Consistent Bayesian Network Scores for Multi-Relational Data

Oliver Schulte, Sajjad Gholami
School of Computing Science
Simon Fraser University
Vancouver-Burnaby, Canada

Abstract

We describe a new method for extending, or upgrading, a Bayesian network (BN) score designed for i.i.d. data to multi-relational data. The method defines a gain function that measures the difference in data fit between two structures. Our upgrade method *preserves consistency*, in the sense that if the i.i.d. score is consistent for i.i.d. data, the upgraded version is consistent for multi-relational data. Empirical evaluation on six benchmark relational databases shows that our gain function method finds a set of informative edges that strikes a balance between overly sparse and overly dense graph structures. A surprising negative finding is that for log-linear relational BN likelihood scores, we could not identify a consistency-preserving model score that is a function of a single model only. Theoretical analysis shows that the reason for the difficulty is that in log-linear models, the number of potential instantiations can differ exponentially for different features.
keywords: Model Selection, Statistical-Relational Learning, Bayesian Networks

Introduction

Learning Bayesian networks for relational data is a major approach to statistical-relational learning (Getoor and Taskar 2007). The most widely used approach to Bayesian network structure learning is search+score, which aims to find a strucure that optimizes a model selection score for a given dataset. Independently distributed data represented in a single table can be viewed as a special limiting case of multi-relational data with no relationships (Nickel et al. 2015; Neville and Jensen 2007). Extending a learning method defined for i.i.d. data to multi-relational data is called *upgrading* the method (Laer and de Raedt 2001). In this paper we propose a general definition for upgrading Bayesian model selection scores for i.i.d. data to relational data. Our upgrade method satisfies two desiderata.

Generalization The i.i.d. model comparison is a special case of the upgraded multi-relational model comparison (Laer and de Raedt 2001; Knobbe 2006).

Preserving Consistency If the i.i.d. model comparison is consistent for i.i.d. data, the upgraded comparison is consistent for multi-relational data.

Consistency is a widely applied theoretical criterion for model selection (Williams 2001) for i.i.d. data, and increasingly for relational data as well (Sakai and Yamanishi 2013; Xiang and Neville 2011; Shalizi and Rinaldo 2013). Informally, it means that as the amount of available data increases, the method selects a correct model structure that matches the true data generating process.

Approach. We focus on log-linear likelihood scores, which are weighted sums of sufficient statistics (Sutton and McCallum 2007). For Bayesian networks, the sufficient statistics are the dataset instantiation counts/frequencies of the possible child-parent configurations (the features). A surprising negative result for is that we have not been able to identify an upgraded score function that achieves our three criteria by assigning a score to a *single* relational model. Intuitively, the cause of non-consistency is the ill-conditioning of log-linear models (Lowd and Domingos 2007): the number of potential instantiations can differ exponentially for different features. Our proposed solution is to use a relational *gain* function that first rescales the scores for different structures G, G', to balance the number of potential instantiations, and then compares the rescaled scores. Experiments indicate that the gain functions select informative edges that provide a good statistical fit to the input data.

Contributions. Our main contributions may be summarized as follows.

1. A novel method for upgrading an i.i.d. model structure score to relational data. The method upgrades the score by defining a gain function that compares the relative fit of two graph structures on relational data.
2. A consistency preservation theorem: if the original score is consistent for i.i.d. data, the upgraded gain function is consistent for relational data. To our knowledge this is the first consistency result for multi-relational structure learning.

Paper Organization. We review background on Bayesian networks and relational data. Then we define our rescaling method for upgrading model selection scores, as well as baseline upgrade methods for comparison. Theoretical analysis demonstrates the consistency of the rescaling method,

and the non-consistency of the baseline scores. Empirical evaluation compares the different Bayesian networks selected with respect to data fit.

Related Work

BN Model Selection. Model selection criteria for BN learning are a classic topic in machine learning; for review see (Bouckaert 1995). Our work extends the theoretical analysis of BN learning to multi-relational data. For relational data, a likelihood function requires aggregating model probabilities for multiple instances of the template BN (Kimmig, Mihalkova, and Getoor 2014). A number of different aggregation mechanisms have been proposed, resulting in different BN likelihood functions.

Likelihood based on Random Instantiations. The most recent proposal is the *random selection pseudo log-likelihood* (Schulte 2011). The random selection log-likelihood score is defined as the expected log-likelihood from a random grounding. The frequency log-likelihood of Table 2 that we utilize in this paper is a closed form for the random selection log-likelihood (Schulte 2011, Prop.4.1).

Likelihood based on Complete Instantiations. This type of likelihood is based on unrolling or grounding the BN (Neville and Jensen 2007; Poole 2003). An inference graph contains all possible instances of edges in the first-order template, obtained by replacing first-order variables with constants. The inference model defines a conditional probability $P(X_{ij}|\mathbf{Pa}_{ij}^G)$, where X_{ij} denotes the j-th grounding of node i in the template BN. This conditional probability aggregates the information from the multiple parent instances of the ground node X_{ij}. There are two main approaches for defining the conditional probability model $P(X_{ij}|\mathbf{Pa}_{ij}^G)$, depending on how multiple instances of the same parent configuration are included (Natarajan et al. 2008). Using (1) aggregate functions (Getoor et al. 2007) and (2) combining rules (Kersting and De Raedt 2007). Model selection scores have been defined for both aggregators and combining rules. To our knowledge, the consistency of these model selection scores has not been investigated. An open problem with the complete instantiation approach is that the ground inference graph may contain cycles even if the template BN structure does not (Lowd and Domingos 2007).

Previous application of the Learn-and-Join algorithm (Schulte and Khosravi 2012) uses a BN learner for i.i.d. data as a subroutine for learning a first-order BN. While this method upgrades BN learning *algorithms*, it does not upgrade BN objective functions.

Markov Logic Model Selection. Model selection scores have been researched for Markov Logic Networks (MLNs). An MLN can be viewed as a template model for an *un*directed graph. Because computing the partition function is generally intractable, Markov structure learning often optimizes the pseudo log-likelihood (Lowd and Domingos 2007; Kok and Domingos 2005). This is the sum of the log-conditional probabilities of each ground node value, given the values of all other ground nodes. Similar to our normalized log-likelihood scores below, the weighted pseudo log-likelihood (WPLL) normalizes the pseudo log-likelihood counts of different target nodes X_{ij} by the total number γ_i of the groundings of template node X_i. Each weight is penalized with a Gaussian shrinkage prior. The closest counterpart in our experiments is the weighted AIC score below.

Background and Notation

While we introduce no new terminology, in some cases the same concept has been given different names by different communities. In that case we employ the name that is most suggestive for our purposes. We use boldface to denote sets and lists of objects.

Bayesian Networks

A **Bayesian Network (BN) structure** is a directed acyclic graph G (DAG) whose nodes comprise a set of random variables (Pearl 1988). Depending on context, we interchangeably refer to the nodes and (random) variables of a BN. A **Bayesian network** B is a structure G together with a set of parameters. The parameters of a Bayesian network specify the distribution of a child node given an assignment of values to its parent node. For an assignment of values to its nodes, a BN defines the joint probability as the product of the conditional probability of the child node value given its parent values, for each child node in the network, as follows.

$$\ln P_B(\boldsymbol{X} = \boldsymbol{x}) = \sum_{i=1}^{n} \ln P_B(X_i = \boldsymbol{x}_i|\mathbf{Pa}_i^G = \mathbf{pa}_i^G) \quad (1)$$

where $\boldsymbol{x}_i$ resp. $\mathbf{pa}_i^G$ is the assignment of values to node X_i resp. the parents of X_i determined by the assignment $\boldsymbol{x}$.

Example. Figure 1 shows an example of a Bayesian network and associated conditional probabilities.

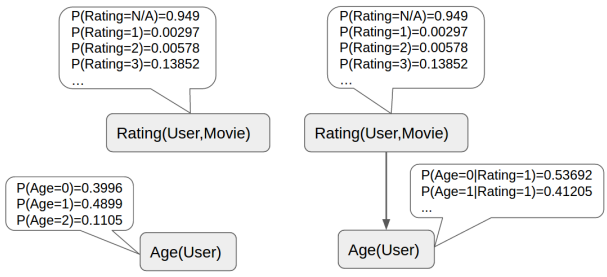

Figure 1: Example Bayesian networks. The type of population variables is shown as in a plate model. Left: A single-node structure G. Right: An expanded structure G_+. Parameter estimates are frequencies computed from the MovieLens database. The rating value is n/a (for "not applicable") if and only if the user has not rated the movie (cf. (Russell and Norvig 2010)) .

User

User_id	Age	Gender
3	0	M
5	1	F
7	2	M
...		

Rating

User_id	Movie_id	Rating
3	The Dictator	1
5	Thor	4
5	The Dictator	3
7	BraveHeart	5
...		

Movie

Movie_id	Action	Drama	Horror
The Dictator	0	0	0
Thor	1	0	0
BraveHeart	1	1	1
...			

Figure 2: An example relational dataset/database. The example follows the closed-world convention: if a relationship between two individuals is not listed, it does not obtain.

Relational Data

We adopt a function-based formalism for combining logical and statistical concepts (Poole 2003; Kimmig, Mihalkova, and Getoor 2014). Table 1 compares statistical concepts for relational and for i.i.d. data.

A multi-relational model is typically a multi-population model. A **population** is a set of individuals of the same type (e.g., a set of $Actors$, a set of $Movies$). Individuals are denoted by lower-case constants (e.g., $brad\_pitt$ and $braveheart$). A k-ary **functor**, denoted f, f' etc., maps a tuple of k individuals to a value. Propositional or i.i.d. data are represented by unary ($k = 1$) functors that return the value of an attribute (column) for each individual (row) from a single population (Nickel et al. 2015). Binary functors can be represented as matrices, $k > 2$-ary functors as tensors of order k. Functors are typed, i.e. their arguments are restricted to appropriate types. Each functor has a set of values (constants) called the **domain** of the functor. Like (Poole 2003), we assume that (1) the domain of all functors is finite, and (2) that functor values are disjoint from individuals.

Figure 2 provides an example of a toy database. A (complete) relational dataset or **database** $\mathcal{D}$, specifies:

1. A finite sample population $\mathcal{I}_1, \mathcal{I}_2 \ldots$, one for each type. Each sample size is denoted by $N\left[\mathcal{I}_i; \mathcal{D}\right]$.
2. The values of each functor, for each input tuple of observed sample individuals of the appropriate type.

First-Order Bayesian Networks

A **population** variable ranges over a population, and is denoted in upper case such as $Actor, Movie, \mathbb{A}, \mathbb{B}$. A (functional) **term** is of the form $f(\tau_1, \ldots, \tau_k)$ where each τ_i is a population variable or an individual of the appropriate type. A term is **ground** if it contains no first-order variables; otherwise it is a **first-order term**. A first-order random variable (FORV) is a first-order term (Wang et al. 2008; Kimmig, Mihalkova, and Getoor 2014). FORV examples are $gender(Actor)$ and $AppearsIn(Actor, Movie)$. When the special syntactic structure of a FORV is not important, we use the traditional random variable notation like X, Y.[1] A FORV can be viewed as a template, instantiated with individual constants, much like an index in a plate model (Kimmig, Mihalkova, and Getoor 2014); see Figure 1. An instantiation or **grounding** for a list of FORVs simultaneously replaces each population variable in the list by a constant. The **number of possible groundings** of a joint assignment is denoted as $N\left[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}\right]$. The **number of satisfying groundings** of a joint assignment in database $\mathcal{D}$ is denoted by $\mathrm{n}\left[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}\right]$. The **database frequency** (Halpern 1990; Schulte 2011) is the number of satisfying instances over the number of possible instances:

$$P_{\mathcal{D}}(\boldsymbol{X} \equiv \boldsymbol{x}) = \frac{\mathrm{n}\left[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}\right]}{N\left[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}\right]}. \tag{2}$$

A first-order Bayesian network (FOB) (Wang et al. 2008), aka Parametrized BN (Kimmig, Mihalkova, and Getoor 2014), is a Bayesian network whose nodes are first-order terms. Following Halpern's well-known random selection semantics for probabilistic logic, a FOB can be viewed as a model of database frequencies (Schulte et al. 2014), that is, as a Statistical-Relational Model (SRM) in the terminology of (Getoor 2001). The basic idea is to view a population variable as a random selection of individuals from its domain. FORVs then denote functions of random variables, hence are themselves random variables. The joint distribution represented in an SRM via Equation (1) can be compared to the database distribution defined in Equation (2) to quantify the data fit of the model.

Relational Model Comparison

We define the relational model comparison scores that we study in this paper. A score $S(G, \mathcal{D})$ measures how well a DAG G fits a database $\mathcal{D}$ (Chickering 2003). A **gain function** $\mathcal{D}(G, G', \mathcal{D})$ measures how much an alternative structure G' improves a current structure G. For every score S, there is an associated gain function defined by $\Delta_S(G, G', \mathcal{D}) = S(G', \mathcal{D}) - S(G, \mathcal{D})$. A score function is **decomposable** if it can be written as a sum of local scores s, each of which is a function only of one node and its parents. Similarly, a gain function is decomposable if the improvement can be written as a sum of local improvements δ. Many structure search algorithms consider adding the addition or deletion of a single edge at a time (Chickering 2003). Let $X_+ \rightarrow X_i$ be an edge not contained in G, and let G_+ be the graph that adds the edge to G. In that case we simplify the notation further by writing the local gain only as a function of the previous parents and the new parent X_+.

We consider upgrading an i.i.d. score of the form (log-likelihood of data under model) - (penalty = function of number of parameters, sample size). We first discuss upgrading the log-likelihood term.

Relational Model Likelihood Scores

We focus on *log-linear likelihood scores*, whose form is similar to the log-linear likelihood of Markov Logic Networks (Schulte 2011). A log-linear likelihood score is a factor product that does not necessarily sum to 1. Log-linearity has the following advantages (Schulte 2011). (1) Generalization: The mathematical form is very close to the i.i.d. log-likelihood function for Bayesian networks. (2) Tractability:

[1] Unfortunately the tradition in statistics clashes with the equally strong tradition in logic of using X, Y for population variables.

Table 1: Statistical concepts for relational vs. i.i.d. data. With a single population and unary functors only, the relational concepts reduce to the i.i.d. concepts.

	Representation	Population	Instances	Sample Size	Empirical Frequency
I.i.d Data	Single Table	Single	Rows	Single N	Sample Frequency
Relational Data	Multiple Tables	Multiple	Groundings	Multiple N, one for each population	Database Frequency

Table 2: Relational Local (Pseudo) Log-likelihood Scores.

Name	Symbol	Definition
Count	$L(X_i, \mathbf{Pa}_i^G, \mathcal{D})$	$\sum_j \sum_k \mathrm{n}_{ijk}^G(\mathcal{D}) \cdot \log_2 \left(\frac{\mathrm{n}_{ijk}^G(\mathcal{D})}{\mathrm{n}_{ij}^G(\mathcal{D})} \right)$
Frequency	$\overline{L}(X_i, \mathbf{Pa}_i^G, \mathcal{D})$	$1/\mathrm{n}_i^G(\mathcal{D}) \times L(X_i, \mathbf{Pa}_i^G, \mathcal{D})$

The score is maximized by the empirical conditional frequencies, as in the i.i.d. case. It can therefore be computed in closed form, given the sufficient statistics in the data. (3) Autocorrelation: The score is well-defined even when the data exhibit cyclic dependencies.

We adopt standard notation for BN sufficient statistics (Heckerman 1998). Let $X_i = x_{ik}, \mathbf{Pa}_i^G = \mathbf{pa}_{ij}^G$ be the assignment that sets node/FORV i to its k-th value, and its parents to their j-th possible configuration. We associate the following concepts with the ijk assignment.

- $\mathrm{n}_{ijk}^G(\mathcal{D}) \equiv \mathrm{n}\left[X_i = x_{ik}, \mathbf{Pa}_i^G = \mathbf{pa}_{ij}^G; \mathcal{D}\right]$ is the number of groundings that satisfy the ijk assignment.
- $\mathrm{n}_{ij}^G(\mathcal{D}) \equiv \sum_k \mathrm{n}_{ijk}^G(\mathcal{D})$ is the number of groundings that satisfy the j-th parent assignment.
- $\mathrm{n}_i^G(\mathcal{D}) \equiv \sum_j \sum_k \mathrm{n}_{ijk}^G(\mathcal{D})$ is the number of possible groundings for node i.

Since the quantity n_i^G plays the same role as the sample size in i.i.d. data, we refer to it as the **local sample size** for node i. A key difference however, is that the *local sample size* n_i^G *depends on both the data and the graph structure* (Lowd and Domingos 2007). In contrast, the global sample size n in i.i.d. data is the same for all nodes and for all graphs.

Table 2 gives the formulas for two previously proposed relational log-likelihood scores that we consider in this paper. The **count log-likelihood** has the same form as the local log-likelihood for i.i.d. data, but replaces counts in a single data table by counts in a database. The **frequency likelihood** (Schulte 2011) normalizes the count score by the local sample size (cf. (Xiang and Neville 2011)). This is equivalent to replacing counts in a single data table by frequencies in a database. Table 3 illustrates the likelihood computations.

For i.i.d. data, and the frequency log-likelihood, adding an edge to a graph G can only increase the log-likelihood score. A big difference is that *adding an edge can decrease the relational count log-likelihood.* This occurs when the new edge **adds a population variable** that was not contained in the child node or the previous parents; see Figure 1. As Table 3 illustrates, adding the edge $Rating(User, Movie) \rightarrow Age(User)$ changes the local sample size from the Users population ($n_i = 941$) to the Users $\times$ Movies population ($n_i = 1,582,762$). The count log-likelihood L therefore decreases simply because the scale of the counts changes.

Family Configuration	n_{ijk}	n_{ij}	n_i	n_{ijk}/n_i	CP	$L = n_{ijk} \cdot log_2(CP)$	$\overline{L} = L/n_i$
Age(User)=0	376	—	941	0.3996	0.3996	-497.6217	-0.5288
Age(User)=0, Rating(User,Movie)=1	2524	4703	1582762	0.0016	0.5367	-2266.2224	-0.0014

Table 3: An edge $Rating(User, Movie) \rightarrow Age(User)$ expands a structure G to a larger structure G_+. The CP-parameter values are maximum likelihood estimates computed from the Movielens database. The L and $\overline{L}$ columns show the contributions of the family configuration in each row (part of the sum for the total likelihood values).

The Normalized Gain Score

Here and below, we write G_+ for the DAG that results from adding a generic edge $X_+ \rightarrow X_i$ to DAG G. Our upgrade method for defining a relational gain function is as follows. (1) Compute the likelihood differential using the frequency likelihood $\overline{L}$. (2) Normalize the penalty term differential by the *larger* sample size $\mathrm{n}_i^+(\mathcal{D})$. (3) The **normalized gain** is computed as (1) minus (2), likelihood differential minus penalty differential. Table 4 gives the relational gain formulas for upgrading the standard AIC and BIC scores (Bouckaert 1995). The normalized gain can be applied with other i.i.d. scores as well.

Balance

One of the fundamental properties of standard model selection scores is that likelihood scores and the penalty terms are standardized to the same scale (Russell and Norvig 2010, Sec.18.4.3). For example, in a Bayesian view, both terms are log-probabilities (Chickering and Meek 2002), and in the MDL view, both can be viewed as being measured in bits (Russell and Norvig 2010, Sec.18.4.3). The normalized gain is balanced because the likelihood scores and the penalty terms are standardized to the same scale. Therefore any balanced gain function should be a rescaling of the normalized gain. Mathematically, a rescaling can be represented as a positive linear transformation $[\alpha\,(\text{normalized gain}) + c]$ where $\alpha > 0$ may depend only on the local sample sizes of the structures compared. The next proposition shows that no balanced gain function can be represented as the difference of a single score.

Local Gain Function	Definition
$\Delta\overline{L}(X_i, \mathbf{Pa}_i^G, X_+, \mathcal{D})$	$\overline{L}(X_i, \mathbf{Pa}_i^G \cup X_+, \mathcal{D}) - \overline{L}(X_i, \mathbf{Pa}_i^G, \mathcal{D})$
$\Delta\overline{AIC}(X_i, \mathbf{Pa}_i^G, X_+, \mathcal{D})$	$\Delta\overline{L}(X_i, \mathbf{Pa}_i^G, X_+, \mathcal{D}) - \frac{\Delta pars(X_i, \mathbf{Pa}_i^G, X_+)}{\mathrm{n}_i^+(\mathcal{D})}$
$\Delta\overline{BIC}(X_i, \mathbf{Pa}_i^G, X_+, \mathcal{D})$	$\Delta\overline{L}(X_i, \mathbf{Pa}_i^G, X_+, \mathcal{D}) - \frac{\frac{1}{2}\log_2(\mathrm{n}_i^+(\mathcal{D}))\Delta pars(X_i, \mathbf{Pa}_i^G, X_+)}{\mathrm{n}_i^+(\mathcal{D})}$

Table 4: Our Proposed Normalized Relational Model Gain Upgrade for AIC and BIC. $\mathrm{n}_i^+(\mathcal{D}) \equiv \mathrm{n}_i^{G_+}(\mathcal{D})$ denotes the local sample size for the expanded graph. $\Delta pars(X_i, \mathbf{Pa}_i^G, X_+) = \#pars(X_i, \mathbf{Pa}_i^{G'}) - \#pars(X_i, \mathbf{Pa}_i^G)$

Likelihood \ Penalty	Count	Normalized
Count	S; not consistent; underfits	—
Normalized	$\widetilde{S}$; not consistent; underfits	$\overline{S}$; not consistent; overfits

Table 5: Rescaling the likelihood count and/or the penalty term defines relational score upgrades.

Proposition 1 *There is no relational model selection score such that its associated model gain function is balanced (i.e., a unit change of the normalized gain).*

The imbalance of log-linear single-model scores leads to non-consistency, as the examples and theoretical analysis of the next section illustrate.

Comparison Multi-Relational Model Scores

The baseline methods for our comparison define relational scores by normalizing the likelihood term and/or the penalty term. This leads to potentially 4 different relational versions of a model selection score; see Table 5. Normalizing the penalty term but not the likelihood term is clearly inadequate. Table 6 gives the formulas for the remaining 3 different relational versions of AIC and BIC.

Consistency Analysis

This section shows that the normalized gain function is relatively consistent, but the model score functions are not. The reason is that the former are balanced and the latter are not. We observe that (*) *for edges that do not add population variables, the normalized gain and model scores are equivalent*, except for the Normalized-Count Score.

A large sample analysis considers a sequence of samples that grow without bound and converge to the true data generating distribution. Following previous work on consistency for relational data (Sakai and Yamanishi 2013; Xiang and Neville 2011; Shalizi and Rinaldo 2013), we formalize this concept as follows. In the limit the sample size for each population $\mathcal{I}_i$ goes to infinity, $N[\mathcal{I}_i; \mathcal{D}_j] \to \infty$, which like (Sakai and Yamanishi 2013) we abbreviate as $\boldsymbol{N}(\mathcal{D}) \to \infty$. Like (Xiang and Neville 2011), we make the identifiability assumption that

$$P_{\mathcal{D}}(\cdot) \to P_w(\cdot) \equiv p \text{ as } \boldsymbol{N}(\mathcal{D}) \to \infty.$$

Here w represents a complete relational structure (network) from which samples are drawn. We abbreviate the frequency distribution associated with the complete structure as p, for brevity and compatibility with (Chickering 2003) where p is used for the data generating distribution. Arbitrarily large samples can be generated either by sampling with replacement, or by sampling from infinite populations. For discussion of network sampling, see (Shalizi and Rinaldo 2013; Frank 1977). Chickering and Meek analysed BN model selection scores in terms of **local consistency**, which we adapt for score gain functions as follows.

Definition 1 *Let p be the data generating distribution. A local gain function is **locally consistent** if, the following hold in the sample size limit as $\boldsymbol{N}(\mathcal{D}) \to \infty$, for any single-edge expansion G_+:*

1. *If X_+ is not independent of X_i given $\mathbf{Pa}_i^G$ in p, then $\Delta(G, G_+, \mathcal{D}) > 0$.*
2. *If X_+ is independent of X_i given $\mathbf{Pa}_i^G$ in p, then $\Delta(G, G_+, \mathcal{D}) < 0$.*

An upgrade method is relatively locally consistent if the upgraded i.i.d. score/gain is locally consistent for relational data whenever it is locally consistent for i.i.d. data. We omit the proof of the next theorem due to space constraints.

Theorem 1 *The normalized gain function (Section) is relatively locally consistent. None of the relational model scores (Table 6) are relatively locally consistent.*

The intuitive reasons for non-consistency are as follows.

Count-Count Score S The count likelihood typically decreases for an edge that adds a population variable, even if the edge represents a true dependency.

Normalized-Count $\widetilde{S}$ The weight of the likelihood term does not increase with sample size, so an edge that represents a true dependency may not be added even in the sample size limit.

Normalized-Normalized $\overline{S}$ The normalized penalty term for the expanded structure G_+ is down-weighted more than the normalized penalty term for the simpler structure G. So a redundant edge may be added even in the sample size limit.

Evaluation

We describe the system and the datasets we used. Code was written in Java, JRE 1.7.0. and executed with 8GB of RAM and a single Intel Core 2 QUAD Processor Q6700 with a clock speed of 2.66GHz (no hyper-threading). The operating system was Linux Centos 2.6.32. The MySQL Server version 5.5.34 was run with 8GB of RAM and a single core

Score	AIC_i	BIC_i
Count	$AIC(X_i, \mathbf{Pa}_i^G, \mathcal{D}) \equiv L(X_i, \mathbf{Pa}_i^G, \mathcal{D}) - \#pars(X_i, \mathbf{Pa}_i^G)$	$BIC(X_i, \mathbf{Pa}_i^G, \mathcal{D}) \equiv L(X_i, \mathbf{Pa}_i^G, \mathcal{D}) - \frac{1}{2}\log_2(\mathrm{n}_i^G(\mathcal{D})) \cdot \#pars(X_i, \mathbf{Pa}_i^G)$
Normalized	$\overline{AIC}(X_i, \mathbf{Pa}_i^G, \mathcal{D}) \equiv \frac{1}{\mathrm{n}_i^G(\mathcal{D})} AIC(X_i, \mathbf{Pa}_i^G, \mathcal{D})$	$\overline{BIC}(X_i, \mathbf{Pa}_i^G, \mathcal{D}) \equiv \frac{1}{\mathrm{n}_i^G(\mathcal{D})} BIC(X_i, \mathbf{Pa}_i^G, \mathcal{D})$
Weighted	$\widetilde{AIC}(X_i, \mathbf{Pa}_i^G, \mathcal{D}) \equiv \overline{L}(X_i, \mathbf{Pa}_i^G, \mathcal{D}) - \#pars(X_i, \mathbf{Pa}_i^G)$	$\widetilde{BIC}(X_i, \mathbf{Pa}_i^G, \mathcal{D}) \equiv \overline{L}(X_i, \mathbf{Pa}_i^G, \mathcal{D}) - \frac{1}{2}\log_2(\mathrm{n}_i^G(\mathcal{D})) \cdot \#pars(X_i, \mathbf{Pa}_i^G)$

Table 6: Relational Local Model Selection Scores, count and frequency versions. Normalized scores divide count scores by the local sample size $\mathrm{n}_i^G(\mathcal{D})$.

Dataset	**#Relationship Tables/ Total**	**#Tuples**	**#Attributes**
University	2	171	12
Movielens	1 / 3	1,010,051	7
Mutagenesis	2 / 4	14,540	11
Financial	3 / 7	225,932	15
Hepatitis	3 / 7	12,927	19
IMDB	3 / 7	1,354,134	17

Table 7: Datasets characteristics. #Tuples = total number of tuples over all tables in the dataset.

processor of 2.2GHz. All code and data is available on-line (Khosravi et al.)

Datasets. We used seven benchmark real-world databases. For detailed descriptions and the sources of the databases, please see references (Schulte and Khosravi 2012; Qian, Schulte, and Sun 2014). Table 7 summarizes basic information about the benchmark datasets. IMDB is the largest dataset in terms of number of total tuples (more than 1.3M tuples) and schema complexity.

Structure Learning Algorithm. We used the previously existing learn-and-join method (LAJ), which is the state of the art for Bayes net learning in relational databases (Schulte and Khosravi 2012; Qian, Schulte, and Sun 2014). We used the LAJ implementation provided by its creators. The LAJ method conducts a search through the lattice of relational paths. At each lattice point, an i.i.d. Bayesian network learner is applied, and learned edges are propagated from shorter paths to longer paths. We reconfigured the LAJ algorithm by changing the `score` class.

Results. For each learned graph G, we use maximum likelihood estimates to obtain a Bayesian network B to be evaluated. To measure how close the joint distribution represented by a learned BN is to the database distribution, we employ the standard Kulback-Leibler divergence metric (KLD) (de Campos 2006) . Figure 3 shows the KLD and parameter results for upgrading AIC resp. BIC. The normalized-count scores select very sparse structures, and the normalized-normalized scores very dense structures. The many edges found by a normalized-normalized score lead to almost no improvement in log-likelihood compared to the normalized gain function. Given their theoretical shortcomings as well, we conclude that *the normalized-count and normalized-normalized scores are clearly inadequate*.

We observed that *the count-count scores never select edges that add population variables*. In contrast, the normalized-gain scores do select all types of edges, as shown in Figure 4. From Figure 3, the impact of the edges with new population variables appears to be mixed. On MovieLens,

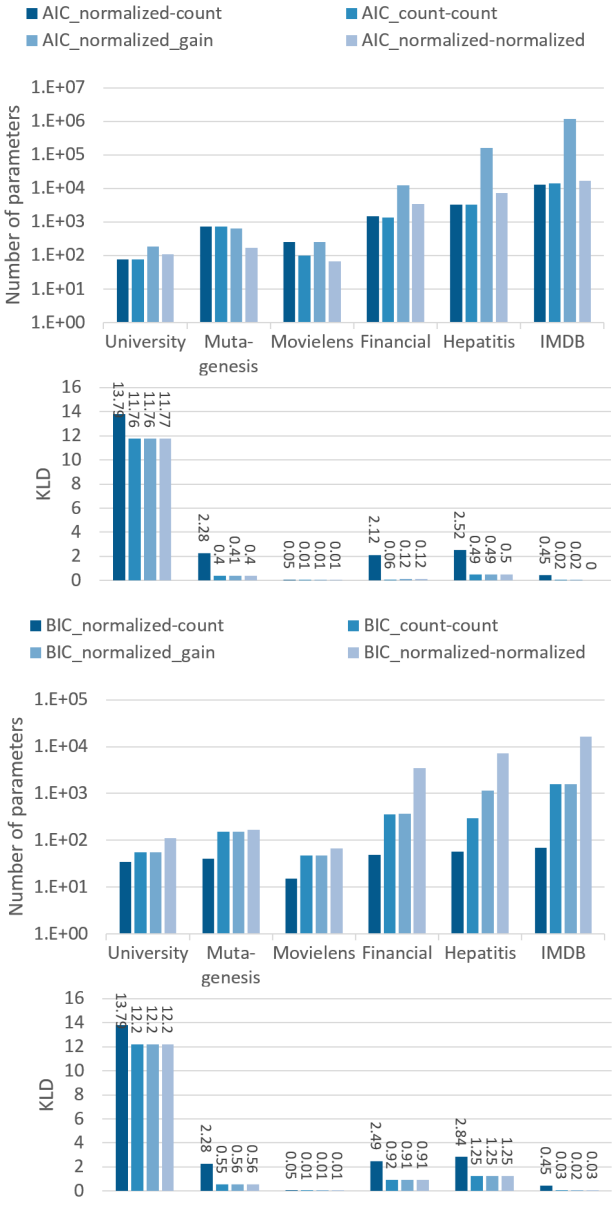

Figure 3: Number of Parameters for different relational score upgrade methods; Kullback-Leibler divergence between the Bayesian network and the database distribution. The number of parameters is shown on log-scale. Top: AIC upgrades. Bottom: BIC upgrades.

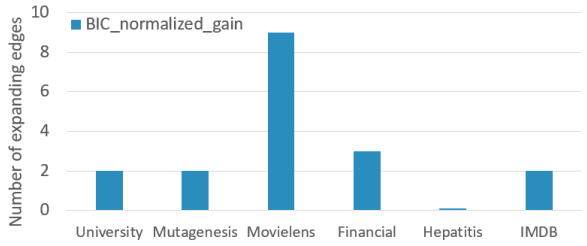

Figure 4: The number of edges that add population variables, for the normalized gain upgrade method.

they improve the log-likelihood score, whereas on IMDB and Hepatitis, the log-likelihood score is worse.

Conclusions The experimental results support the normalized gain as the best upgrade method. The normalized-normalized and normalized-count upgrade methods are clearly inadequate. The count-count method is incapable of selecting edges that add population variables. Inspection suggests that such edges can be informative. While the quantitative statistical evidence for their importance is mixed, such a strong a priori bias against certain types of edges is clearly undesirable. In addition, the normalized gain is the only upgrade method in our set that has the theoretical properties of balance and (relative) consistency.

References

Bouckaert, R. 1995. *Bayesian belief networks: from construction to inference*. Ph.D. Dissertation, U. Utrecht.

Chickering, D. M., and Meek, C. 2002. Finding optimal Bayesian networks. In *UAI*, 94–102.

Chickering, D. 2003. Optimal structure identification with greedy search. *Journal of Machine Learning Research* 3:507–554.

de Campos, L. 2006. A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *JMLR* 2149–2187.

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

Getoor, L., and Taskar, B. 2007. *Introduction to Statistical Relational Learning*. MIT Press.

Getoor, L.; Friedman, N.; Koller, D.; Pfeffer, A.; and Taskar, B. 2007. Probabilistic relational models. In *Introduction to Statistical Relational Learning* (2007). chapter 5, 129–173.

Getoor, L. 2001. *Learning Statistical Models From Relational Data*. Ph.D. Dissertation, Department of Computer Science, Stanford University.

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

Heckerman, D. 1998. A tutorial on learning with Bayesian networks. In *NATO ASI on Learning in graphical models*, 301–354.

Kersting, K., and De Raedt, L. 2007. Bayesian logic programming: Theory and tool. In *Introduction to Statistical Relational Learning* (2007). chapter 10, 291–318.

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

Kimmig, A.; Mihalkova, L.; and Getoor, L. 2014. Lifted graphical models: a survey. *Machine Learning* 1–45.

Knobbe, A. J. 2006. *Multi-relational data mining*, volume 145. Ios Press.

Kok, S., and Domingos, P. 2005. Learning the structure of Markov logic networks. In Raedt, L. D., and Wrobel, S., eds., *ICML*, 441–448. ACM.

Laer, W. V., and de Raedt, L. 2001. How to upgrade propositional learners to first-order logic: A case study. In *Relational Data Mining*. Springer Verlag.

Lowd, D., and Domingos, P. 2007. Efficient weight learning for Markov logic networks. In *PKDD*, 200–211.

Natarajan, S.; Tadepalli, P.; Dietterich, T. G.; and Fern, A. 2008. Learning first-order probabilistic models with combining rules. *Annals of Mathematics and Artifical Intelligence* 54(1-3):223–256.

Neville, J., and Jensen, D. 2007. Relational dependency networks. *Journal of Machine Learning Research* 8:653–692.

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2015. A Review of Relational Machine Learning for Knowledge Graphs.

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

Poole, D. 2003. First-order probabilistic inference. In *IJCAI*.

Qian, Z.; Schulte, O.; and Sun, Y. 2014. Computing multi-relational sufficient statistics for large databases. In *CIKM*, 1249–1258.

Russell, S., and Norvig, P. 2010. *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Sakai, Y., and Yamanishi, K. 2013. An nml-based model selection criterion for general relational data modeling. In *Big Data*, 421–429. IEEE.

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

Schulte, O.; Khosravi, H.; Kirkpatrick, A.; Gao, T.; and Zhu, Y. 2014. Modelling relational statistics with bayes nets. *Machine Learning* 94:105–125.

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

Shalizi, C. R., and Rinaldo, A. 2013. Consistency under sampling of exponential random graph models. *Annals of statistics* 41(2):508.

Sutton, C., and McCallum, A. 2007. An introduction to conditional random fields for relational learning. In *Introduction to Statistical Relational Learning* (2007). chapter 4, 93–127.

Wang, D. Z.; Michelakis, E.; Garofalakis, M.; and Hellerstein, J. M. 2008. Bayesstore: managing large, uncertain data repositories with probabilistic graphical models. volume 1, 340–351. VLDB Endowment.

Williams, D. 2001. *Weighing the Odds*. Cambridge University Press.

Xiang, R., and Neville, J. 2011. Relational learning with one network: An asymptotic analysis. In *Artificial Intelligence and Statistics*, 779–788.