, University of Washington., 2009. Version 30.

[Lowd and Domingos, 2007] Daniel Lowd and Pedro Domingos. Efficient weight learning for Markov logic networks. In *PKDD*, pages 200–211, 2007.

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

[Natarajan *et al.*, 2010] Sriraam Natarajan, Tushar Khot, Daniel Lowd, Prasad Tadepalli, Kristian Kersting, and Jude W. Shavlik. Exploiting causal independence in markov logic networks: Combining undirected and directed models. In *ECML/PKDD (2)*, pages 434–450, 2010.

[Natarajan *et al.*, 2012] Sriraam Natarajan, Tushar Khot, Kristian Kersting, Bernd Gutmann, and Jude W. Shavlik. Gradient-based boosting for statistical relational learning: The relational dependency network case. *Machine Learning*, 86(1):25–56, 2012.

[Neville and Jensen, 2007] Jennifer Neville and David Jensen. Relational dependency networks. In *Introduction to Statistical Relational Learning* [2007], chapter 8, pages 239–268.

[Poole, 2003] David Poole. First-order probabilistic inference. In *IJCAI*, pages 985–991, 2003.

[Russell and Norvig, 2010] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010.

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

[Schulte *et al.*, 2012] Oliver Schulte, Hassan Khosravi, and Tong Man. Learning directed relational models with recursive dependencies. *Machine Learning*, 89:299–316, 2012.

[Schulte, 2011] Oliver Schulte. A tractable pseudo-likelihood function for Bayes nets applied to relational data. In *SIAM SDM*, pages 462–473, 2011.

[Sutton and McCallum, 2007] C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. In *Introduction to Statistical Relational Learning* [2007], chapter 4, pages 93–128.

[Taskar *et al.*, 2002] Benjamin Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *UAI*, pages 485–492. Morgan Kaufmann Publishers Inc., 2002.