Relational Inference With Bayes Nets

Oliver Schulte OSCHULTE@CS.SFU.CA
Arthur E. Kirkpatrick TED@SFU.CA
School of Computing Science, Simon Fraser University
Vancouver-Burnaby, Canada

Yuke Zhu YUKEZ@STANFORD.EDU
Computer Science Department, Stanford University
Stanford, California, United States

Zhensong Qian ZQIAN@SFU.CA
School of Computing Science, Simon Fraser University
Vancouver-Burnaby, Canada

Tianxiang Gao TGAO@CS.UNC.EDU
Department of Computer Science, University of North Carolina at Chapel Hill
Chapel Hill, North Carolina, United States

Abstract

Probabilistic predictions for relational structures are a major topic in the intersection of Machine Learning and Artificial Intelligence. We describe a new log-linear multi-relational model for making predictions based on Bayes nets. This method makes accurate inferences using only the maximum likelihood estimates of the Bayes net parameters, estimates with computationally simple closed forms. The feature functions in the log-linear model are the frequencies with which relevant features occur in a relational structure. The weights are computed as log-transformations of the Bayes net parameters. Under mild assumptions, the prediction of our log-linear model is equivalent to the expected value of a prediction computed from a random instantiation of the Bayes net. Compared with state-of-the-art Markov net methods (Alchemy weight learning) on five benchmark datasets, Bayes net learning is much faster—parameter learning took seconds vs. hours— while its predictive accuracy was superior for four datasets and competitive for the fifth.

1. Introduction

Bayes nets have the attractive feature that when applied to data that are independent and identically distributed, learning their parameters is both scalable and intuitive. Under the maximum likelihood criterion, the parameters are estimated by the empirical conditional frequencies, which are efficient to compute, simple to interpret, and accurate in their predictions. In striking contrast, current methods for estimating Bayes nets parameters for *relational* data—which typically are dependent and drawn from multiple distributions—require expensive optimization techniques such as local search, due to the use of combining rules (?, ?) or aggregation functions (?) (see Section 1.4 below).

We describe an approach to efficiently learning Bayes nets from relational data that accurately predicts attributes of individuals using maximum likelihood estimates. The model predicts an attribute of a individual by combining weights derived from a Bayes net (BN), whose structure and parameters are learned for multiple, interrelated *populations*,

with predictors (feature functions) efficiently computable for that individual attribute. To achieve this accuracy we must address the **imbalance problem**: in relational data, the range of the feature functions can vary greatly, such that the impact of functions with larger values can easily overwhelm those with smaller values. The model solves this by normalizing the feature function ranges.

Our approach follows the widely adopted semantics for population-level Bayes nets known as knowledge-based model construction (KBMC) (?, ?, ?), which views the Bayes net as a **template model**. The template model summarizes a much larger **instantiated model**, which in principle is obtained by instantiating the first-order variables in the template with every constant from the appropriate domain; see Figure 1. For example, the node $gender(\mathbb{A})$ in the template is instantiated as nodes $gender(a_1), \ldots, gender(a_n)$ for a domain with n persons (?, ?, ?, ?, ?). Since the resulting instantiated model would be huge, the template semantics is used instead as a conceptual aid to define valid probabilistic inferences (?). We explain the differences between our approach and previous applications of KBMC to Bayes nets in Section 1.4.

1.1 Gibbs Conditional Probabilities for relational data

We base our model on a fundamental quantity for inference in generative models, the conditional probability of a random variable given an assignment of values to *all* remaining random variables. We call these probabilities fundamental because they are sufficient to compute a full joint probability distribution over all random variables via Gibbs sampling (?, ?, ?). We therefore refer to them as **Gibbs conditional probabilities**, or simply Gibbs probabilities, distinguishing them from the conditional probability parameters of a Bayes net.[1] In this paper we present a method for computing relational Gibbs probabilities for a Bayes net template model.

For i.i.d. data, a Bayes net is usually viewed as defining a joint probability distribution over assignments of values to its nodes, via the standard product formula (?). The product formula entails that a Gibbs probability can be computed by a log-linear equation (?, Ch.14.5.2), which we refer to as the Bayes net Gibbs probability equation. It is natural to also define Gibbs probabilities in log-linear form for the more general relational case where the BN is viewed as a template.

A discriminative log-linear equation models a conditional probability for a target node value, given an assignment of values to other variables, called the input variables. In the case of a Gibbs probability, the input variables comprise all variables other than the target node. A log-linear model requires defining a set of features, and for each feature a feature function that returns a number for that feature and a given conditional probability to be computed (?). The parameters of the model are feature weights, one for each feature. The conditional probability of the target node value given values for the input variables is proportional to the exponentiated weighted sum of the feature functions. In our proposed model, the set of features is the set of joint value assignments to a child and its parents in the BN structure. For each feature, the feature function is the frequency with which it is instantiated in the given query (i.e., its frequency given the conjunction of input variable

1. In the terminology of dependency networks (?), Gibbs probabilities are referred to as local probability distributions. The WinBUGS system refers to them as full conditional probabilities (?).

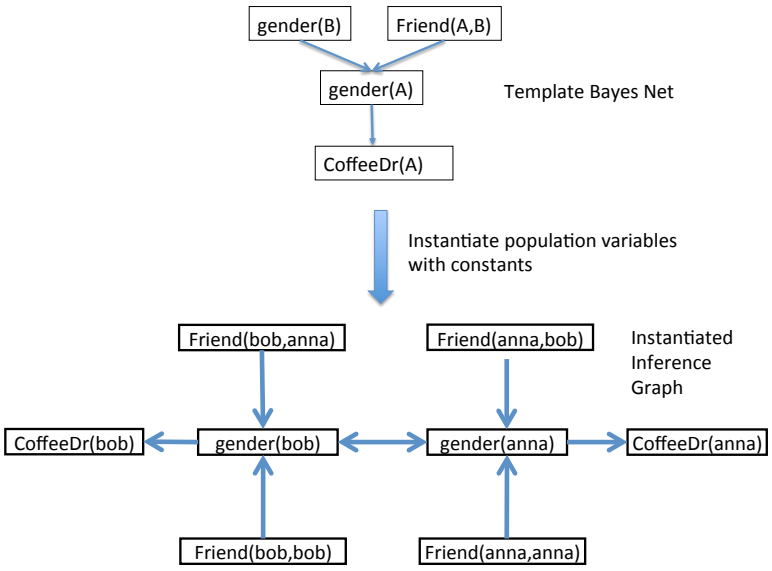

Figure 1: A Bayes net template model (top) and the instantiated graph (bottom), with two individuals Anna and Bob in the domain of the first-order or population variables $\mathbb{A}, \mathbb{B}$. Population variables are to be instantiated with all constants that denote a member of the applicable set. Cyclicity: The two instantiations $\mathbb{A}\backslash anna, \mathbb{B}\backslash bob$ and $\mathbb{A}\backslash bob, \mathbb{B}\backslash anna$ produce a cycle involving the two edges $gender(anna) \rightarrow gender(bob)$ and $gender(bob) \rightarrow gender(anna)$. Imbalance: When predicting the value of $gender(anna)$, there is only one conditional probability factor associated with the instantiated child $CoffeeDr(anna)$, whereas there are many factors associated with the parents $gender(\mathbb{B}), Friend(\mathbb{A}, \mathbb{B})$.

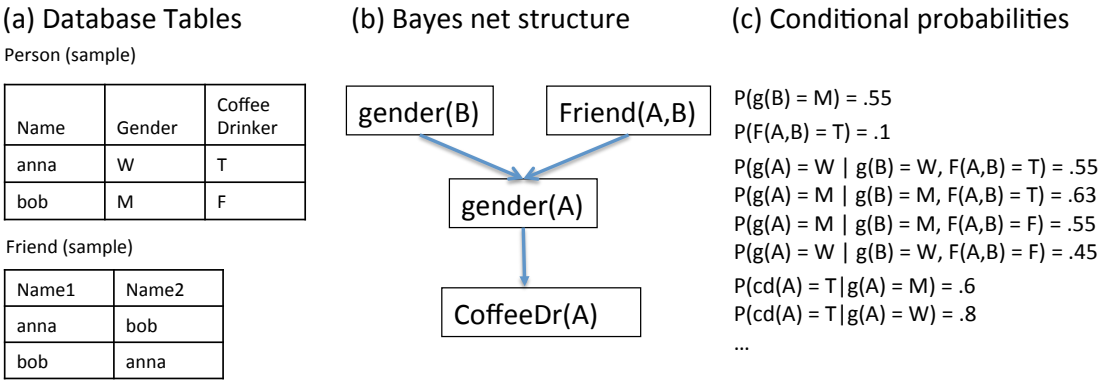

Figure 2: Constructing a template Bayes net. (a) A relational database. By convention, a pair not listed in the Friend table are not friends. The *Friend* relation represented by the Friend table is symmetric, $Friend(\mathbb{A}, \mathbb{B}) = Friend(\mathbb{B}, \mathbb{A})$, and irreflexive, $Friend(\mathbb{A}, \mathbb{A}) = \mathrm{F}$. Only a subset of the rows are shown. (b) A Bayes net structure G learned from the all the rows of the database. (c) The conditional probabilities $\boldsymbol{\theta}_G$ for G. There are slightly more men than women, there is a weak correlation between the gender of a person and that of their friends, and a woman is more likely to be a coffee drinker.

values and the target node value). The frequencies need to be normalized with respect to the set of relevant features only, see Section 3.3. The weight associated with a child-parent feature is computed as the log-difference of two quantities determined by the BN parameters: (i) the conditional probability of the child value, given its parent configuration, and (ii) the marginal probability of the child value. This weight measures the extent to which information about the parents changes the prior probability of the child value. We refer to the resulting log-linear equation as the **log-difference frequency equation**. The log-difference frequency equation reduces to the BN Gibbs probability equation in the case where the instantiation frequencies are either 0 or 1, that is, when the input variables determine a unique value for each node.

To illustrate, suppose we wish to compute the probability that individual $\mathbb{A} = sam$ is male, using the template BN of Figure 2 (b). The input variables specify whether Sam is a coffee drinker, which other persons are his friends, and the gender of other people. There are two parents of the gender node in this template model. One feature in the log-linear model assigns these parents the values $Friend(\mathbb{A}, \mathbb{B}) = \mathrm{T}$ and $gender(\mathbb{B}) = \mathrm{M}$. The feature function returns the percentange of Sam's friends that are male, given the values specified in the input variables. The weight associated with this feature is the log-ratio of two quantities: (i) the conditional probability that $gender(\mathbb{A}) = \mathrm{M}$, given the assignment of values to its parents, which is specified as .63 in the BN. (ii) The BN marginal or unconditional probability that $gender(\mathbb{A}) = \mathrm{M}$, which can be computed via BN inference as .55. So the weight of this feature is $w = ln(0.63/0.55) \approx 0.136$. The positive weight indicates that according to the BN template model, having a male friend raises the probability of being male.

This generalization of the standard log-linear BN Gibbs probability equation has theoretical justification as well. We prove that, under mild assumptions, the log-difference frequency equation can be viewed as a random instantiation extension of the BN equation: The log-difference value is equivalent to the expected value of choosing a random instantiation of the nodes in the template BN and applying the BN equation to this instantiation.

1.2 Motivation

The log-difference frequency equation has several advantages over previous inference methods for relational data.

(1) Interpretability. Log-linear weights learned using general optimization methods can be difficult to interpret, as they may reflect complex interactions between different correlated predictor variables. In contrast, a Bayes net parameter can be interpreted as a conditional probability, and reflects local statistics restricted to a parent-child configuration. Under the maximum likelihood criterion, the conditional probabilities can be interpreted in terms of empirical frequencies observed in a relational structure.

The log-difference transformation allows weights to be interpreted as usual in a log-linear model: a positive weight indicates positive relevance (the parent values raise the probability of a child value), a negative weight as negative relevance (parent values lower the child value probability), and a zero weight as irrelevance.

(2) Scalability. Frequency estimates can be viewed as a type of **lifted learning** (?), the approach of using only the sufficient statistics for a dataset 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.
(3) Balancing Predictor Scales. A common choice of feature function is to use instantiation counts rather than frequencies. Whereas in the case of i.i.d. data, counts and frequencies differ only by a constant population size factor, in relational data there is no such simple one-to-one relationship. For instance Sam and Morgan may each have 10 male friends, but if Sam has 10 friends in total and Morgan has 100 friends in total, the frequency of male friends for Sam is 100%, whereas for Morgan it is only 10%. In relational data, normalization involves *local* scaling factors (10 for Sam, 100 for Morgan), whereas in i.i.d. data it involves a single glocal factor (the sample size). It is intuitively clear that the count of male friends should be treated very differently in inferences for Sam than in inferences for Morgan. The general problem with using feature counts is that in a log-linear model with counts, features with more instantiations carry exponentially more weight. Count models tacitly conflate number of instantiations with degree of information. In contrast frequency feature functions are on the common scale [0,1]. We refer to the diverging scale of feature counts as the **imbalance problem**.

1.3 Evaluation.

Using five standard databases, we evaluate the predictive accuracy obtained from log-difference frequency regression. We compare predictive performance with different choices of feature functions (counts vs. frequencies), and different choices of weights, including optimizing weights using general log-linear learning methods. We find that our log-linear model performs competitively with alternative methods and is far faster than general weight learning. Our experiments provide evidence that because of the imbalance problem, maximum likelihoods do not lead to accurate predictions when they are used with feature counts.

1.4 Comparison to Knowledge-Based Model Construction

We contrast our approach at a high level with the traditional knowledge-based model construction approach. Section 8 below provides a detailed discussion of previous work. In the KBMC approach, a template BN is instantiated to produce a *single instantiated BN*, called the inference graph by Neville and Jensen (?); see Figure 1. The standard Bayes net product formula can be applied in the instantiated model to assign joint probabilities for all instantiated nodes. In the relational case, this provides a method for computing, from a template BN, a joint probability for each collection of facts about individuals and their relationships (?, ?). An advantage of our approach is that it avoids several challenges for KBMC.
(1) Multiple Parents and Scalability. In the instantiated model, a parent configuration is usually instantiated multiple times for a given child node. This "multiple-parent problem" (?) requires merging the information from different parents into a single conditional probability for a child node given a specification of values for the multiple parents. Two common approaches are using (i) combining rules (?) and (ii) aggregation functions (?). Adding combining rules or aggregation functions leads to very rich and expressive models. The resulting likelihood function does not factor as a product, unlike the BN likelihood function in the nonrelational case, and therefore optimizing it requires local search methods

rather than using empirical frequency estimates (?, ?). Also, it may be necessary to learn which combining rules/aggregate functions are best for a given dataset, which increases the learning complexity further.
(2) The Cyclicity Problem. Cyclicity arises because the instantiated Bayes net may contain cycles even if the template Bayes net does not. This occurs in the common case that a dataset features auto-correlations, where the value of an attribute for an individual depends on the values of the same attribute for related individuals (?, ?). Figure 1 provides an example. In the presence of cycles, the instantiated model is not a valid Bayes net structure (which must be acyclic). This has been a knotty problem for defining a joint distribution using a template semantics (?, ?, ?). Since a Gibbs conditional equation defines a local distribution that pertains to a single instantiated node, not a full instantiated graph, it is well-defined even when the full instantiated graph contains cyclic dependencies.
(3) The imbalance problem. A joint probability distribution entails conditional probabilities. We can therefore compare the Gibbs probabilities that are entailed by an instantiated model with the Gibbs probabilities defined by our proposed log-linear model. As noted above, the product formula for the joint probability of a Bayes net model entails that that the Gibbs conditional probability has a log-linear form corresponding to a product of conditional probabilities. In the instantiated model, there is one such conditional probability for each instantiation of a parent-child configuration. This means that *in the instantiated model, the feature function is the instantiation count.* For example, consider the template structure $\mathit{intelligence}(S) \rightarrow \mathit{difficulty}(C) \leftarrow \mathit{Registered}(S, C)$. This represents a correlation between the difficulty of a course and the intelligence of students who take the course. If our target node value is $\mathit{intelligence}(\mathit{sam}) = \mathit{hi}$, the prediction will be based on a product of conditional probabilities of the form $\prod_c P(\mathit{difficulty}(c)|\mathit{intelligence}(\mathit{sam}) = \mathit{hi})$, one for each course c that Sam is registered in. Notice that the product of conditional probabilities arises regardless of what combining rule or aggregation function is specified for the parents of an instantiated node. Therefore using the standard BN product formula with a single instantiated inference model entails a "ground-and-count" log-linear model, and suffers from the imbalance problem. The imbalance problem also arises for other graphical model classes that use the KBMC approach with a factored joint probability function, such as undirected Markov net models (?, ?).

Contributions and Significance. Our main contributions are:

1. An approach to relational Bayes net inference that applies a template semantics to Gibbs conditional probabilities on a local region of an instantiated Bayes net, rather than the entire net.

2. A new log-linear equation that computes, given a template Bayes net, a Gibbs conditional distribution for making inferences about relational data. The new log-linear model performs well with easily computed maximum likelihood estimates for the Bayes net parameters (observed conditional frequencies).

Standardizing the scale of variables is a common preprocessing steps in building linear models (?). It may therefore seem unsurprising that normalizing counts improves statistical-relational predictions. Nonetheless, we believe that the innovation of replacing counts by

frequencies as the feature function has important consequences, both conceptual and practical.

(1) As we discussed, the most common KBMC approach of defining inferences with respect to a single instantiated inference model implicitly entails using counts as feature functions. In contrast, using frequencies as feature functions is consistent with a random selection interpretation of the Bayes net model, where population variables represent a random draw from the associated population. The random selection interpretation was developed in detail by Halpern (?) and Bacchus (?) in their classic work on first-order probability logic. Adapting the random selection interpretation for first-order Bayes net models is a recent development in statistical-relational learning (?). Thus the difference between counts and frequencies mirrors an alternative in the semantics of a first-order Bayes net.

(2) The instantiation frequencies of Bayes net node assignments, which we propose to use for computing Bayes net parameters, can be interpreted in terms of *class-level probabilities* (?) (also called type 1 probabilities (?)). For instance, the frequency of the condition $gender(\mathbb{A}) = \text{W}$ represents the proportion of women in the class of persons. The inference models that we develop in this paper, define *instance-level probabilities* for the attributes of specific individuals (also called type 2 probabilities (?)). For instance, our log-linear equation computes a value for the probability that $gender(sam) = \text{W}$, conditional on known facts about *sam*, where *sam* denotes a particular individual. The log-linear equation we propose in this paper therefore computes instance-level probabilities—for the target node values—from given class-level probabilities—the Bayes net parameters. The question of how to compute instance-level probabilities from class-level statistics has received considerable attention from leading AI researchers (?, ?). Our log-linear model represents a new approach to this long-standing question.

(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, and provides a strong scalable baseline learning method.

Paper Organization. We begin (Section 2) with background and define the notation for relational Bayes net models. Section 3 presents the new log-linear relational regression model, which we compare with alternative formulations in Section 4 and then evaluate on five benchmark databases in Sections 5 and 6. Section 7 shows the theoretical equivalence between the frequency regression value and a random instantiation semantics. We end with related work in Section 8, and conclude with suggestions for future work in Section 9.

2. Background: Bayes Nets for Relational Data

We adopt function-based notation from logic for combining Bayes nets with relational concepts (?, ?, ?, ?). Different communities in statistics and logic use different terms for similar concepts, and similar terms for different concepts. We strive for notation that is as broadly accessible as possible. Table 1 summarizes our notation for relational concepts.

Table 1: Summary of Notation for Relational Concepts

Notation	Explanation	Example
$\mathbb{A}, \mathbb{B}$	**Population Variables**	$\mathbb{A}$
T, U, V	**Terms**. A term consists of a **functor** and its arguments. The arguments may be any combination of **population variables** and **constants**.	$gender(\mathbb{A})$
$\vec{T}, \vec{U}, \vec{V}$	**Term Tuples**. A list of terms. A given term can occur at most once. Order is significant.	$gender(\mathbb{B}); Friend(\mathbb{A}, \mathbb{B})$
$\mathrm{Ra}(T), \mathrm{Ra}(\vec{T})$	The **range of a term** and the **range of a term tuple**. The range of a **term** is the range of its functor. The range of a tuple is the Cartesian product of the ranges of the constituent **terms**.	$\mathrm{Ra}(gender(\mathbb{B}), Friend(\mathbb{A}, \mathbb{B})) = \{\mathrm{W}, \mathrm{M}\} \times \{\mathrm{T}, \mathrm{F}\}$
t, u, v	**Values** from the ranges of T, U, V.	$\mathrm{M}; \mathrm{T}$
$\vec{t}, \vec{u}, \vec{v}$	**Tuples of values** from the ranges of $\vec{T}, \vec{U}, \vec{V}$.	(M, T)
$T = t$	**Literal**. A term bound to a value from its range.	$Friend(\mathbb{A}, \mathbb{B}) = \mathrm{T}$
$\vec{T} = \vec{t}$	**Literal conjunction**. A conjunction bound to a tuple of values from its **range**. Every **term** in a literal conjunction is bound to its corresponding value.	$Friend(\mathbb{A}, \mathbb{B}) = \mathrm{T}, gender(\mathbb{A}) = \mathrm{W}$
γ	**Instantiation**. An instantiation maps zero or more **population variables** to appropriate **constants**.	$\gamma = \{\mathbb{A} \backslash anna, \mathbb{B} \backslash bob\}$
$T^*, \vec{T}^*$	**Fully ground term/tuple**. All arguments to all functors are constants. Equivalently, the term/tuple does not contain any population variables.	$gender(anna), Friend(anna, bob)$
Λ^*	Complete **literal conjunction** comprising **fully grounded** terms. The conjunction binds *every* ground term to a value.	See Table 2.
$\mathrm{n}\left[\vec{T} = \vec{t}; \Lambda^*\right]$	**Instantiation Count.** Counts the number of groundings of $\vec{T} = \vec{t}$ that evaluate as true in Λ^*. A **grounding** instantiates *all* population variables in $\vec{T}$.	See Table 2.

2.1 Relational Concepts.

A **population** $\mathcal{P}$ is a set of individuals. Individuals are denoted by lower case identifiers (e.g., *sam*). Identifiers representing individuals are called **constants**.

2.1.1 Functor Notation

A **functor** $f : \mathcal{P}_1, \ldots, \mathcal{P}_a \rightarrow V_f$ maps a list of individuals to a functor value, where V_f is the output type or **range** of the functor. In this paper we consider only functors with a finite range. If $V_f = \{\mathrm{T}, \mathrm{F}\}$, the functor f is a (Boolean) **predicate**; other functors are called **attributes**. A predicate with more than one argument is called a **relationship**. Predicates typically represent the presence of a binary property or a relationship, while attributes typically represent properties that can take multiple values. We use lowercase for attribute functors and uppercase for predicates. We define two kinds of variables, with distinct domains: (i) A **population variable** varies over a population domain. The same population may be associated with more than one population variable. We use an outline Latin font for population variable ($\mathbb{A}, \mathbb{B}, \ldots$). (ii) A **term** is an expression of the form $f(\boldsymbol{\tau}) \equiv f(\tau_1, \ldots, \tau_k)$, where each τ_i is a population variable or a constant of the appropriate argument type for that functor.[2] A term can be assigned values in the range of its functor.

In a context where the functor and arguments of terms are not important, we denote them by uppercase Latin letters ($T, U, \ldots$).

A **literal** is an assignment to a term, denoted generically as $T = t$, where t is in the range of the functor for T. Literals can be combined to generate **formulas**. In this paper we consider only formulas that are **conjunctions** of literals, denoted by the Prolog-style comma notation, e.g., $T = t, \ldots, U = u$, for which we also use the vector notation $\vec{T} = \vec{t}$.

2.1.2 Groundings and Relational Structures

An term is **ground** if it contains no population variables. We mark fully ground terms and formulas by an asterisk, T^*. An **instantiation** for a term assigns a constant to a set of the population variables in the term. Formally, an instantiation γ is a set $\{\mathbb{A}_i \backslash a_1, \ldots, \mathbb{A}_j \backslash a_j\}$ that assigns a constant $\gamma(\mathbb{A}_i)$ to each variable $\mathbb{A}_i$ from its population. The expression T_γ denotes the term that results from applying the substituition γ to all variables in T. If the instantiation γ specifies a constant for all the population variables that occur in the term, the resulting term is ground, and γ is called a **grounding**. An instantiation is applied to a formula by applying it to all terms in the formula. Thus a grounding of a formula is an instantiation that grounds of all its terms.

A **relational structure** is a (model, interpretation) pair that assigns a unique value to each ground term (?). Assuming a finite list of functors, and that all populations are finite, a relational structure is equivalent to a **complete conjunction** of ground literals $\Lambda^* \equiv (\vec{T}^* = \vec{t})$. The ground literals in the relational structure are called **facts**. A conjunction of ground literals evaluates as true in a relational structure if each of its conjuncts is a fact. A well-studied operation (?) for statistical learning is to count, for a given nonground formula $\vec{T} = \vec{t}$ and complete conjunction Λ^*, the number of groundings of the formula that

2. The traditional term in first-order logic for a term variable is "function term". In statistical-relational learning, alternative terms include "parametrized random variable" (?), "atom" (?), "Bayesian atom" (?), and "functor random variable" (?).

evaluate as true in the complete conjunction. Note that only groundings that instantiate exactly the population variables that appear in $\vec{T}$. We refer to this quantity as the formula's **instantiation count**, denoted by

$$\mathrm{n}\left[\vec{T} = \vec{t}; \Lambda^*\right].$$

2.1.3 Examples.

Figure 2(a) shows the facts in a relational structure, represented as database tables. For this example only, let us assume that the tables represent a relational structure completely, and hence specifies a complete conjunction Λ^*. Table 2 shows examples of instantiation counts for this complete conjunction.

Table 2: Instantiation counts in the database of Figure 2. For the sake of the example, we treat the database as specifying a complete conjunction $\Lambda^* = \{gender(anna) = W,\ gender(bob) = M,\ Friend(anna, bob) = T,\ Friend(bob, anna) = T,\ Friend(anna, anna) = F,\ Friend(bob, bob) = F\}$.

Formula	Instantiation Count
$gender(\mathbb{A}) = \mathrm{M}$	1
$Friend(\mathbb{A}, \mathbb{B}) = \mathrm{T}$	2
$Friend(bob, \mathbb{B}) = \mathrm{T}$	1
$gender(\mathbb{B}) = \mathrm{M}, Friend(\mathbb{A}, \mathbb{B}) = \mathrm{T}$	1
$gender(anna) = \mathrm{M}, Friend(\mathbb{A}, anna) = \mathrm{T}$	0

2.2 Bayes Nets for Relational Data.

A Bayes net (BN) is a pair $\langle G, \boldsymbol{\theta}_G \rangle$, where G is a directed acyclic graph and $\boldsymbol{\theta}_G$ is a set of parameters that specify the probability distributions of children conditional on instantiations of their parents. A **Template Bayes Net** (TBN) is a Bayes net whose nodes are nonground terms (?, ?). That is, every node in the Bayes net is a term containing one or more population variables. When describing Bayes nets, we use "term" and "node" interchangeably. Table 3 summarizes our notation for Bayes nets and Figure 2(b) shows an example net.

A **family** comprises a node and its parents. A **family configuration** specifies a value for a child node and each of its parents. Since we consider functors with discrete ranges only, there are only finitely many family configurations. Using the notation in Table 3, a family configuration is equivalent to the conjunction $T = t, \mathrm{Pa}(T) = \vec{t}_{pa}$. For each family configuration, a **Bayes net parameter**

$$\theta(T = t | \mathrm{Pa}(T) = \vec{t}_{pa})$$

Table 3: Summary of Notation for Template Bayes Nets

Notation	Explanation	Example
$\mathrm{Pa}(T) : T \rightarrow \vec{T}$	**Parents**. The **terms** associated with the parent nodes of the unique Bayes net node associated with T. The parent terms will be distinct.	$\mathrm{Pa}(\mathit{gender}(\mathbb{A})) = \langle \mathit{gender}(\mathbb{B}), \mathit{Friend}(\mathbb{A}, \mathbb{B}) \rangle$
$\mathrm{Ch}(T) : T \rightarrow \vec{T}$	**Children**. The **terms** associated with the child nodes of the unique Bayes net node associated with T.	$\mathrm{Ch}(\mathit{gender}(\mathbb{A})) = \mathit{CoffeeDr}(\mathbb{A})$
$\vec{t}_{pa}$	**Value in the parent range**. Tuple of values from $\mathrm{Ra}(\mathrm{Pa}(T))$.	$\langle \mathrm{M}, \mathrm{T} \rangle$
$T = t, \mathrm{Pa}(T) = \vec{t}_{pa}$	**Family Configuration** that specifies values for node T and its parent nodes $\mathrm{Pa}(T)$.	$\mathit{gender}(\mathbb{A}) = \mathrm{M}$, $\mathit{gender}(\mathbb{B}) = \mathrm{W}$, $\mathit{Friend}(\mathbb{A}, \mathbb{B}) = \mathrm{T}$
$\theta(T = t \mid \mathrm{Pa}(T) = \vec{t}_{pa})$	**Conditional probability** of a node value given a parent configuration.	$\theta(\mathit{gender}(\mathbb{A}) = \mathrm{W} \mid \mathit{gender}(\mathbb{B}) = \mathrm{W}, \mathit{Friend}(\mathbb{A}, \mathbb{B}) = \mathrm{T}) = 0.55$
$\theta(T = t)$	**Marginal probability** of a node value entailed by the Bayes net.	$P(\mathit{gender}(\mathbb{A}) = \mathrm{M}) = 0.55$

specifies the probability of the child value t given the parent values $\vec{t}_{pa}$.[3] Given a complete conjunction Λ^*, the number of family configurations is[4]

$$\mathrm{n}\left[T = t, \mathrm{Pa}(T) = \vec{t}_{pa}; \Lambda^*\right].$$

The parameter values for a Bayes net define a joint distribution over its nodes via the standard product formula. A parametrization therefore entails a marginal, or unconditional, probability for a single node. We denote the marginal probability that node T has value t by the notation

$$\theta(T = t).$$

2.3 Bayes Net Gibbs Probabilities.

We review the equation for Gibbs conditional probabilities for a Bayes net that specifies a joint distribution over its nodes via the standard product formula (?). Inferring a Gibbs conditional probability can be represented as a probabilistic query

$$P(T = t | \vec{V} = \vec{v}) = ?$$

where T is the target node, t is a value for the target node, and $\vec{v} = \vec{V}$ specifies a value for every other node. The product formula for the Bayes net joint distribution entails that the Gibbs probability is proportional to a product of conditional probabilities for the target node and its children (?, Ch.14.5.2):

$$P(T = t | \vec{V} = \vec{v}) \propto \prod_{U \in \{T\} \cup \mathrm{Ch}(T)} \theta(U = u | \mathrm{Pa}(U) = \vec{u}_{pa}) \quad (1)$$

3. For i.i.d. data, a commonly used notation for a BN parameter is θ_{ijk} (?).
4. For i.i.d. data, a commonly used notation for this quantity is n_{ijk} (?).

where u and $\vec{u}_{pa}$ are specified by the values in $t, \vec{v}$ for the corresponding nodes. The product is a number between 0 and 1 that needs to be normalized to obtain the Gibbs conditional probability.

Example. For the Bayes net of Figure 2, consider the query

$$P(gender(\mathbb{A}) = \mathrm{W} | gender(\mathbb{B}) = \mathrm{W}, Friend(\mathbb{A}, \mathbb{B}) = \mathrm{T}, CoffeeDr(\mathbb{A}) = \mathrm{T}).$$

By Equation 1, this probability is proportional to

$$\begin{aligned} \theta(gender(\mathbb{A}) = \mathrm{W} | gender(\mathbb{B}) = \mathrm{W}, Friend(\mathbb{A}, \mathbb{B}) = \mathrm{T}) \quad &\cdot \quad \theta(CoffeeDr(\mathbb{A}) = \mathrm{T} | gender(\mathbb{A}) = \mathrm{W}) \\ = 0.55 \quad &\cdot \quad 0.8 = 0.44. \end{aligned}$$

2.4 Log-Linear Models

Equation 1 can be seen as an instance of a standard log-linear schema as follows. The general equation form that defines a **discriminative log-linear model** (?, Sec.4.2.2.1) is

$$P(T = t | \vec{V} = \vec{v}) \propto exp(w_t + \sum_{i=1}^{K} w_i f_i(t, \vec{V})), \tag{2}$$

where T is the target or output variable, and $\vec{v} = \vec{V}$ represents an assignment of values to the input variables. The model is based on a finite set of K features, and for each feature there is a real-valued weight parameter w_i. The term w_t is a bias weight that may depend on the target node value t but not the input variables. The functions $f_1, \ldots, f_k$ are **feature functions**, such that f_i returns a real number for feature i given values for both the target and input variables. A log-linear equation defines a log-linear model; we use the term "equation" to emphasize the mathematical form, and the term "model" to emphasize the parameter space.

Rewriting Equation 1 as

$$P(T = t | \vec{V} = \vec{v}) \propto exp(\sum_{U \in \{T\} \cup \mathrm{Ch}(T)} \ln \theta(U = u | \mathrm{Pa}(U) = \vec{u}_{pa})) \tag{3}$$

shows that a BN Gibbs probability follows a log-linear model with the following specifications. (1) The features are all family configurations whose child node is either the target node or a child of the target node. (2) The feature weights are the log-conditional probabilities associated with a family configuration; the bias weight is 0. (3) The feature function for each a family configuration returns 1 if the family configuration is specified by the conjunction $(t, \vec{v})$ of input and output variables, 0 otherwise.

This log-linear equation specifies the Gibbs probability for the nonground terms/nodes in the template Bayes net. Our goal in this paper is to define inference for queries whose target are ground terms. We therefore want to generalize Equation (3) for ground terms. The log-linear equations for ground terms that we consider are also based on products of the Bayes net parameters, combined with transformations.

Table 4: Summary of Notation for Relational Gibbs Probabilities

Notation	Explanation	Example
T^*_γ	Fully Ground Target Node	$[gender(\mathbb{A})]_{\{\mathbb{A}\setminus sam\}} = gender(sam)$
t	Target Node Value	M
Δ^*	**Query Conjunction.** A fully ground literal conjunction that binds every term to a value except for the target node	See Table 2
$P(\mathsf{T}^*_\gamma = t \mid \Delta^*)$	**Gibbs conditional probability**, or a **Query**	$P(gender(sam) = \text{M} \mid \Delta^*)$
$\mathsf{U}_\gamma = u, \text{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}$	Partially ground query family configuration	$gender(sam) = \text{M}$, $gender(\mathbb{B}) = \text{W}$, $Friend(sam, \mathbb{B}) = \text{T}$
$\text{n}^\text{r}\left[\mathsf{U}_\gamma = u, \text{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right]$	**Relevant Instantiation Count** of a query family configuration.	See Section 3.3.
$\text{p}^\text{r}\left[\mathsf{U}_\gamma = u, \text{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right]$	**Relevant Instantiation Frequency** of a query family configuration.	See Section 3.3.

3. The Log-Difference Frequency Equation

We propose a log-linear equation for computing a Gibbs conditional probability in closed form, given (i) a ground target node (the output variable), (ii) a target value for the target node, (iii) a complete set of values for all ground terms other than the term of the target node (the input variables), and (iv) a template Bayes net. We refer to our proposal as the **log-difference frequency equation**. Our notation is summarized in Table 4. For now we assume that component (iv), the template BN, is fixed (Section 3.3 discusses structure learning), and consider in this section components (i)–(iii). Figure 3 shows the program flow for computing a Gibbs probability using the log-difference frequency equation.

3.1 Conditional Queries

Conditional queries comprise the following elements:

1. A **target literal** $\mathsf{T}^*_\gamma = t$, where T^*_γ denotes a ground term that results from applying the grounding γ to the term T.

2. A conjunction Λ^* that specifies a value for each ground term other than T^*_γ. The conjunction $(\mathsf{T}^*_\gamma = t, \Delta^*)$ specifies the value of *every* ground term; we refer to it as the **query conjunction**.

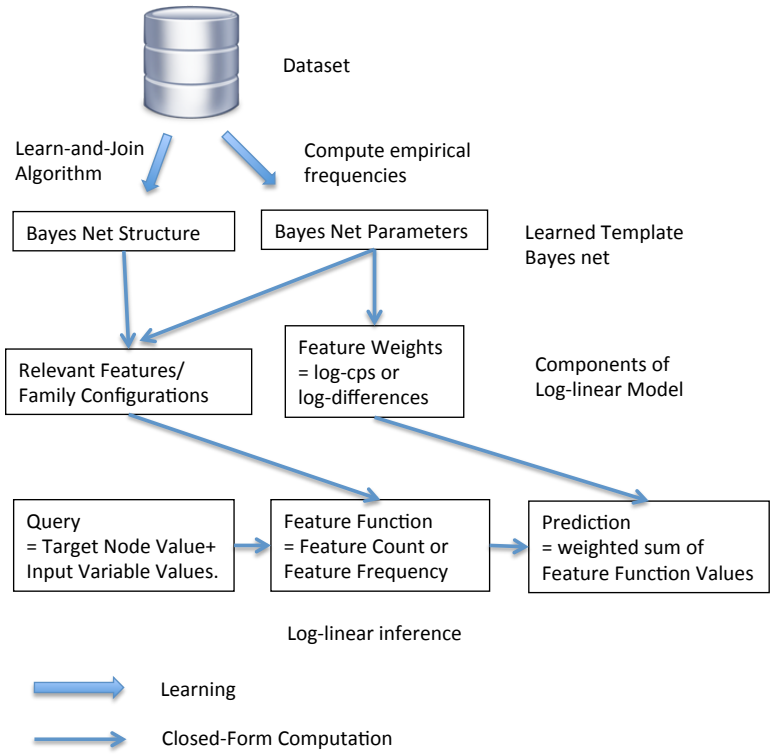

Figure 3: The program flow for computing relational Gibbs probabilities with a template Bayes net. Features and weights are computed from the Bayes net. Feature function values are computed for each query.

A Gibbs conditional probability corresponds to a probabilistic **query**

$$P(\mathsf{T}^*_\gamma = t|\Lambda^*) =?$$

For example, a target node may be $gender(sam)$, which results from the grounding $gender(\mathbb{A})\{\mathbb{A}\backslash sam\}$.[5] The conjunction Λ^* specifies a value for all ground terms other than $gender(sam)$.

3.2 Features and Feature Weights

The features are all family configurations whose child node is either the target node or a child of the target node, as with nontemplate BNs. Thus the set of features equals the set of **query family configurations**

$$QFC \equiv \{\mathsf{U} = u, \mathrm{Pa}(\mathsf{U}) = \vec{u}_{pa} : U \in \{T\} \cup \mathrm{Ch}(T), u \in \mathrm{Ra}(U), \vec{u}_{pa} \in \mathrm{Ra}(\mathrm{Pa}(U))\}.$$

The weights are computed as follows.

1. The offset weight

$$w_0 \equiv \ln \theta(T = t)$$

5. The node $gender(sam)$ can also be defined by the grounding $gender(\mathbb{B})\{\mathbb{B}\backslash sam\}$. To make our definition unambiguous, we assume that the BN is in main functor format and that the template node T is the main functor node for its functor; see (?).

Table 5: Relevant and Irrelevant Features, or Family Configurations, for the Bayes net of Figure 2. Marginal probabilities are computed using standard Bayes net inference.

Child Node Value	Marginal Probability	Parent configuration	Conditional Probability	Relevant?
$\mathit{CoffeeDr}(\mathbb{A}) = \mathrm{T}$	0.70	$\mathit{gender}(\mathbb{A}) = \mathrm{W}$	0.80	yes
$\mathit{gender}(\mathbb{A}) = \mathrm{W}$	0.45	$\mathit{gender}(\mathbb{B}) = \mathrm{W}$, $\mathit{Friend}(\mathbb{A}, \mathbb{B}) = \mathrm{F}$	0.45	no

is the log-marginal probability of the target node value, as entailed by the template Bayes net. (Apply the standard product formula for joint probabilities and sum over all joint probabilities where $T = t$).

2. For each query family configuration there is an associated weight

$$w \equiv [\ln \theta(U = u|\mathrm{Pa}(U) = \vec{u}_{pa}) - \ln \theta(U = u)] .$$

Since the log-linear weights are a deterministic transformation of the BN parameters, our approach in effect changes the parameter space from log-linear weights to BN parameters. The weight measures the relevance of the parent configuration to predicting the child value. If $\theta(U = u|\mathrm{Pa}(U) = \vec{u}_{pa}) = \ln \theta(U = u)$, then the parent configuration is probabilistically independent of the child condition (according to the template BN). In that case we say that the feature defined by the family configuration is **irrelevant**, otherwise **relevant**. In the log-difference equation, irrelevant features receive weight 0, which is equivalent to eliminating them from the model. Table 5 illustrates relevant and irrelevant features.
Discussion. It is well-known that eliminating irrelevant features is important for predictive accuracy in statistical-relational learning (?, ?, ?, ?). For example, individuals whose every relationship with the target individual has value F are often irrelevant to predicting features of the target. In the example above, the gender of nonfriends (all $\mathbb{B}$ such that $\mathit{Friend}(\mathit{sam}, \mathbb{B}) = \mathrm{F}$) is probabilistically independent of the gender of the target. In a realistic social network, where 99% or more of the users are *not* friends with a given individual, this would entail that the vast majority of groundings are irrelevant to predicting an individual's gender. A common approach to eliminating irrelevant predictors is to stipulate a logical condition that must be met for the predictor to be included (?, ?, ?, ?). The log-difference model instead defines irrelevant features in terms of the Bayes net parameters, and eliminates them by assigning 0 weight.

3.3 Feature Functions

For each feature, a feature function maps the query conjunction to a real number. A common feature function choice in log-linear models is the number of times that the feature is instantiated in the query conjunction. Our basic proposal is to use, instead, the *frequency* with which each feature is instantiated in the query conjunction. To compute feature frequencies, we first compute feature counts, then normalize. To count all and only instantiations that are related to the grounding of a target node, we apply the grounding to its parents, children, and co-parents, as illustrated in Figure 4.

Normalizing feature counts to obtain feature frequencies must be done with care to take account of irrelevant features. Since irrelevant features receive log-difference weight 0, they

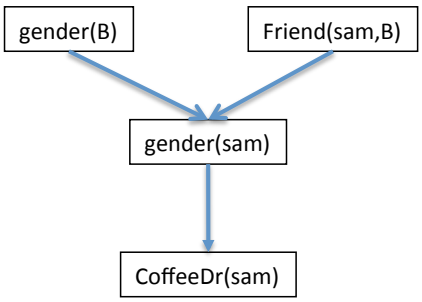

Figure 4: Graphical representation of partial grounding for the target node *gender*(*sam*). The graph results from instantiating the population variable $\mathbb{A}$ with the constant *sam* in the template graph of Figure 2. Feature counts are computed with respect to the instantiated nodes. Notice that, for some nodes, this instantiation leads to only a *partial* grounding.

are effectively pruned from the model, so their portion of the instantiation counts should be shifted to the relevant features, analogous to conditioning on relevant features. We therefore define the following two feature functions, where a feature is a family configuration.

Relevant Count The relevant count n^{r} is 0 if the family configuration is irrelevant; otherwise it is the instantiation count.

Relevant Frequency The relevant frequency of a feature is its relevant count, divided by the sum of all relevant counts for the same family.

Algorithm 1 describes the computation of relevant counts and relevant frequencies. Table 6 gives examples. This completes our definition of the set of features, weights, and feature functions. All told, the resulting log-linear equation is as follows.

[The Log-Difference Frequency Equation]

$$P(\mathsf{T}^*_\gamma = t|\Delta^*) \quad \propto \quad \exp(\ln\theta(T=t) + \sum_U \sum_{u,\vec{u}_{pa}} [\ln\theta(U=u|\mathrm{Pa}(U)=\vec{u}_{pa}) - \ln\theta(U=u)] \quad \cdot \quad \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right])$$

where

$$\begin{array}{rlll} U & \text{varies over} & T \cup \mathrm{Ch}(T), & \\ u & \text{varies over} & \mathrm{Ra}(U), & \text{and} \\ \vec{u}_{pa} & \text{varies over} & \mathrm{Ra}(\mathrm{Pa}(U)). & \end{array}$$

3.4 Discussion: Advantages of weights as log-differences of Bayes parameters

Using Bayes nets in log-difference form is more scalable and interpretable than previous inference methods:

Algorithm 1: Computing Feature Functions: Relevant Counts and Relevant Frequencies. Feature = Query Family Configuration.

Input: Template Bayes Net B; Query $P(\mathsf{T}^*_\gamma = t|\Delta^*) = ?$; Feature $F = \mathsf{U} = u, \mathrm{Pa}(\mathsf{U}) = \vec{u}_{pa}$

Output: Relevant Feature Count $\mathrm{n^r}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right]$

1: **if** the feature F is not relevant in B **then**
2: **return** 0.
3: **else**
4: Partially Ground Conjunction $C := (\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa})$.
5: Complete Conjunction $\Lambda^* := (\Delta^*, \mathsf{T}^*_\gamma = t)$.
6: **return** $\mathrm{n^r}\left[C; \Lambda^*\right] := \mathrm{n}\left[C; \Lambda^*\right]$.
7: **end if**

Output: Relevant Feature Frequency

8: *Total_Relevant_Count* $:= \sum_{u', \vec{u}'_{pa}} \mathrm{n^r}\left[\mathsf{U}_\gamma = u', \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}'_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right]$
9: **return** $\mathrm{p^r}\left[C; \Lambda^*\right] := \mathrm{n^r}\left[C; \Lambda^*\right] / \mathit{Total\_Relevant\_Count}$.

Child Value	Prior Prob.	Parent State	Cond. Prob.	w	$\mathrm{p^r}$	$w \times \mathrm{p^r}$	$\mathrm{n^r}$	$w \times \mathrm{n^r}$
$cd(sam) = \mathrm{T}$	0.70	$g(sam) = \mathrm{W}$	0.80	0.13	1.0	0.13	1	0.13
$cd(sam) = \mathrm{F}$	0.30	$g(sam) = \mathrm{W}$	0.20	−0.40	0.0	0.00	0	0.00
$g(sam) = \mathrm{W}$	0.45	$g(B) = \mathrm{W}$, $F(sam, B) = \mathrm{T}$	0.55	0.20	0.4	0.08	40	8.02
$g(sam) = \mathrm{W}$	0.45	$g(B) = \mathrm{M}$, $F(sam, B) = \mathrm{T}$	0.37	−0.19	0.6	−0.11	60	−11.74
$g(sam) = \mathrm{W}$	0.45	n/a	n/a	−0.79	1.0	−0.79	1	−0.79
Sum $(\ln P(gender(sam) = \mathrm{W}\|\Delta^*))$						−0.70		−4.38
$cd(sam) = \mathrm{T}$	0.70	$g(sam) = \mathrm{M}$	0.60	−0.15	1.0	−0.15	1	−0.15
$cd(sam) = \mathrm{F}$	0.30	$g(sam) = \mathrm{M}$	0.40	0.28	0.0	0.00	0	0.00
$g(sam) = \mathrm{M}$	0.55	$g(B) = \mathrm{W}$, $F(sam, B) = \mathrm{T}$	0.45	−0.20	0.4	−0.08	40	−8.02
$g(sam) = \mathrm{M}$	0.55	$g(B) = \mathrm{M}$, $F(sam, B) = \mathrm{T}$	0.63	0.13	0.6	0.08	60	8.14
$g(sam) = \mathrm{M}$	0.55	n/a	n/a	−0.59	1.0	−0.59	1	−0.59
Sum $(\ln P(gender(sam) = \mathrm{M}\|\Delta^*))$						−0.75		−0.63

Table 6: Applying the log-difference frequency equation with the BN of Figure 2 to compute $P(gender(sam) = \mathrm{W}|\Delta^*)$ and $P(gender(sam) = \mathrm{M}|\Delta^*)$. Each row represents a feature/family configuration. For the sake of the example we suppose that the conjunction Δ^* specifies that Sam is a coffee drinker, has 60 male friends, and 40 female friends. The last two columns show the result of replacing frequencies by counts (the log-difference count equation in Section 4).

(1) Interpretable features/structure. General log-linear weights learned from data can be difficult to interpret, as they may reflect complex interactions between different correlated features. In contrast, a Bayes net parameter can be interpreted as a conditional probability, and reflects local statistics restricted to a family of nodes.
(2) Interpretable weights/scale. The log-difference transformation of the Bayes parameters means that weights can be interpreted as usual in a log-linear model: a positive weight indicates positive relevance (the parent values raise the probability of a child value), a negative weight as negative relevance (parent values lower the child value probability), and a zero weight as irrelevance.
(3) Scalable weight learning. The Bayes net parameters can be estimated using the empirical conditional frequencies observed in an input dataset $\mathcal{D}^*$: The parameter estimate for a family configuration is the number of instantiations of that family configuration in $\mathcal{D}^*$, divided by the sum of all instantiation counts for that family that agree on the parent values and vary the child values. In our notation, the estimate is defined by

$$\widehat{\theta}(T = t|\mathrm{Pa}(T) = \vec{t}_{pa}; \mathcal{D}^*) = \frac{\mathrm{n}\left[T = t, \mathrm{Pa}(T) = \vec{t}_{pa}; \mathcal{D}^*\right]}{\sum_{t' \in \mathrm{Ra}(T)} \mathrm{n}\left[T = t', \mathrm{Pa}(T) = \vec{t}_{pa}; \mathcal{D}^*\right]}. \tag{4}$$

A theoretical justification for using the observed conditional frequencies is that these estimates maximize a pseudo-likelihood function that measures how well a template BN matches an input dataset (?, ?). The pseudo-likelihood can be interpreted as the expected value of the log-likelihood of a random grounding of the BN nodes in the template model.
(4) Compatibility with regularization. Because 0 is the neutral point on the scale, the log-difference formulation is compatible with standard methods to regularize weights towards zero (?, ?).

Frequency estimates can be viewed as a type of **lifted learning**, the approach of using only the sufficient statistics for a dataset 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 (?).

In Section 7 we prove that, given mild assumptions about the Bayes net structure, the log-difference value is equivalent to a random grounding definition: the expected value of choosing a random grounding of the BN nodes in the template model, and applying the BN log-linear equation (3) to this random grounding. Thus the log-difference model can be viewed as an application of the random selection semantics for template Bayes nets (?). Before we describe the equivalence to random grounding inference, we provide experiments to evaluate our model's predictive performance. The next section presents several bases for comparison, alternative log-linear regression equations that can be computed in closed-form from a Bayes net template.

4. Alternative Log-Linear Equations

We consider three alternatives to the log-difference frequency equation. First, an alternative method for computing weights from Bayes net parameters: replace the log-difference of conditional probabilities with log-conditional probabilities. This is exactly the weight computation used in the BN log-linear equation (3). Second, a different feature function: Replace relevant feature frequencies by relevant counts. These alternatives define 4 different

equations, shown in Table 7. Table 8 shows the prediction of the gender of *sam* for each equation.

Table 7: Log-Linear Regression Equations for Relational Data.

Name	Weight	Function	Offset w_0
Log-difference frequency	$\ln \theta(U = u \mid \text{Pa}(U) = \vec{u}_{pa}) - \ln \theta(U = u)$	p^{r}	Yes
Log-cp frequency	$\ln \theta(U = u \mid \text{Pa}(U) = \vec{u}_{pa})$	p^{r}	No
Log-difference count	$\ln \theta(U = u \mid \text{Pa}(U) = \vec{u}_{pa}) - \ln \theta(U = u)$	n^{r}	Yes
Log-cp count	$\ln \theta(U = u \mid \text{Pa}(U) = \vec{u}_{pa})$	n^{r}	No

Table 8: The log-probability difference for the gender of *sam* in our running example, according to each of the equations. Notice that the two frequency equations give the same log-ratio, as entailed by Theorem 4.1. The frequency equations give the intuitively correct answer, that Sam is most likely to be a woman. The two count equations give the opposite answer, with log-cp weights by a greater margin than log-difference weights.

Name	log-linear sum for		log-ratio=
	$P(sam = \text{W} \mid \Lambda^*)$	$P(sam = \text{M} \mid \Lambda^*)$	column1 - column2
Log-difference frequency	-0.70	-0.75	0.05
Log-cp frequency	-1.06	-1.10	0.05
Log-difference count	-4.38	-0.63	-3.75
Log-cp count	-83.79	-60.17	-23.62

4.1 The Frequency Equations

The frequency equation in the second row of Table 7 uses the log-conditional probabilities of a child value given a parent configuation as weights. It does not normalize the conditional probabilities by the marginal probability of the child value, nor does it include an offset. Despite these differences, the two regression equations make the same prediction. This is true in general:

Frequency regression with log-conditional probabilities is equivalent to frequency regression with log-differences.

See the Appendix.

Although the two frequency equations are mathematically equivalent, the log-difference weights are more interpretable than the log-conditional probability weights: With log-cps, all weights are negative or zero, with a zero weight indicating infinite importance, rather than irrelevance.

4.2 The Count Equations

The count equations replace relevant frequencies with relevant counts as feature functions, but are otherwise the same as their frequency counterparts. While normalizing counts

to frequencies is the most direct and effective way to address the imbalance problem, to some extent the scales of different feature counts can be balanced through the weights, by assigning weights of smaller magnitude to features that tend to have larger counts. The example of Table 8 suggests that log-difference weights scale counts more effectively than log-cp weights. In the next section we report empirical results that confirm this finding on real-world benchmark datasets. We also report results for when weights are learned using general log-linear optimization methods, rather than computed from Bayes net parameters. These optimized weights also exhibit scaling effects, where features with larger counts tend to be assigned weights of smaller magnitude.

5. Empirical Comparison of Bayes Net Log-linear Equations

Our first set of experiments compared the predictive accuracy of different Bayes net regression equations. The next section describes experiments comparing the Bayes net methods with general weight learning for log-linear models.

5.1 Experimental Conditions and Metrics

All experiments were done on with 8GB of RAM and a single Intel Core 2 QUAD Processor Q6700 with a clock speed of 2.66GHz (there is no hyper-threading on this chip). The operating system was Linux Centos 2.6.32. Code was written in Java, JRE 1.7.0. All code and datasets are available (?).

5.1.1 Datasets

We describe the datasets in terms of their representation as databases with tables. The databases follow an Entity-Relationship (E-R) design (?). An E-R schema can be translated into our function-based logical notation as follows: Entity sets correspond to populations, descriptive attributes to functions, relationship tables to predicates, and foreign key constraints to type constraints on the arguments of relationship predicates. We used 5 benchmark real-world databases from prior work (?).

MovieLens Database This is a standard dataset from the UC Irvine machine learning repository. It contains two tables representing entity sets: User with 941 tuples and Item (Movies) with 1,682 tuples. The User table has 2 descriptive attributes, *age* and *gender*. We discretized the attribute *age* into three equal-frequency bins. The table Item represents information about the movies. It has 17 Boolean attributes that indicate the genres of a given movie. There is one relationship table Rated corresponding to a Boolean predicate. The Rated contains Rating as descriptive attribute; 80,000 ratings are recorded. We performed a preliminary data analysis and omitted genres that have only weak correlations with the rating or user attributes, leaving a total of three genres (Drama, Horror, Action).

Mutagenesis Database This dataset is widely used in Inductive Logic Programming research (?). We used a previous discretization (?). Mutagenesis has two entity tables, Atom with 3 descriptive attributes, and Mole (decribing molecules), with 5 descriptive attributes. There are two relationship tables, MoleAtom, indicating which atoms are

parts of which molecules, and Bond, which relates two atoms and has 1 descriptive attribute.

Hepatitis Database This data is a modified version of the PKDD02 Discovery Challenge database (?). The database contains information on laboratory examinations of 771 hepatitis B- and C-infected patients, taken between 1982 and 2001. The data are organized in 7 tables (4 entity tables, 3 relationship tables) with 16 descriptive attributes. They contain basic information about the patients, results of biopsy, information on interferon therapy, results of out-hospital examinations, and results of in-hospital examinations.

Mondial Database This dataset contains data from multiple geographical web data sources. We follow the modification of She *et al.* (?), and use a subset of the tables and discretized features: 2 entity tables, $Country, Economy$. The descriptive attributes of Country are continent, government, percentage, majority religion, population size. The descriptive attributes of Economy are inflation, gdp, service, agriculture, industry. A relationship table Economy_Country specifies which country has what type of economy. A self-relationship table Borders relates two countries.

UW-CSE database This dataset lists facts about the Department of Computer Science and Engineering at the University of Washington, such as entities (e.g., $Person$, $Course$) and the relationships (i.e. $AdvisedBy$, $TaughtBy$).

5.1.2 Prediction Metrics

We evaluate the algorithms using classification accuracy and conditional log likelihood (CLL). These metrics have been used in previous evaluations of MLN learning (?, ?). For each fact $\mathsf{T}^* = t$ in the test dataset, we evaluate the accuracy of the predicted Gibbs probability $P(\mathsf{T}^* = t|\Delta^*_{-\mathsf{T}})$, where $\Delta^*_{-\mathsf{T}}$ is a complete conjunction for all ground terms other than T^*. Thus $\Delta^*_{-\mathsf{T}}$ represents the values of the input variables as specified by the test dataset. For classification accuracy, a model's prediction is scored as correct if the true value of the ground term in the test dataset receives the highest Gibbs probability. CLL is the average of the logarithm of the Gibbs probability for each fact in the test dataset. Thus $\exp(CLL)$ is the geometric mean of the Gibbs probabilities.[6] Both metrics are reported as averages over all functors 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 datasets are nonbinary. The learning methods were evaluated using 5-fold cross-validation. Each database was split into 5 folds by randomly selecting entities from each entity table, and restricting the relationship tuples in each fold to those involving only the selected entities (i.e., subgraph sampling (?, ?)). The models were trained on 4 of the 5 folds, then tested on the remaining one. All results are averages from 5-fold cross validation, over all descriptive attributes in the database.

6. The geometric mean of a list of numbers $x_1, \ldots, x_n$ is $(\prod_i x_i)^{1/n}$.

5.2 Learning the Bayes Net Structure and Parameters

All the methods compared in this experiment require a prior Bayes net structure and parameters. To obtain the structure, the learn-and-join algorithm (?) was applied to each benchmark database. The parameters were computed from the empirical conditional frequencies in the database (Eq. 4) using previously-published algorithms (?). The resulting structure and parameters were used for all methods in this experiment. Table 9 shows the 3 log-linear equations compared with abbreviations. To determine the set of relevant features, we eliminated conjunctions that involved negated relationships (e.g., $\mathit{Friend}(\mathbb{A},\mathbb{B}) = \mathrm{F}$), for the following reasons. (i) We inspected the log-difference weights for these features and found them close to 0. Eliminating such weights approximates regularization. We leave a full regularization approach for eliminating irrelevant features for future work. (ii) Eliminating information from unrelated entities is standard practice in statistical-relational learning (?, ?). (iii) Log-linear weight learning methods (e.g., Alchemy, see Section 6) do not scale to our datasets when features with negated relationships are included (because of the difficulty of computing sufficient statistics for such features).

Table 9: The Bayes net log-linear equations compared in our experiments; cf. Table 7. Theorem 4.1 shows that the log(cp) + frequency and log-diff + frequency are equivalent.

		Feature Function	
		Count	Frequency
Weights	Log-CPs	log(cp) + count	log-diff + freq
	Log-diff.	log-diff + count	

5.3 Results

Table 10 summarizes the results for the Bayes net regression equations. The numbers represent an average over many individual scores, one for each fact in the database. For instance in the biggest dataset, MovieLens, the average is over a total of 170,000 scores; see Table 13.

To aid interpretability, we also report the following transformation of CLL: prob. ratio $= \exp(CLL(\mathit{method}) - CLL(\mathit{log}(\mathit{cp}) + \mathit{count}))$, where *method* is one of log-diff + count and log-diff + frequency. This quantity represents the geometric mean, over all test facts, of the fact likelihood ratio of *method* over the $\mathit{log}(\mathit{cp}) + \mathit{count}$ equation. For instance, the value of 1.05 for the dataset UW and the log-diff + count method means that on (geometric) average, the likelihood that the log-diff + count method assigns to the correct value for the target node is 1.05 times that assigned by the log-(cp) + count method.

5.4 Discussion

Frequency feature functions achieve top performance for classification accuracy. However, classification scores are similar across the three methods, and the difference between frequen-

Table 10: Conditional log-likelihood (log-probabilities) and classification accuracy (in percent) of Bayes net regression predictions. We show averages and standard deviations. The probability ratios take as the baseline the log(cp) + count method. They can be interpreted as the geometric average ratio of likelihoods assigned by the model to the true target node value.

Accuracy	UW	Mondial	MovieLens	Mutagenesis	Hepatitis
log(cp) + count	78 ± 0.08	40 ± 0.05	64 ± 0.01	62 ± 0.05	49 ± 0.03
log-diff + count	**81** ± 0.06	**45** ± 0.04	62 ± 0.02	**67** ± 0.03	**55** ± 0.02
log-diff + freq	**81** ± 0.06	**45** ± 0.04	**65** ± 0.01	**67** ± 0.03	**55** ± 0.02

CLL	UW	Mondial	MovieLens	Mutagenesis	Hepatitis
log(cp) + count	-0.47 ± 0.10	-1.47 ± 0.17	-1.19 ± 0.07	-0.84 ± 0.03	-1.33 ± 0.07
log-diff + count	-0.42 ± 0.05	-1.36 ± 0.11	-1.10 ± 0.16	-0.77 ± 0.03	-1.20 ± 0.07
Prob. ratio	1.05	1.12	1.09	1.07	1.14
log-diff + freq	**-0.41** ± 0.04	**-1.34** ± 0.09	**-0.71** ± 0.01	**-0.73** ± 0.04	**-1.07** ± 0.10
Prob. ratio	1.06	1.14	1.62	1.12	1.30

cies and counts + log-difference weights is small ($< 1\%$), except for a bigger improvement (3%) on MovieLens. Whereas classification accuracy is a 0-1 loss function, CLL is continuous, so the frequency approach's balancing of factors has substantially more impact on CLL. With respect to CLL, we observe the following ranking of methods on each dataset:

Frequency > log-difference count > count.

Therefore: *Using Bayes net parameters, frequency feature functions outperform count feature functions, with both the log-cp and the log-diff weight computation methods.* This finding supports our hypothesis that using frequencies as feature functions is an effective way of addressing the imbalance problem.

Applied to feature counts, log-difference weights improve on log-conditional probabilities. Our explanation for this finding is that log-difference weights are a heuristic for solving the imbalance problem, for the following reasons. Generally speaking, the number of instantiation count increases for a feature when the feature is based on longer relationship chains. In other words, (i) the number of instantiation counts increases with distance to the target entity. For instance, in the Bayes net of Figure 2, with target node $gender(sam)$, the feature $CoffeeDrinker(sam)$ is at distance 0 from Sam, and has instantiation count at most 1. The feature $Friend(sam, \mathbb{B}) = \mathrm{T}, gender(\mathbb{B}) = \mathrm{W}$ is at distance 1 from Sam (one link away), and its maximum instantiation count is the number of Sam's friends. A feature like $Friend(sam, \mathbb{B}) = \mathrm{T}, Friend(\mathbb{B}, \mathbb{C}) = \mathrm{T}, gender(\mathbb{C}) = \mathrm{W}$—referring to women friends of Sam's friends—is at distance 2 from Sam (two links away), and its maximum instantiation count is the number of friends of friends of Sam. We also expect that the correlation between Sam's gender and that of a friend will generally be stronger than than between Sam's gender and the gender of a friend of a friend. Therefore we can expect that (ii) the probabilistic association of the target node with related entities decreases with distance to

the target entity. (iii) As the probabilistic association decreases, so does the log-difference weight since it measures the strength of the probabilistic association. Combining the observations (i), (ii), (iii) entails that features with high instantiation counts tend to have low log-difference weights. Therefore log-difference weights address the imbalance problem by tending to assign lower weight magnitudes to features with high instantiation counts. Table 11 summarizes the quantities related to this analysis and how they relate to each other.

Table 11: Connections between different quantities for features that are consistent with the observed performance of count feature functions.

Distance to Target Entity	Feature Instantiation Count	Correlation with Target Attribute	Log-Difference Weight
+ increases	+ increases	- decreases	- decreases

6. General Weight Learning Experiments

The experiments in Section 5 held the Bayes net structure and parameters constant and compared transformations of the BN parameters into log-linear weights. In this section we examine a setting where the same features are computed from the BN structure, but weights are *not* computed from the BN parameters. Instead weight values are optimized by a local search method. This experiment used the same conditions and metrics described in Section 5.1. In addition to comparing predictive performance, we also report learning times.

To learn the weights, we applied the default training procedure of the Alchemy package (?). This procedure takes as input a set of features specified as logical formulas, and returns a weight for each formula. We followed the method recommended by the Alchemy group (?) for converting a Bayes net structure to a Markov Logic Network structure: For each family configuration F_{ijk} in the BN, add a conjunction of literals that specifies the state. We also added unit clauses for each node-value combination, as recommended by the Alchemy group; unit clause weights can represent the bias weight of a log-linear equation.

We refer to moralization+weight learning as the **MBN** method, for "Moralized Bayes Net" (?). MBN has been the state-of-the-art method for log-linear prediction with Bayes nets (?). Figure 5 shows the program flow for the MBN method. Markov Logic Network weight learning optimizes for log-linear inference with counts as feature functions (?). The prediction probabilities were computed exactly using the log-linear equation 2 with counts as feature functions. We used an exact computation rather than approximate inference (e.g., MC-SAT), to avoid confounding the effect of the log-linear equation with that of inference implementation. Experiments with MC-SAT produced similar results. We also computed the results with frequencies as feature functions, which for optimized weights were very similar, so we do not present them.

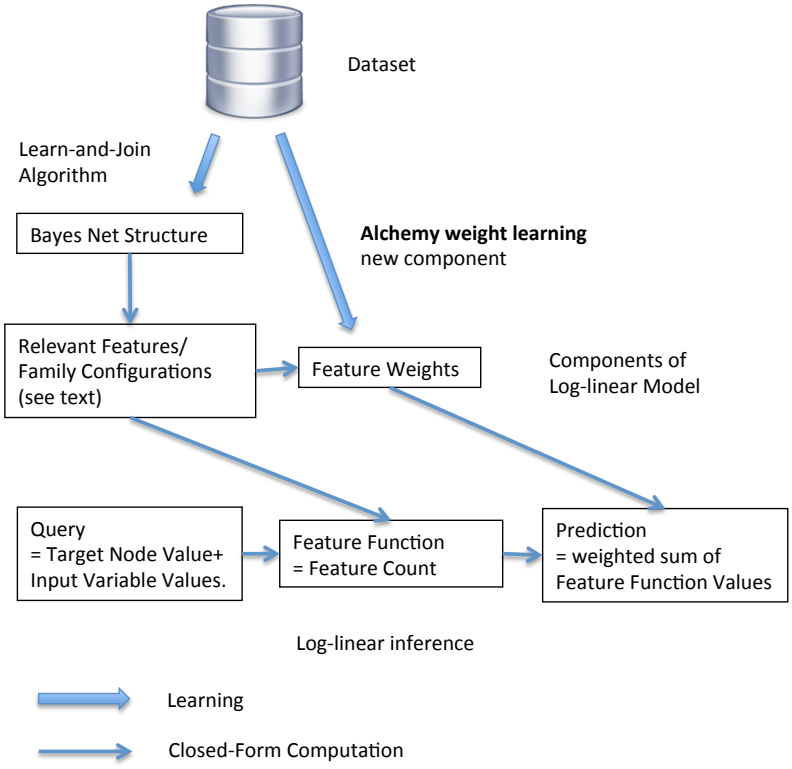

Figure 5: The MBN method: Program flow for computing relational Gibbs probabilities with general weight learning. A BN template structure is learned from the data using the learn-and-join algorithm. Relevant features are computed from the BN structure. Feature weights are computed by applying Markov Logic Network weight learning to the data. The Bayes net parameters are not estimated.

6.1 Results

The Bayes net frequency regression predictions are competitive with those from a model with optimized general weights.
Accuracy. Table 12 shows the scores of the MBN method, together with the log-difference frequency results from Table 10. The log-difference frequency model scores slightly higher than the MBN weights on every dataset, with the biggest differences on Mutagenesis (5%), MovieLens (5%) and Hepatitis (4%).

Table 12: Conditional log-likelihood (log-probabilities) and classification accuracy (percent) of MBN and log-difference frequency predictions.

Accuracy	UW	Mondial	MovieLens	Mutagenesis	Hepatitis
MBN	80 ± 0.05	44 ± 0.04	60 ± 0.02	62 ± 0.02	51 ± 0.02
log-diff + freq	**81** ± 0.06	**45** ± 0.04	**65** ± 0.03	**67** ± 0.03	**55** ± 0.02

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-diff + freq	**-0.41** ± 0.04	-1.34 ± 0.09	**-0.71** ± 0.01	**-0.73** ± 0.04	**-1.07** ± 0.10
Prob. ratio	1.03	0.94	1.08	1.20	1.12

CLL. The log-difference frequency model scores better than MBN model on UW, MovieLens, Mutagenesis and Hepatitis (probabilities 3–20% higher) and scores slightly worse on Mondial (6% lower probability); see Figure 6.
Learning Times. Table 13 shows run time results for structure and parameter learning. We see *clear scalability advantages for the maximum likelihood conditional probability estimates used in the Bayes approach*: they take seconds to compute, whereas Alchemy weight optimization requires as much as 10 hours in the worst case (Hepatitis).

6.2 Evidence for Scaling

The boxplots in Figure 7 compare the spread of the weights learned by Alchemy and the corresponding spreads for the log-CP and log-difference methods. The plots separate *1-variable formulas* that contain only one population variable from *2-variable formulas* that contain more than one population variable. The 1-variable formulas have just one grounding for a given target node, whereas 2-variable formulas have many. If the computed weights include a scaling component, we expect that the absolute size of weights will be smaller for 2-variable formulas.

The optimized MBN-weights show clear scaling effects on five databases, with the biggest effect in the MovieLens weights. This is also the dataset where the Bayes net frequency model outperforms the count model the most on the CLL metric (Table 10). Overall, *the observed weight magnitudes provide evidence that optimal weights are scaled to balance the diverging instantiation counts for different features.*

The log-difference weights also show scaling effects on four of the databases, with the largest effect again occuring for MovieLens. This confirms our expectation that information

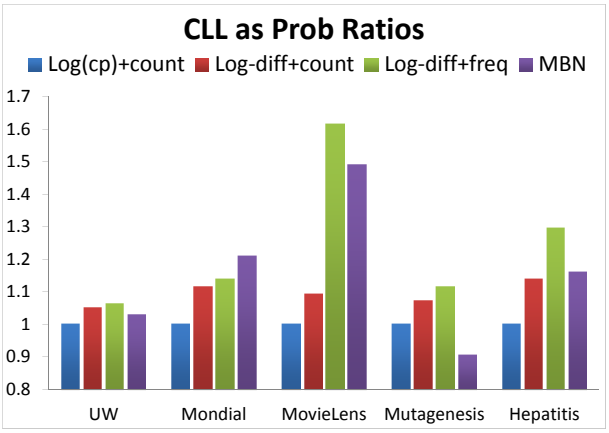

Figure 6: Predictive Performance averaged over all five benchmark databases. With Bayes net parameters, the frequency model performs better than the count model in terms of likelihood ratios.

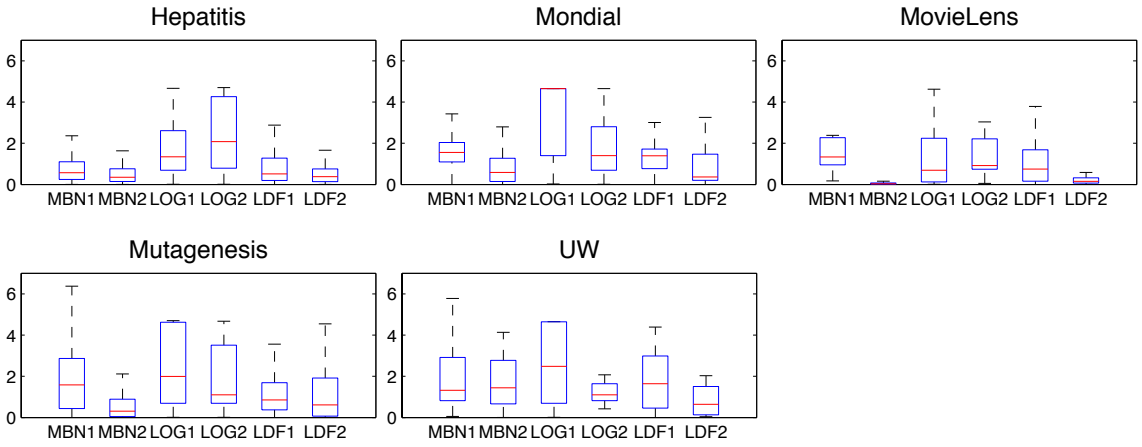

Figure 7: The absolute magnitudes of weights learned from five databases using three estimators for two sizes of formula. MBN=Moralized Bayes Nets, LOG=log(cp), LDF = log-diff. The suffix of 1 or 2 indicates the number of population variables in the formulas. Whiskers represent the 95th percentile.

Table 13: Parameter learning times for MBN and Bayes net methods. We characterize each database by its counts of ground atoms, tuples, and Bayes net parameters, and its time for structure learning. The Bayes net parameter count is the number of family configurations in the Bayes net.

Dataset	Literals (×1000)	Tuples (×1000)	Parms. (×10)	Struct. (s)	MBN (s)	Bayes (s)
UW	3	1	12	36	5	**2**
Mondial	2	1	58	12	90	**3**
MovieLens	170	82	33	72	10800	**8**
Mutagenesis	35	15	88	30	14400	**3**
Hepatitis	71	15	79	24	36000	**3**

from related entities is less important, although the scaling effect is not as great as with the optimized MBN weights.

Summary. The findings from this section and the previous one support our claim that balancing the scales of feature functions is important for using Bayes net parameters in a log-linear model. The combination of transformed BN parameters +feature frequencies is as predictively accurate as using feature counts with optimized general weights. While the Bayes net log-linear model is comparable in accuracy, its parameters can be learned much faster than general log-linear weight learning.

7. Random Selection Inference

The experimental results in the previous two sections indicate that the log-difference frequency equation achieves good predictive accuracy, in addition to scalability and interpretability. In this section, we extend our theoretical understanding of the Bayes net freuquency equations by providing a different deriviations of this method.

The standard BN product formula defines a joint distribution over template nodes. Recall that the Gibbs conditional probability for this joint distribution is given by the log-linear equation 3. In this section we show that frequency regression can be interpreted as the expected value of Equation 3, over *random* groundings of the template Bayes net. This value can be computed as follows for a given template Bayes net B and query.

1. Choose a random grounding γ' of all population variables in B. The grounding γ' must be consistent with the target node grounding γ. A random grounding selects a constant for each population variable independently and uniformly.

2. Apply the random grounding to obtain a fully ground Bayes net $B_{\gamma'}$. The query conjunction defines a unique value for each node in $B_{\gamma'}$. Apply the standard Bayes net equation 3 to compute a log-linear score for the query in the single ground BN.

3. Return the average log-linear score over all groundings γ'.

We need to refine this basic idea to eliminate irrelevant features, i.e., parent-child configurations. A simple way to do this is to consider only groundings such that all features defined by the grounding are relevant. Algorithm 2 provides pseudo-code for this computation. Table 14 provides a sample computation of a random selection value for predicting the gender of *sam* given the database instance of Figure 2.

Algorithm 2: Computation for Random Selection Inference.

Input: Template Bayes Net B; Query $P(\mathsf{T}^*_\gamma = t|\Delta^*) =?$

Output: Average log-linear score for a random grounding of the template BN.

1: Let $\mathbb{A}_1, \ldots, \mathbb{A}_a$ be the population variables in B_γ.
2: **for all** groundings γ' of $\mathbb{A}_1, \ldots, \mathbb{A}_a$, such that γ' extends the query grounding γ **do**
3: **for all** nodes U **do**
4: Let $(U = u_{\gamma'}, \text{Pa}(U) = \vec{u}_{pa,\gamma'})$ be the family configuration assigned by the query conjunction to the nodes $U, \text{Pa}(U)$ after applying the grounding γ'.
5: **end for**
6: **if** for all $U \in \{T\} \cup \text{Ch}(T)$, the family configuration $(U = u_{\gamma'}, \text{Pa}(U) = \vec{u}_{pa,\gamma'})$ is relevant **then** $\{\gamma'$ is relevant$\}$
7: $score_{\gamma'} := \sum_{U \in \{T\} \cup \text{Ch}(T)} \ln \theta(U = u_{\gamma'}|\text{Pa}(U) = \vec{u}_{pa,\gamma'})$ {see Equation 3}
8: **end if**
9: **end for**
10: **return** The average of the relevant scores $score_{\gamma'}$.

Notice that the frequency equations and random selection inference lead to exactly the same value. The next theorem provides a general proof for *the equivalence between frequency and random selection inference*. The general reason for the equivalence is this: Observe that the log-cp frequency equation is a sum of expected values, while the random selection value is the expected value of a sum. As is well known, the sum of expected values of random variables equals the expected value of their sum.

Suppose that all child-parent configurations of the target node are relevant. Then the Gibbs conditional probabilities defined by the following are equal:

1. Log-difference frequency regression.
2. Frequency Regression.
3. Random regression.

See the Appendix.

Discussion. We remark that the above equivalence holds for any graphical model based on a template of logical terms, not only Bayes nets. As Table 14 illustrates, the equivalence can hold even when some features are irrelevant (e.g., the gender of persons who are not friends with Sam). It suffices for random selection inference to be equivalent to the frequency equation if for each family configuration, the expected value of its log-linear score, with respect to groundings of the family nodes only, equals the expected value of its log-linear score with respect to groundings of *all* Bayes net nodes. This is guaranteed if for the population variables that are not ground by the query, the set that appears in the parents

Feature #	Family Configuration
1	$g(sam) = W$,$g(B) = W$, $F(sam, B) = T$)
2	$g(sam) = W$,$g(B) = \mathrm{M}$, $F(sam, B) = T$)
3	$cd(sam) = T, g(sam) = \mathrm{W}$

Grounding	Child = $g(B)$	CP1	Child = $cd(sam)$	CP2	ln(CP1)+ln(CP2)
y1	Feature 1	0.55	Feature 3	0.80	-0.82
...	...	...	...	...	
y40	Feature 1	0.55	Feature 3	0.80	-0.82
y41	Feature 2	0.37	Feature 3	0.80	-1.22
...	...	...	...	...	
y100	Feature 2	0.37	Feature 3	0.80	-1.22
				Average	-1.06

Table 14: Computing the random regression for target node value $gender(sam) = \mathrm{W}$. CP = conditional probability. The first 40 rows show the log-linear features for applying the nonrelational Bayes net regression equation 3 when the population variable $\mathbb{B}$ is instantiated with one of *sam*'s female friends (cf. Table 6). The next 60 rows show the log-linear factors for instantiating the population variable $\mathbb{B}$ with one of *sam*'s male friends. The random regression result is the average over all the instantiations. Note that the random regression result -1.06 is exactly the same as the result for frequency regression (Table 8, row 2, column 1).

of the target node is disjoint from the set that appears in the children and spouses of the target node. In our running example, the disjointness condition holds since the only child of the target node is $CoffeeDr(\mathbb{A})$ and $\mathbb{A}$ is ground by the query (i.e., $\mathbb{A}\backslash sam$). Another sufficient condition for the equivalence is that the relevant family configurations can be defined by a uniform condition for the children and parents of the target node, such as $Friend(\mathbb{A}, \mathbb{B}) = \mathrm{T}$, which is also true in our running example. The close correspondence between random selection inference and the frequency equations completes our argument in support of the frequency log-linear equations.

We discussed key differences between our approach and KBMC template semantics in Section 1.4. The next section provides further details about relevant related work in statistical-relational learning.

8. Related Work

We first discuss work on directed graphical models for relational data, then consider other graphical model classes such as Markov and dependency networks. (?) *et al.* compare Bayes, Markov and dependency nets in detail for nonrelational data.

Directed Graphical Models There are several proposals for defining directed relational template models, based on graphs with directed edges or rules in clausal format (?, ?, ?, ?). While the template model need not be a Bayes net (in contrast with our Template Bayes nets), the usual application of KBMC semantics requires that the instantiated model should be a Bayes net (cf. Section 1.4). In order to define the probability of a child node conditional on multiple instantiations of a parent set, template semantics for directed models therefore

requires the addition of combining rules (?) or aggregation functions (usually with extra parameters) (?). As described by (?), aggregate functions can be added to a Template Bayes net by including functor nodes with aggregates (e.g., $AvgGrade(\mathcal{S})$ may represent the average grade of a student). Combining rules such as the arithmetic mean (?) combine global parameters with a local scaling factor, as does our log-linear model. Our frequency model is similar to a combining rule using the *geometric mean* rather than the arithmetic mean, with the important difference that the geometric mean is applied to the entire Markov blanket of the target node, whereas usually a combining rule applies only to the parents of the target node. To our knowledge, the geometric mean has not been used with ground Bayes nets before.

Markov Networks Relational log-linear models are usually associated with undirected graphs (?), such as Relational Markov networks (?) and Markov Logic Networks (?). To our knowledge, this is the first paper that specifies a log-linear model for Bayes nets directly. Several researchers have examined converting a Bayes net relational model to a Markov net. Since Markov nets define a log-linear inference model, this conversion strategy entails using a log-linear model for Bayes nets. Richardson and Domingos propose converting a Bayes net to a Markov Logic network using moralization, with log-conditional probabilities as weights (?). This is also the standard Bayes net conversion recommended by the Alchemy system (?). The moralization method inferences are equivalent to our equation log(cp)+ count. To our knowledge, our experiments are the first that evaluate the log(cp)+count equation, and our three comparison equations are novel. Natarajan et al. (?) consider moralization with Bayes nets that have been augmented with combining rules. We consider tabular Bayes nets whose parameters are CP-table entries only. Combining rules do not generally lead to log-linear models.

Dependency Networks Dependency networks are directed graphical models that allow cycles (?). For instance, the ground BN of Figure 1(bottom) is a valid dependency net structure. Relational Dependency networks (RDNs) (?) have been an influential recent development because they can accommodate the cyclic dependencies that are common in relational data. The parameter space of dependency networks comprises the set of Gibbs conditional probabilities. The RDN solution to the cyclicity problem is therefore the same as what we propose in this paper: define local Gibbs probability distributions rather than a single acyclic global model. The original RDN work used aggregate functions to combine different instantiations of a target node's Markov blanket. Learning dependency networks with log-linear relational models was investigated by Khot *et al.* (?); this work used feature counts, not frequencies. In this paper, we do not learn Gibbs conditional distributions directly. Instead, we apply previous algorithms to learn a BN structure, and derive Gibbs probabilities from the learned structure and maximum (pseudo)-likelihood estimates of the BN parameters.

In sum, while log-linear models for Gibbs probabilities have been investigated for other graphical model classes, ours is the first multi-relational model of this type for Bayes nets. Our use of frequencies vs. counts for feature functions is new.

9. Conclusion and Future Work

This paper presented a new log-linear inference equation for applying Bayes nets to relational data. For a fixed template Bayes net, the equation defines the Gibbs conditional probability of a target node given an assignment of values to all other nodes. A log-linear model is defined by: a set of features, and for each feature, a feature function and a feature weight. The predicted conditional probability is the exponentiated weighted sum of feature function values. In our proposed model, the features are all family configurations in for a family in the Bayes net, whose child node is either the target node or a child of the target node. The weight associated with a child-parent feature is computed as the log-difference of two quantities determined by the BN parameters: (i) the conditional probability of the child value, given its parent configuration, and (ii) the marginal probability of the child value. The feature function is the frequency with which the feature is instantiated in the given query, normalized with respect to relevant features only.

Our experiments on five benchmark datasets compared our log-difference frequency equation to several alternatives: using counts as feature frequencies, and using log-conditional probabilities as feature weights. We also compared using transformed Bayes net parameters as weights to using weights directly learned from the data by log-linear optimization methods. Our frequency equation achieved the best predictive performance on all but one dataset. Using the maximum likelihood values as Bayes net parameters is much faster than optimizing weights using standard log-linear methods (Markov Logic), typically seconds vs. hours.

Different model classes each have their advantages and disadvantages. Nonetheless the combination of Bayes nets and our proposed log-linear model, offers a unique set of advantages compared to other inference methods for multi-relational data, in terms of the *interpretability* and *scalability* of both structure and parameter learning: Feature weights are readily interpreted as a log-transformation of the Bayes net conditional probability parameters, and the Bayes net parameters can be computed in closed-form as the empirical frequencies.

We have established novel connections between the use of frequency feature functions and what, at first sight, appear to be unrelated issues such as the imbalance problem, pruning irrelevant features, maximum likelihood estimation, and the random selection interpretation of template Bayes nets. (1) The imbalance problem arises because feature instantiation counts in relational data can diverge by orders of magnitude. Rescaling counts as frequencies produces feature function values on the same scale. According to our experiments, changing the feature function is a more effective approach to the imbalance problem than using the weight parameters to rescale (assign smaller weight magnitudes to larger feature counts). (2) It is important not only to prune irrelevant features, but also to define instantiation frequencies over the space of relevant features only. (3) Maximum likelihood estimation is competive with optimizing weight parameters from the data only when relevant feature frequencies are used as feature functions. (4) Under mild assumptions, our frequency equation is equivalent to a *random selection* method, where the prediction score for a target node value is defined as the expected score, with respect to a random instantiation of the template Bayes net, computed using the standard Bayes net equation for a Gibbs conditional probability.

There are several avenues for future work. While we focus on Bayes nets, the imbalance 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. (?, ?, ?). The frequency equation can be combined with other log-linear learning methods, for example within a model ensemble. Functional gradient boosting (?) is a powerful technique for learning such ensembles.

Our model introduces a 1-1 correspondence between log-linear weights and Bayes net parameters. Therefore log-linear regularization techniques (?) can be used for smoothing parameter estimates in template Bayes nets, and for detecting irrelevant features.

Local Gibbs probability models may be inconsistent in the sense that there is no joint distribution that agrees with the local conditional probabilities (?). An open theoretical question is whether our local frequency equations for different target nodes are guaranteed to be mutually consistent. If they are inconsistent, a possible approach is to apply the recent averaging methods for dependency networks (?, ?).

Our log-linear equation with relevant feature frequencies appears to be a principled, fast-to-learn, and accurate model for relational prediction with Bayes nets.

Acknowledgements

This work was supported by Discovery Grants to Oliver Schulte from the Natural Science and Engineering Council of Canada. Zhensong Qian was supported by a grant from the China Scholarship Council. A preliminary versions of this paper was presented at the StarAI 2012 workshop. We are indebted to workshop reviewers and participants for helpful comments.

Appendix: Proofs

Our two theorems concern the frequency regression equation, where the predictors are relevant frequencies and the weights are log-conditional probabilities. The formal definition of the **log-cp frequency equation** is

$$P(\mathsf{T}^*_\gamma = t|\Delta^*) \propto \exp \sum_U \sum_{u,\vec{u}_{pa}} \ln \theta(U = u|\mathrm{Pa}(U) = \vec{u}_{pa}) \cdot \mathrm{p^r}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right]. \quad (5)$$

Frequency regression returns the same result when used with log-cp weights and when used with log-difference weights.

Our approach is to show that the log-difference equation can be factored such that (i) the bias weight is cancelled out, and (ii) the difference between the result and the frequency equation is a constant factor that is independent of the target node value. Thus the normalized probability defined by each equation is the same.

We begin with some observations about the independence of relevant counts from the target node value. The first observation is that for a node that is not the target node, *the relevant count of any of its family configurations is independent of the target node value*:

$$\mathrm{n^r}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right] = \mathrm{n^r}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t', \Lambda^*\right] \quad (6)$$

for any target node values t resp. t'. The reason for the independence is that a true family grounding with t can be changed to a true family grounding where the only change is that the target node is assigned value t' rather than t.

If the target node is the child in the family configuration, the relevant count is, for the same reason, independent of the target node value, as long as the child node value is consistent with the target node value

$$\mathrm{n}^{\mathrm{r}}\left[\mathsf{T}^*_\gamma = t, \mathrm{Pa}(\mathsf{T}_\gamma) = \vec{t}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right] = \mathrm{n}^{\mathrm{r}}\left[\mathsf{T}^*_\gamma = t', \mathrm{Pa}(\mathsf{T}_\gamma) = \vec{t}_{pa}; \mathsf{T}^*_\gamma = t', \Lambda^*\right] \tag{7}$$

for any target node values t resp. t'. If the assignment to the child node is inconsistent with the target node value, the number of true instantiations is 0:

$$\mathrm{n}^{\mathrm{r}}\left[\mathsf{T}^*_\gamma = t, \mathrm{Pa}(\mathsf{T}_\gamma) = \vec{t}_{pa}; \mathsf{T}^*_\gamma = t', \Lambda^*\right] = 0 \tag{8}$$

whenever $t \neq t'$. The **marginal frequency** of a child node value is obtained by summing over the frequencies of family configurations with that child node value:

$$\mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u; \mathsf{T}^*_\gamma = t, \Delta^*\right] \equiv \sum_{\vec{u}_{pa}} \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right]$$

For the special case in which the target node is the child in the family, all consistent family configurations agree with the target node value, which therefore has marginal frequency 1. In other words, Equation (8) implies that

$$\mathrm{p}^{\mathrm{r}}\left[\mathsf{T}^*_\gamma = t; \mathsf{T}^*_\gamma = t, \Delta^*\right] = 1. \tag{9}$$

From Equation (9) and the definition of marginal frequency we have the equalities

$$\begin{aligned}
&\sum_U \sum_u \sum_{\vec{u}_{pa}} \ln \theta(U = u) \cdot \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right] \\
&= \sum_U \sum_u \ln \theta(U = u) \sum_{\vec{u}_{pa}} \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right] \\
&= \ln \theta(T = t) \cdot 1 + \sum_{U \neq T} \sum_u \ln \theta(U = u) \cdot \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u; \mathsf{T}^*_\gamma = t, \Delta^*\right]
\end{aligned} \tag{10}$$

We next factor the log-difference equation by multiplying out the log-difference. To simplify the expressions, we work with the linear sum inside the exp expression.

$$\ln\theta(T=t) + \sum_U \sum_{u,\vec{u}_{pa}} [\ln\theta(U=u|\mathrm{Pa}(U)=\vec{u}_{pa}) - \ln\theta(U=u)] \cdot \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right]$$

$$= \ln\theta(T=t) + \sum_U \sum_{u,\vec{u}_{pa}} \ln\theta(U=u|\mathrm{Pa}(U)=\vec{u}_{pa}) + \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right]$$

$$- \sum_U \sum_{u,\vec{u}_{pa}} \ln\theta(U=u) \cdot \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right]$$

$$= \ln\theta(T=t) + \sum_U \sum_{u,\vec{u}_{pa}} \ln\theta(U=u|\mathrm{Pa}(U)=\vec{u}_{pa}) \cdot \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right]$$

$$- \ln\theta(T=t)) - \sum_{U\neq T} \sum_u \ln\theta(U=u) \cdot \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u; \mathsf{T}^*_\gamma = t, \Delta^*\right]$$

$$= \sum_U \sum_{u,\vec{u}_{pa}} \ln\theta(U=u|\mathrm{Pa}(U)=\vec{u}_{pa}) \cdot \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \mathsf{T}^*_\gamma = t, \Delta^*\right]$$

$$- \sum_{U\neq T} \sum_u \ln\theta(U=u) \cdot \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u; \mathsf{T}^*_\gamma = t, \Delta^*\right]$$

The last step follows by cancelling out the marginal log-probabilities $\ln\theta(T=t)$. The last-but-one-step follows from Equation (10). Comparing the exponential of the last line and the log-cp frequency equation (5), we see that the only difference between the log-difference and the log-cp frequency equation is the factor

$$\exp\left(-\sum_{U\neq T}\sum_u \ln\theta(U=u)\cdot \mathrm{p}^{\mathrm{r}}\left[\mathsf{U}_\gamma = u; \mathsf{T}^*_\gamma = t, \Delta^*\right]\right).$$

Clearly the marginal probability $\theta(U=u)$ does not depend on the target node value when $U \neq T$. Also, Equation (6) implies that the relevant family configuration counts do not depend on the target node value. Therefore neither do the relevant family configuration frequencies. Hence the displayed term is independent of the target node value. This means that the log-cp and log-difference frequency regression equations differ only by a factor that is independent of the target node value. Therefore they agree on the prediction for the normalized probability of the target node value.

Suppose that all child-parent configurations of the target node are relevant. Then the frequency regression value for a target node (Equation (5)) equals the random regression value.

Let B be the partially ground template model that results from applying the target grounding. We write Γ for the set of all groundings of *all* population variables in the partially ground graph B. For each partially ground family formula $\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}$, let

$$\mathrm{N}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right]$$

be the number of simultaneous groundings of *all* variables in B that satisfy $\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}$. Consider a family formula $(\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa})$ whose child is the target node or one of its children. For each simultaneous grounding $\gamma \in \Gamma$ that satisfies the family formula

in the query conjunction, the associated factor $\ln \theta(U = u|\mathrm{Pa}(U) = \vec{u}_{pa})$ appears once in the logarithm of the non-relational Gibbs probability equation 3. Therefore random regression is equivalent to the formulas

$$P(\mathsf{T}^*_\gamma = t|\Delta^*) \propto$$

$$\exp(\frac{1}{|\Gamma|}\sum_U \sum_{u,\vec{u}_{pa}} \ln \theta(U = u|\mathrm{Pa}(U) = \vec{u}_{pa}) \cdot \mathrm{N}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right]) \tag{11}$$

$$= \exp(\sum_U \sum_{u,\vec{u}_{pa}} \ln \theta(U = u|\mathrm{Pa}(U) = \vec{u}_{pa}) \cdot \frac{\mathrm{N}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right]}{|\Gamma|})$$

where the sum ranges over family configurations of the target node and its children as usual. To establish the equivalence of Equation (11) with the log-cp frequency equation (5), it suffices to show that the count fraction in the last random regression equality is another way of computing the frequency of a family configuration. In symbols, we show that

$$\mathrm{p}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right] = \frac{\mathrm{N}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right]}{|\Gamma|} \tag{12}$$

The equivalence with relevant frequency regression follows from the theorem's premise that the relevant frequency $\mathrm{p}^{\mathrm{r}}[;]$ and the unqualified frequency $\mathrm{p}[;]$ are the same.

We introduce some notation to establish Equation 12, which says that the frequency of a family configuration can be computed in terms of groundings for all variables in the Bayes net. The basic reason why this is true is that, compared to groundings for variables in the family only, the number of possible groundings for the family increases by a constant factor that cancels out when we divide by the total number of possible groundings $|\Gamma|$. Write m_{U_γ} for the number of possible groundings of the population variables that occur in the family of child node U_γ. Write $\bar{m}_{\mathsf{U}_\gamma}$ for the number of possible groundings of the population variables that do *not* occur in the family of child node U_γ. For instance, if variables $\mathbb{A}_1, \mathbb{A}_2$ occur in the family of U_γ, and variable $\mathbb{A}_3$ does not, then $m_{\mathsf{U}_\gamma} = |\mathcal{P}_{\mathbb{A}_1}| \cdots |\mathcal{P}_{\mathbb{A}_2}|$, and $\bar{m}_{\mathsf{U}_\gamma} = |\mathcal{P}_{\mathbb{A}_3}|$ where $\mathcal{P}_{\mathbb{A}_i}$ is the population associated with variable $\mathbb{A}_i$. Then we have

$$\begin{aligned}
|\Gamma| &= m_{\mathsf{U}_\gamma} \cdot \bar{m}_{\mathsf{U}_\gamma} \\
\mathrm{N}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right] &= \mathrm{n}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right] \cdot \bar{m}_{\mathsf{U}_\gamma} \\
\frac{\mathrm{N}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right]}{|\Gamma|} &= \frac{\mathrm{n}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right]}{m_{\mathsf{U}_\gamma}} \\
= \mathrm{p}\left[\mathsf{U}_\gamma = u, \mathrm{Pa}(\mathsf{U})_\gamma = \vec{u}_{pa}; \Delta^*, \mathsf{T}_\gamma = t\right]
\end{aligned}$$

The first, second, and fourth equality follow from the definition of the relevant concepts. The third equation follows from the first two after cancelling $\bar{m}_{\mathsf{U}_\gamma}$ from numerator and denominator. The equations entail the alternative expression (12) for the frequency of a family configuration. This suffices to establish the equivalence of the frequency equation (5) with the random regression formula (11).