Locally Consistent Bayesian Network Scores for Multi-Relational Data*

Oliver Schulte and Sajjad Gholami
Simon Fraser University, Burnaby, Canada
{oschulte,sgholami}@cs.sfu.ca

Abstract

An important task for relational learning is Bayesian network (BN) structure learning. A fundamental component of structure learning is a model selection score that measures how well a model fits a dataset. We describe a new method that upgrades for multi-relational databases, a log-linear BN score designed for single-table i.i.d. data. Chickering and Meek showed that for i.i.d. data, standard BN scores are locally consistent, meaning that their maxima converge to an optimal model, that represents the data generating distribution *and* contains no redundant edges. Our main theorem establishes that if a model selection score is locally consistent for i.i.d. data, then our upgraded gain function is locally consistent for relational data as well. To our knowledge this is the first consistency result for relational structure learning. A novel aspect of our approach is employing a *gain function* that compares two models: a current vs. an alternative BN structure. In contrast, previous approaches employed a score that is a function of a single model only. Empirical evaluation on six benchmark relational databases shows that our gain function is also practically useful: On realistic size data sets, it selects informative BN structures with a better data fit than those selected by baseline single-model scores.

1 Introduction

Many organizations maintain their data in a multi-relational database. I.i.d. data can be viewed as a special limiting case of multi-relational data with no relationships [Nickel *et al.*, 2016]. Statistical-relational learning (SRL) aims to generalize i.i.d. machine learning methods for multi-relational data; this is called *upgrading* the method [Getoor and Taskar, 2007; Laer and de Raedt, 2001]. Statistical-relational models have achieved state-of-the-art performance in a number of application domains, such as ontology matching, information extraction, entity resolution, link-based clustering, query optimization, representing uncertainty in databases, etc [Domingos and Richardson, 2007; Niu *et al.*, 2011; Getoor *et al.*, 2001a]. This paper addresses the important SRL task of learning a Bayesian network (BN) structure from a relational dataset.

The most common approach to BN structure learning is to search for a structure that optimizes a model selection score for a given dataset. We propose a general method for upgrading BN model selection scores. Our method can be applied with any of the standard BN scores, such as AIC, BIC, BDeu, MDL etc. [Bouckaert, 1995]. Its main theoretical property is *preserving local consistency* [Chickering and Meek, 2002]: If the i.i.d. model criterion is locally consistent for i.i.d. data, the upgraded criterion is locally consistent for multi-relational data. Local consistency combines (i) consistency: as the amount of available data increases, the model selection criterion selects a graph that can represent the data generating distribution, and (ii) optimality: the graph contains no edges that are redundant for representing the data generating distribution. While our theorem generalizes the classic i.i.d. results [Chickering and Meek, 2002], a major point of departure is that we employ a *gain function* that compares a current vs. an alternative BN structure, rather than a single-model score. The gain function transforms the sufficient statistics for compared structures to the same scale.

Our experiments indicate that the gain function in practice strikes a desirable balance between selecting overly dense and overly sparse structures. In contrast, for baseline scores that are a function of a single model only, the scores either under-weight or over-weight model complexity, selecting either overly dense or overly sparse structures.

Contributions. Our main contributions may be summarized as follows.

1. A novel method for upgrading an i.i.d. BN structure score to relational databases, based on a gain function that compares the data fit of two graph structures.
2. Preserving local consistency proof: if a 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 relational structure learning.

Paper Organization. We review background on Bayesian networks and relational data. Then we define our gain function method for upgrading model selection scores, as well as

*Supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada. We thank Mark Schmidt and Leonid Chindelevitch for helpful suggestions.

baseline upgrade methods for comparison. Theoretical analysis demonstrates that the gain function method preserves local consistency, whereas the baseline single-model scores do not. Empirical evaluation on six benchmark data sets compares the BN structures selected by the gain function to those selected by the baseline scores, with respect to data fit and model complexity.

2 Related Work

Relational Consistency. There have been several recent studies of the consistency of relational learning. Sakai and Yamanishi (2013) provide an asymptotic analysis of selecting the number of relational clusters by optimizing minimum description length. For BN parameter learning, Schulte (2011) upgraded the i.i.d. log-likelihood score by *normalizing*, which converts feature counts to proportions. Xiang and Neville (2011) prove that the normalized log-likelihood (NLL) is consistent for *parameter* learning in Markov Logic Networks. We use their framework of learning from one network, to investigate consistency for Bayesian network *structure* learning. Our gain function extends the NLL score with a normalized model complexity penalty term. The weighted pseudo log-likelihood score for Markov Logic networks [Lowd and Domingos, 2007], also normalizes the log-likelihood term, but not the penalty term, and is non-consistent for the same reason as the count score defined below.

Consistency and Frequencies. A BN structure G can be parametrized to represent a distribution p if and only if G is an I-map of p, meaning that every d-separation in G corresponds to conditional independence in p [Pearl, 1988]. The blueprint for a consistency argument in the i.i.d. setting is that as the sample size increases, the empirical frequencies approach the data generating distribution p, and the score approaches the maximum likelihood score, and therefore selects an I-map of p. The most straightforward way to generalize this blueprint is to view a multi-relational BN structure as a model of database frequencies, rather than a template model [Getoor, 2001; Schulte *et al.*, 2014]. Using Getoor's terminology, we consider a Statistical-Relational Model (SRM) rather than a Probabilistic-Relational Model (PRM).

Relational Template Models. Many SRL models employ a log-linear likelihood function [Kimmig *et al.*, 2014]; our upgrade method generalizes to any such model. A common approach for defining relational likelihood functions with directed graphical models is to aggregate the information from the multiple parent instances of a ground node using aggregate functions [Getoor *et al.*, 2001b] or combining rules [Poole, 2003]. Recent representation results [Buchman and Poole, 2015] show that such aggregators can be represented in a log-linear model that introduces complex functions (e.g. the number of action movies rated by a user). Since our upgrade method is defined for complex functions, it can in principle be applied to aggregate functions and combining rules. A direct empirical evaluation is currently not possible as there is no implementation of relational BN structure learning with complex functions.

The Inductive Logic Programming FOIL system [Quinlan and Cameron-Jones, 1993] defined the information gain that results from adding a new condition (literal) to a first-order rule. The FOIL information gain is similar to our approach in that 1) it defines a gain function rather than a score, and 2) the key issue concerns adding population variables. It is different in that 1) it is applied with a discriminative not generative model, and 2) different rule groundings are combined using existential quantification rather than a log-linear model.

Previous application of the Learn-and-Join search strategy [Schulte and Khosravi, 2012] used a BN learner for i.i.d. data as a subroutine for learning a multi-relational BN. LAJ search upgrades a BN learning algorithm, but does not define an objective function for model optimization.

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 1: Excerpt from a relational dataset/database.

3 Background and Notation

We adopt a function-based formalism for combining relational and statistical concepts [Poole, 2003; Russell, 2015]. For a set of random variables $\boldsymbol{X} = \{X_1, \ldots, X_n\}$, the notation $P(\boldsymbol{X} = \boldsymbol{x}) \equiv P(\boldsymbol{x})$ denotes the joint probability that each random variable X_i takes on value $\boldsymbol{x}_i$.

Relational 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 $Users$, a set of $Movies$). Individuals are denoted by constants (e.g., $user_3$ and $thor$). A k-ary **functor**, denoted f, f' etc., maps a tuple of k individuals to a value. The arguments of a functor are restricted to appropriate types. The possible values of a functor form the **domain** of the functor. Like [Poole, 2003], we assume that (1) the domain of each functor is finite, and (2) functor values are disjoint from individuals. Throughout the paper we assume complete data. 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.
2. The values of each functor, for each input tuple of observed sample individuals of the appropriate type.

Figure 1 shows a toy database. The example follows the closed-world convention: if a relationship between two individuals is not listed, it does not obtain.

Relational Random Variables A **population** variable ranges over a population, and is denoted in upper case such as $User, Movie, \mathbb{A}$. A **term** is of the form $f(\tau_1, \ldots, \tau_k)$ where each τ_i is a population variable or a constant/individual of the appropriate type. A term is **ground** if it contains only constants; otherwise it is a **first-order term** with at

least one population variable. A first-order random variable (FORV) is a first-order term [Wang *et al.*, 2008]. FORV examples are $age(User)$,$rating(User, Movie)$. We use traditional random variable notation like X, Y for FORVs.[1] A FORV can be instantiated with individual constants, much like an index in a plate model [Kimmig *et al.*, 2014]. A **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 given by $N[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}] \equiv N[\mathbb{A}_1; \mathcal{D}] \times \cdots \times N[\mathbb{A}_m; \mathcal{D}]$ where the $\mathbb{A}_i$ are the population variables in $\boldsymbol{X}$ and $N[\mathbb{A}; \mathcal{D}]$ is the size of the sample population of $\mathbb{A}$. The **number of satisfying groundings** of a joint assignment in database $\mathcal{D}$ is denoted by $\mathrm{n}[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}]$. The **database frequency** [Halpern, 1990] is the number of satisfying groundings over the number of possible groundings:

$$P_{\mathcal{D}}(\boldsymbol{X} = \boldsymbol{x}) = \frac{\mathrm{n}[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}]}{N[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}]}. \quad (1)$$

First-Order Bayesian Networks A **Bayesian Network (BN) structure** is a directed acyclic graph G (DAG) whose nodes comprise a set of random variables [Pearl, 1988]. A **Bayesian network** B is a structure G together with a set of parameter values, which 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 via the standard product formula:

$$P_B(\boldsymbol{X} = \boldsymbol{x}) = \prod_{i=1}^{n} P_B(X_i = \boldsymbol{x}_i | \mathrm{Pa}_i^G = \mathbf{pa}_i^G) \quad (2)$$

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 $\boldsymbol{x}$.

A first-order Bayesian network (FOB) [Wang *et al.*, 2008], aka Parametrized BN [Kimmig *et al.*, 2014], is a BN whose nodes are first-order terms. Via Equation (2), a FOB defines a joint distribution over FORVs, so a FOB can be viewed as a Statistical-Relational Model (SRM) of database frequencies [Getoor, 2001]. The semantics of first-order probability logic provides a frequency semantics for FOBs, where a population variable represents an independent random selection from its population [Halpern, 1990; Schulte *et al.*, 2014]. The basis of a model fit score is comparing the joint distribution $P_B(\cdot)$ from Equation (2) to the empirical database distribution $P_{\mathcal{D}}(\cdot)$ from Equation (1).

Examples Figure 2 shows an example of two small FOBs. 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]). *Throughout the paper, conditional probability estimates are computed from the IMDb database described below.* Table 1 illustrates database frequencies using the IMDb dataset. The number of users is 941, of which 376 are at age level 0, so the frequency of age 0 users is 376/941. The number of user-movie pairs is 1,582,762 of which 2,524 have the user at age level 0 and a rating of 1. Marginal and joint BN probabilities are computed using Equation (2). The expanded BN B_1^+ matches the database distribution perfectly but at the cost of more parameters.

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

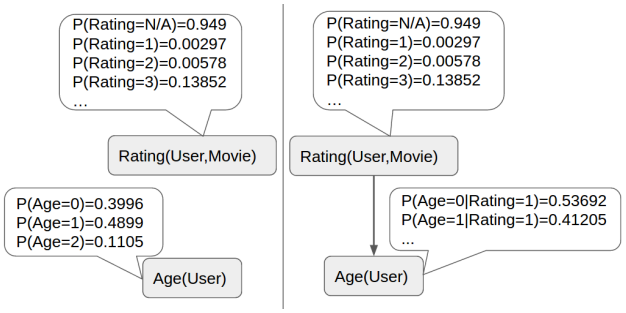

Figure 2: Example First-Order Bayesian networks: left = B_1 with graph G_1, right = B_1^+ with graph G_1^+.

Table 1: The IMDb database frequency of a joint assignment to first-order random variables, compared to the BN probabilities computed using the network parameters of Figure 2.

$\boldsymbol{X} = \boldsymbol{x}$	$Age(User) = 0$	$Age(User) = 0$, $Rating(User, Movie) = 1$
$\mathrm{n}[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}]$	376	2,524
$N[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}]$	941	1,582,762
$P_{\mathcal{D}}(\boldsymbol{X} = \boldsymbol{x})$	$376/941 \approx 0.3996$	$2,524/1,582,762 \approx 0.0016$
$P_{B_1}(\boldsymbol{X} = \boldsymbol{x})$	0.3996	$0.00297 \cdot 0.3996 \approx 0.0012$
$P_{B_1^+}(\boldsymbol{X} = \boldsymbol{x})$	0.3996	$0.00297 \cdot 0.53692 \approx 0.0016$

4 Multi-Relational Model Comparison

An i.i.d. score measures how well a DAG G fits an i.i.d. dataset D [Chickering and Meek, 2002]. A BN score defines a function $S(G, \mathbf{n}_{ijk}^G(D))$ that depends on the graph structure and the **sufficient statistics** $\mathbf{n}_{ijk}^G(D)$. For Bayesian networks, the sufficient statistics are the observed instantiation counts of the possible child-parent configurations. Let $X_i = x_{ik}, \mathrm{Pa}_i^G = \mathbf{pa}_{ij}^G$ be the assignment that sets node i to its k-th value, and its parents to their j-th possible configuration. Then $\mathrm{n}_{ijk}^G(D) \equiv \mathrm{n}\left[X_i = x_{ik}, \mathrm{Pa}_i^G = \mathbf{pa}_{ij}^G; D\right]$ is the number of data points that satisfy the ijk assignment. A standard BN score is *decomposable*, that is, the score can be written as a sum of **local scores** S_i, each of which is a function only of one node X_i and its parents:

$$S(G, \mathbf{n}_{ijk}^G(D)) := \sum_i S_i(G, \mathbf{n}_{ijk}^G(D)). \quad (3)$$

We use the following notation for *relational* sufficient statistics.

- $\mathrm{n}_{ijk}^G(\mathcal{D}) \equiv \mathrm{n}\left[X_i = x_{ik}, \mathrm{Pa}_i^G = \mathbf{pa}_{ij}^G; \mathcal{D}\right]$ 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.

Algorithm 1: The normalized gain method upgrades a decomposable i.i.d. BN score S for multi-relational data.

Input: Database $\mathcal{D}$; Bayesian network DAGs G, G^+ where $\mathrm{Pa}_i^G \subseteq \mathrm{Pa}_i^{G^+}$ for each node X_i.
Output: Gain value $\Delta\overline{S}(G, G^+, D)$
Calls local i.i.d. score $S_i(G, \mathbf{n}_{ijk})$. (Eq. (3))

1: $\overline{\mathbf{n}}_{ijk}^G(\mathcal{D}) := \mathbf{n}_{ijk}^G(\mathcal{D}) \times \frac{\mathrm{n}_i^{G^+}(\mathcal{D})}{\mathrm{n}_i^G(\mathcal{D})}$ {rescale sufficient statistics for graph G}
2: **for all** nodes i **do**
3: $\quad\Delta\overline{S}_i(G, G^+, \mathcal{D}) := [S_i(G^+, \mathbf{n}_{ijk}^{G^+}(\mathcal{D})) - S_i(G, \overline{\mathbf{n}}_{ijk}^G(\mathcal{D}))]/\mathrm{n}_i^{G^+}(\mathcal{D})$ {gain = [score of G^+-scaled score of G]/local sample size}
4: **end for**
5: **return** $\sum_i \Delta\overline{S}_i(G, G^+, \mathcal{D})$

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.

We propose a relational **gain function** $\Delta S(G, G', \mathcal{D})$ that measures how much an alternative structure G' improves a current structure G according to criterion S. Our definition focuses on the case where the alternative G' adds parents to a node X_i in G. The case where G' removes parents reverses the role of G and G'. This is sufficient for applying standard BN structure search algorithms, which consider adding or deleting a single edge at a time, or distinct phases for adding and deleting edges. The gain for edge reversals adds the gains for a deletion and addition. Algorithm 1 shows pseudo code for the gain function. Table 2 gives the normalized gain penalty formulas for upgrading the standard log-likelihood, AIC, and BIC scores [Bouckaert, 1995]. Algorithm 1 can be applied with other scores as well (e.g. for $BDeu$ the normalized gain formula is given in [Gholami, 2016, Section 3.1.3]). We focus on AIC and BIC because they are widely used and have a relatively simple definition.

Motivation Rescaling sufficient statistics for the current graph G (line 1) makes comparable the scores of the current graph and the alternative graph G^+. Normalization (line 3) makes comparable the gains for different alternative graphs. The normalization measures the gain per local instance.

Table 3 illustrates the importance of re-scaling counts. The $LL_i(\cdot, \mathbf{n}_{ijk}^{\cdot}(\cdot))$ column shows the likelihood score with instantiation counts. This term is an order of magnitude lower for the expanded BN structure G_1^+ (-2266 vs. -497), simply because the expanded structure increases the local sample size by the number of Movies.

Relationship to Normalized Likelihood In previous work on parameter learning, [Xiang and Neville, 2011; Schulte, 2011], the log-likelihood score LL was upgraded by the **normalized log-likelihood score** NLL

$$\overline{LL}_i(G, \mathbf{n}_{ijk}^G(\mathcal{D})) \equiv LL_i(G, \mathbf{n}_{ijk}^G(\mathcal{D}))/\mathrm{n}_i^G(\mathcal{D}),$$

which converts log-likelihood scores to the same scale, as shown in Table 3. The normalized gain for the log-likelihood score is equivalent to the normalized log-likelihood score differential.

Observation 4.1 *The normalized gain equals the difference in normalized log-likelihood. In symbols:*

$$\Delta\overline{LL}_i(G, G^+, \mathcal{D}) = \overline{LL}_i(G^+, \mathbf{n}_{ijk}^{G^+}(\mathcal{D})) - \overline{LL}_i(G, \mathbf{n}_{ijk}^G(\mathcal{D})).$$

Proof. It suffices to show that $LL_i(G, \overline{\mathbf{n}}_{ijk}^G(\mathcal{D}))/\mathrm{n}_i^{G^+}(\mathcal{D}) = LL_i(G, \mathbf{n}_{ijk}^G(\mathcal{D}))/\mathrm{n}_i^G(\mathcal{D})$. In the scaled log-likelihood $LL_i(G, \overline{\mathbf{n}}_{ijk}^G(\mathcal{D}))$, the scale factor $\frac{\mathrm{n}_i^{G^+}(\mathcal{D})}{\mathrm{n}_i^G(\mathcal{D})}$ does not affect the conditional probability ratio, and can be moved to the front of the sum. Therefore

$$LL_i(G, \overline{\mathbf{n}}_{ijk}^G(\mathcal{D})) = \frac{\mathrm{n}_i^{G^+}(\mathcal{D})}{\mathrm{n}_i^G(\mathcal{D})} LL_i(G, \mathbf{n}_{ijk}^G(\mathcal{D})).$$

Many standard BN scores, such as AIC and BIC, are **likelihood scores** that combine the maximum likelihood of the data under the model with a *penalty* term $f^S(\#pars_i^G, \mathbf{n}_{ijk}^G(D))$ that is a function of the number of parameters and the sample size [Bouckaert, 1995]. Observation 4.1 implies that the normalized gain for likelihood scores is equivalent to adding a normalized penalty term to the normalized likelihood. Whereas the normalized likelihood gain can be represented as the difference of two fixed single-model scores, this is no longer true for likelihood scores with penalty terms, because the scaling factor $\mathrm{n}_i^{G^+}(\mathcal{D})$ is applied to the current graph but depends on the alternative graph. Our evaluation compare the gain function concept with single-model scores as baselines.

Comparison With Single-Model Likelihood Scores. The simplest approach to upgrading an i.i.d. score is to use it with relational instance counts (i.e., $S_i(G, \mathbf{n}_{ijk}^G(\mathcal{D}))$). However, this approach has the serious drawback that when a new edge increases the local sample size by connecting different populations, the likelihood decreases while the model complexity increases (see Table 3). Therefore an instance count likelihood score is not consistent, because it fails to add edges that introduce new population variables, no matter how large the sample size (see [Schulte and Gholami, 2016] for empirical confirmation). Our comparison therefore uses likelihood scores that extend the *normalized* log-likelihood score $\overline{LL}$ with a penalty term. The count method simply adds the penalty term; the normalized method divides it by the local sample size, which is equivalent to normalizing the instance count score (i.e. $S_i(G, \mathbf{n}_{ijk}^G(\mathcal{D}))/\mathrm{n}_i^G(\mathcal{D})$).

Count $\overline{LL}_i(G, \mathbf{n}_{ijk}^G(\mathcal{D})) - f^S(\#pars_i^G, \mathbf{n}_{ijk}^G(D))$

Normalized $\overline{LL}_i(G, \mathbf{n}_{ijk}^G(\mathcal{D})) - f^S(\#pars_i^G, \mathbf{n}_{ijk}^G(\mathcal{D}))/\mathrm{n}_i^G(\mathcal{D})$

Table 4 gives the corresponding formulas for the AIC and BIC penalty terms. Table 5 shows example values for the scores and gains.

Score S	$S_i(G^+, \mathbf{n}^{G^+}_{ijk}(\mathcal{D}))$	$S_i(G, \overline{\mathbf{n}}^{G}_{ijk}(\mathcal{D}))$	$\Delta\overline{S}_i(G, G^+, \mathcal{D})$
LL	$\sum_j \sum_k \mathrm{n}^{G^+}_{ijk}(\mathcal{D}) \cdot \log_2 \left(\frac{\mathrm{n}^{G^+}_{ijk}(\mathcal{D})}{\mathrm{n}^{G^+}_{ij}(\mathcal{D})} \right)$	$\sum_j \sum_k \overline{\mathbf{n}}^{G}_{ijk}(\mathcal{D}) \cdot \log_2 \left(\frac{\overline{\mathbf{n}}^{G}_{ijk}(\mathcal{D})}{\overline{\mathbf{n}}^{G}_{ij}(\mathcal{D})} \right)$	$[LL_i(G^+, \mathbf{n}^{G^+}_{ijk}(\mathcal{D}))-$ $LL_i(G, \overline{\mathbf{n}}^{G}_{ijk}(\mathcal{D}))]/\mathrm{n}^{G^+}_i(\mathcal{D})$
AIC	$LL_i(G^+, \mathbf{n}^{G^+}_{ijk}(\mathcal{D})) - \#pars_i^{G^+}$	$LL_i(G, \overline{\mathbf{n}}^{G}_{ijk}(\mathcal{D})) - \#pars_i^{G}$	$\Delta\overline{LL}_i(G, G^+, \mathcal{D})+$ $[\#pars_i^{G} - \#pars_i^{G^+}]/\mathrm{n}^{G^+}_i(\mathcal{D})$
BIC	$LL_i(G^+, \mathbf{n}^{G^+}_{ijk}(\mathcal{D}))-$ $\log_2(\mathrm{n}^{G^+}_i(\mathcal{D}))/2 \cdot \#pars_i^{G^+}$	$LL_i(G, \overline{\mathbf{n}}^{G}_{ijk}(\mathcal{D}))-$ $\log_2(\mathrm{n}^{G^+}_i(\mathcal{D}))/2 \cdot \#pars_i^{G}$	$\Delta\overline{LL}_i(G, G^+, \mathcal{D})+$ $\frac{\log_2(\mathrm{n}^{G^+}_i(\mathcal{D}))}{2\mathrm{n}^{G^+}_i(\mathcal{D})}[\#pars_i^{G} - \#pars_i^{G^+}]$

Table 2: The normalized gain for selected standard BN scores. LL denotes the log-likelihood score. $\#pars_i^H$ denotes the number of parameters for node X_i in DAG H. Some constant factors are omitted. Note that $\overline{\mathrm{n}}^{G}_i(\mathcal{D}) = \mathrm{n}^{G^+}_i(\mathcal{D})$.

Family Configuration	n_{ijk}	n_{ij}	n_i	n_{ijk}/n_i	CP	$LL_i(\cdot, \mathbf{n}^{\cdot}_{ijk}(\mathcal{D}))$	$\frac{LL_i(\cdot, \mathbf{n}^{\cdot}_{ijk}(\mathcal{D}))}{\mathrm{n}^{G^+}_i(\mathcal{D})}$
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: For the node $Age(User)$, and the IMDb dataset, the contribution of one family configuration to the unnormalized resp. normalized log-likelihood score. Top: For the G_1 structure of Figure 2. Bottom: For the expanded structure G_1^+.

Criterion	AIC_i	BIC_i
Count Score	$\#pars_i^G$	$\#pars_i^G \cdot \frac{1}{2}\log_2(\mathrm{n}^G_i(\mathcal{D}))$
Normalized Score	$\frac{\#pars_i^G}{\mathrm{n}^G_i(\mathcal{D})}$	$\#pars_i^G \cdot \frac{\log_2(\mathrm{n}^G_i(\mathcal{D})}{2\mathrm{n}^G_i(\mathcal{D})}$

Table 4: Relational Penalty Terms for the AIC and BIC scores. The evaluated scores add the penalty term to the normalized log-likelihood $\overline{LL}$.

5 Theoretical Consistency Analysis

We formalize consistency for relational data following previous work [Sakai and Yamanishi, 2013; Xiang and Neville, 2011]. The notation $\boldsymbol{N}(\mathcal{D}) \to \infty$ from denotes that each population size $\mathcal{I}_i$ goes to infinity. Similar to Sakai and Yamanishi, we make the identifiability assumption that

$$P_{\mathcal{D}}(\cdot) \to P_w(\cdot) \equiv p \text{ as } \boldsymbol{N}(\mathcal{D}) \to \infty, \tag{4}$$

where w represents a complete relational structure (network) from which samples are drawn, and p denotes the generative distribution associated with w. This assumption holds under various sampling schemes such as subgraph sampling [Frank, 1978].[2] Chickering and Meek introduced the concept of **local consistency**, which we adapt for gain functions.

Definition 1 *Let p be the data generating distribution. A gain function is **locally consistent** if the following hold as $\boldsymbol{N}(\mathcal{D}) \to \infty$, for any graph G and expansion G_+ that adds a single edge $X_+ \to X_i$ to G:*

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

[2] Assumptions for consistent parameter learning in PRMs (but not SRMs) are discussed in several papers [Xiang and Neville, 2011; Shalizi and Rinaldo, 2013; Sakai and Yamanishi, 2013].

An upgrade method **preserves local consistency** if local consistency for an i.i.d. gain function entails local consistency for its upgrade. In the sample size limit, clause 1 entails that the gain of a DAG model is (1) positive for any edge that is necessary for eliminating an independence constraint that does not hold in the generative distribution, and (2) is negative for any edge that is unnecessary. Together, these clauses ensure consistency—necessary edges are learned—and optimality—only necessary edges are learned [Chickering and Meek, 2002].

Theorem 1 *The normalized gain upgrade preserves local consistency, and therefore consistency. The single-model comparison scores do not preserve local consistency.*

Appendix A gives the local consistency proof for the normalized gain upgrade method. We provide the intuition rather than a formal proof, for why the single-model scores are not locally consistent. The count score fails Clause 1 because neither the NLL nor the parameter count increase with sample size. E.g., if G_1^+ is correct, its parameter count is 12, whereas the NLL is -1.177 (in Table 5). The penalty term will remain much bigger than the NLL even at large samples from G_1^+.

The normalized score fails Clause 2 because the number of parameters are divided by the local sample sizes. Adding a redundant edge can increase the NLL *and* decrease the normalized parameter count. For example, the parameter count 12 for G_1^+ is divided by 1,582,762, whereas the parameter count 2 for G_1 is divided by only 941 (Table 5). So even if G_1 is optimal, G_1^+ will receive a higher normalized score even at large samples drawn from G_1. This analysis predicts that the count score selects overly sparse structures, and the normalized score overly dense structures.

				AIC			BIC		
	#groundings	#parameters	$\overline{LL}$	count	normalized gain	normalized	count	normalized gain	normalized
G_1	941	2	-1.384	-3.38	—	-1.3865	-11.26	—	-1.3948
G_1^+	1582762	12	-1.177	-13.18	—	-1.1775	-124.74	—	-1.1776
GAIN		10	0.207	-9.79	0.20684	0.2090	-113.48	0.20678	0.2173

Table 5: Example values for the scores and gain functions defined in this section, for the IMDb dataset and the structures of Figure 2. Note that count gain $<$ normalized gain $<$ normalized score gain. E.g., for AIC gains $-9.79 < 0.020684 < 0.2090$.

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 6: Datasets characteristics. #Tuples = total number of tuples over all tables in the dataset. The datasets contain multiple relationships and populations of different types.

6 Empirical Evaluation

Code and Datasets Our code is available on-line.[3] We used six benchmark real-world databases from the CTU Prague Relational Learning Repository, described in [Motl and Schulte, 2015] (also available[4] in text format). Table 6 summarizes basic information about the benchmark datasets. IMDb is the largest dataset in terms of number of total table tuples (more than 1.3M tuples) and schema complexity. It combines the MovieLens database with the Internet Movie Database (IMDb). [5]

Model Search Algorithm We used the previous learn-and-join method (LAJ) for relational BN model search [Schulte and Khosravi, 2012], with the implementation provided by its creators. The LAJ method conducts a search through the lattice of relational paths, similar to the iterative deepening strategy of [Friedman *et al.*, 1999]. 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 for each of the 6 upgraded criteria.

Results For each learned graph G, we use maximum likelihood estimates to obtain a Bayesian network B to be evaluated. We report the normalized log-likelihood (NLL) of the input data and the number of parameters for each learned graph. The likelihood is the natural evaluation measure for generative learning [Van Haaren *et al.*, 2016]. Figure 3 shows the metrics for the different upgrade methods.

Count Score. On each dataset, the count score introduces *no* edges, therefore the smallest number of parameters (for instance 69 on IMDb for AIC count vs. 14,450 for normalized gain). Its NLL metric is substantially worse than the gain NLL on 4/6 databases (e.g. on Financial -12.79 for AIC count

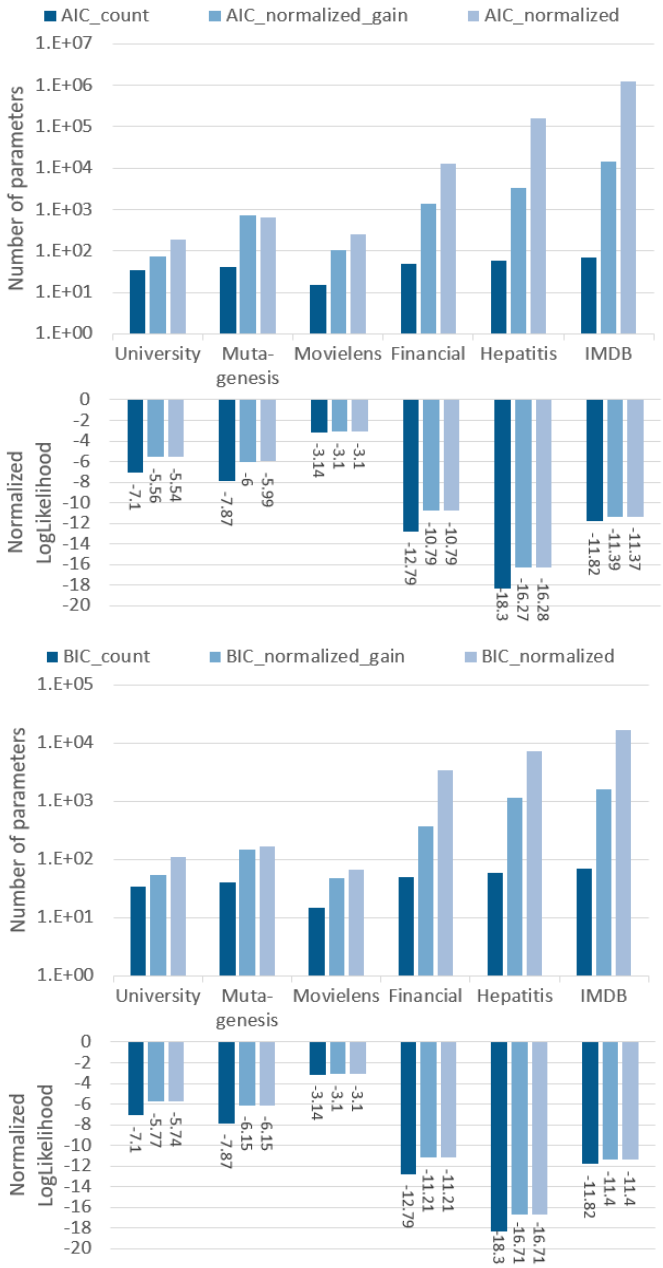

Figure 3: Log-likelihood and Number of Parameters for different relational score upgrade methods. Top: AIC upgrades. Bottom: BIC upgrades.

[3] `github.com/sfu-cl-lab/FactorBase_Consistent`

[4] `www.cs.sfu.ca/~oschulte/jbn/`

[5] `grouplens.org`, 1M version; `IMDb.com`, July 2013

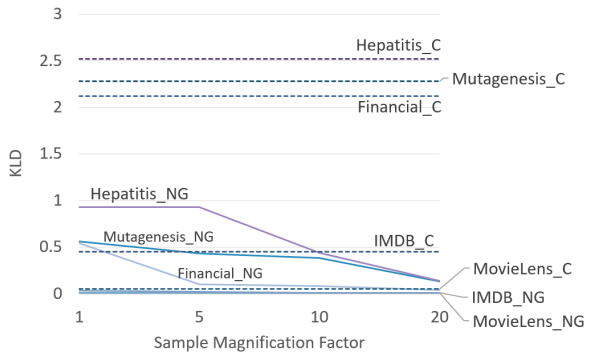

Figure 4: BN learning curves for the BIC normalized gain criterion (labelled as "dataset"_NG) and the count score (labelled as "dataset"_C). The normalized gain BNs converge to the database distribution (KLD = 0). The BIC count score (_C) selects the empty graph on all data sets, so its KLD remains constant.

vs. -10.79 for the normalized gain). The relatively small absolute difference in NLL on MovieLens and IMDb is due to weak correlations and large local sample sizes. *We conclude that the count score structures are overly sparse.*

Normalized Score. On each dataset, the normalized score produces the largest number of parameters (for instance 1,199,853 on IMDb for AIC count vs. 14,450 for normalized gain). The normalized gain function achieves almost the same NLL with many fewer parameters. *We conclude that the count score structures are overly dense.*

Consistency. The BIC score is consistent for i.i.d. data [Chickering and Meek, 2002], so Theorem 1 entails that the BIC normalized gain is also consistent for relational data. Figure 4 empirically verifies the consistency of the normalized gain criterion, by showing the convergence to the database distribution on our benchmark databases, meaning that the standard Kulback-Leibler divergence (KLD) metric goes to 0. Similar to previous experiments [Getoor *et al.*, 2001a; Schulte *et al.*, 2014], we duplicate entities by a magnification factor of $m = 1, 5, 10, 20$, which multiplies local sample sizes by m. (We leave out the small University dataset where convergence requires a higher magnification factor.) The BIC count score adds no edges even with larger sample sizes, so it fails to be consistent. The BIC normalized score outputs a denser graph with KLD equivalent to the normalized gain score (Figure 3). So it is consistent but not locally consistent because it selects redundant edges.

7 Conclusion and Future Work

Generalizing i.i.d. model scores designed for i.i.d. data is an important fundamental topic for relational learning. The normalized gain, which measures the difference in data fit between two BN structures, is a novel scalable method for generalizing a Bayesian network score. For complete data, it can be computed in closed form given the BN sufficient statistics. Normalized gain functions preserve the convergence guarantees of i.i.d. scores, and show good empirical performance: they select structures that succinctly represent the data correlations, compared with baseline single-model scores.

A promising avenue for future work is to apply our approach to other statistical-relational models, such as Markov Logic Networks. Implementing a BN structure learning system for functors that represent complex terms would allow us to apply the normalized gain score with aggregate functions/combining rules.

A Local Consistency Proof for the Normalized Gain Upgrade Method (Theorem 1)

We show the local consistency of the rescaled gain, which is the normalized gain with rescaled sufficient statistics but without dividing by the local sample size:

$$\Delta R(G, G^+, D) \equiv S_i(G^+, \mathbf{n}^{G^+}_{ijk}(\mathcal{D})) - S_i(G, \overline{\mathbf{n}}^{G}_{ijk}(\mathcal{D})) \quad (5)$$

Since the rescaled gain has the same sign as the normalized gain, *proving the local consistency of the rescaled gain implies the local consistency of the normalized gain.* We say that an edge **adds a population variable** if the parent contains a population variable that is not contained in the child.

Case 1: The additional edge $X_+ \rightarrow X_i$ adds no population variables. For such edges, the rescaled counts are the same as the nonscaled counts used in the original i.i.d. score (i.e., $\overline{\mathbf{n}}^{G}_{ijk}(\mathcal{D}) = \mathbf{n}^{G}_{ijk}(\mathcal{D})$). So the arguments of [Chickering and Meek, 2002] can be applied to relational data, and the rescaled gain score is locally consistent in this case.

Case 2: The additional edge $X_+ \rightarrow X_i$ adds a population variable. For concreteness, assume that the edge is of the form $g(\mathbb{A}, \mathbb{B}) \rightarrow f(\mathbb{A})$, so the added sample size is $N[\mathbb{B}; \mathcal{D}]$. Consider a transformed database $\mathcal{D}'$ where $f(\mathbb{A})$ is replaced by $f'(\mathbb{A}, \mathbb{B})$, with an inert second argument: $f'(a, b) = f(a)$. Since in the transformed schema, the additional edge $X_+ \rightarrow X'_i$ does not add a population variable, from case 1 we conclude that (i) the rescaled gain is locally consistent when applied to the transformed data $\mathcal{D}'$. We next show that local consistency for $\mathcal{D}'$ data implies local consistency in $\mathcal{D}$ data. The transformation does not change the information content and is equivalent to rescaling counts:

$$\mathbf{n}^{G}_{ijk}(\mathcal{D}') = \mathbf{n}^{G}_{ijk}(\mathcal{D}) \times N[\mathbb{B}; \mathcal{D}] = \overline{\mathbf{n}}^{G}_{ijk}(\mathcal{D}).$$

Since we also have $\mathbf{n}^{G_+}_{ijk}(\mathcal{D}') = \mathbf{n}^{G_+}_{ijk}(\mathcal{D})$, the transformed and the original data agree on the rescaled gain:

$$\Delta R(G, G^+, \mathcal{D}) = \Delta R(G, G^+, \mathcal{D}'), \quad (6)$$

and agree on the conditional probabilities of a child node value given parent node values:

$$\frac{\mathrm{n}^{G}_{ijk}(\mathcal{D})}{\mathrm{n}^{G}_{ij}(\mathcal{D})} = \frac{\mathrm{n}^{G}_{ijk}(\mathcal{D}')}{\mathrm{n}^{G}_{ij}(\mathcal{D}')}. \quad (7)$$

Therefore the identifiability condition (4) for the original data $\mathcal{D}$ entails that in the sample size limit, the transformed data $\mathcal{D}'$ identify whether node X_i is independent of X_+ given its parents in the data generating distribution p. So by condition (6), the rescaled gain is locally consistent for the original data $\mathcal{D}$. Hence in either case, the normalized gain is locally consistent.

References

[Bouckaert, 1995] R. Bouckaert. *Bayesian belief networks: from construction to inference*. PhD thesis, U. Utrecht, 1995.

[Buchman and Poole, 2015] David Buchman and David Poole. Representing aggregators in relational probabilistic models. In *AAAI*, pages 3489–3495, 2015.

[Chickering and Meek, 2002] David Maxwell Chickering and Christopher Meek. Finding optimal Bayesian networks. In *UAI*, pages 94–102, 2002.

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

[Frank, 1978] O. Frank. Estimation of the number of connected components in a graph by using a sampled subgraph. *Scandinavian Journal of Statistics*, pages 177–188, 1978.

[Friedman *et al.*, 1999] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *IJCAI*, volume 99, pages 1300–1309, 1999.

[Getoor and Taskar, 2007] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[Getoor *et al.*, 2001a] Lise Getoor, Benjamin Taskar, and Daphne Koller. Selectivity estimation using probabilistic models. *ACM SIGMOD Record*, 30(2):461–472, 2001.

[Getoor *et al.*, 2001b] Lise Getoor Getoor, Nir Friedman, and Benjamin Taskar. Learning probabilistic models of relational structure. In *ICML*, pages 170–177. Morgan Kaufmann, 2001.

[Getoor, 2001] Lise Getoor. *Learning Statistical Models From Relational Data*. PhD thesis, Department of Computer Science, Stanford University, 2001.

[Gholami, 2016] Sajjad Gholami. Upgrading Bayesian network scores for multi-relational data. Master's thesis, School of Computing Science, Simon Fraser University, 2016.

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

[Kimmig *et al.*, 2014] Angelika Kimmig, Lilyana Mihalkova, and Lise Getoor. Lifted graphical models: a survey. *Machine Learning*, pages 1–45, 2014.

[Laer and de Raedt, 2001] Wim Van Laer and Luc de Raedt. How to upgrade propositional learners to first-order logic: A case study. In *Relational Data Mining*. Springer Verlag, 2001.

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

[Motl and Schulte, 2015] Jan Motl and Oliver Schulte. The CTU prague relational learning repository. *CoRR*, abs/1511.03086, 2015.

[Nickel *et al.*, 2016] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.

[Niu *et al.*, 2011] Feng Niu, Christopher Ré, AnHai Doan, and Jude W. Shavlik. Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. *PVLDB*, 4(6):373–384, 2011.

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

[Poole, 2003] David Poole. First-order probabilistic inference. In *IJCAI*, 2003.

[Quinlan and Cameron-Jones, 1993] J. Ross Quinlan and R. Mike Cameron-Jones. Foil: A midterm report. In *ECML*, volume 667, pages 3–20. Springer, 1993.

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

[Russell, 2015] Stuart Russell. Unifying logic and probability. *Communications of the ACM*, 58(7):88–97, 2015.

[Sakai and Yamanishi, 2013] Yoshiki Sakai and Kenji Yamanishi. An NML-based model selection criterion for general relational data modeling. In *Big Data*, pages 421–429. IEEE, 2013.

[Schulte and Gholami, 2016] Oliver Schulte and Sajjad Gholami. Consistent bayesian network scores for multi-relational data. StarAI-IJCAI Workshop, July 2016.

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

[Schulte *et al.*, 2014] Oliver Schulte, Hassan Khosravi, Arthur Kirkpatrick, Tianxiang Gao, and Yuke Zhu. Modelling relational statistics with Bayes nets. *Machine Learning*, 94:105–125, 2014.

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

[Shalizi and Rinaldo, 2013] Cosma Rohilla Shalizi and Alessandro Rinaldo. Consistency under sampling of exponential random graph models. *Annals of Statistics*, 41(2):508, 2013.

[Van Haaren *et al.*, 2016] Jan Van Haaren, Guy Van den Broeck, Wannes Meert, and Jesse Davis. Lifted generative learning of Markov logic networks. *Machine Learning*, 103(1):27–55, 2016.

[Wang *et al.*, 2008] Daisy Zhe Wang, Eirinaios Michelakis, Minos Garofalakis, and Joseph M Hellerstein. Bayesstore: managing large, uncertain data repositories with probabilistic graphical models. In *Proceedings VLDB*, pages 340–351. VLDB Endowment, 2008.

[Xiang and Neville, 2011] Rongjing Xiang and Jennifer Neville. Relational learning with one network: An asymptotic analysis. In *AISTATS*, pages 779–788, 2011.