Model Selection Scores for Multi-Relational Bayesian Networks[*] Extended Abstract for DeLBP Workshop at IJCAI 2017

Sajjad Gholami, Oliver Schulte, Vidhi Jian, Qiang Zhao
School of Computing Science
Simon Fraser University, Burnaby, Canada
{oschulte,sgholami}@cs.sfu.ca

Abstract

Many organizations maintain their data in a relational database, which contains information about entities, their attributes, relationships among the entities, and attributes of the relationships. Statistical-relational learning (SRL) aims to generalize traditional single-table machine learning methods for multi-relational data. Many SRL models are defined using a combination of graphs and first-order logic. This lecture addresses the task of learning the graph structure of a first-order Bayesian network (BN). A key component of structure learning is a model selection score that measures how well a model fits a dataset. We introduce a new method that generalizes for multi-relational databases, a BN score designed for single-table data. We present several applications that leverage a learned model, such as modeling database statistics, exception mining, and extracting features for classification and anomaly detection.

1 Introduction: Relational Learning

Multi-relational databases in SQL format are very widely used to store enterprise data. Relational data are also known as network data, graph data, matrix data, and tensor data. Traditional machine learning analyzes data represented in a single table; such data can be viewed as a special limiting case of multi-relational data with no relationships [Nickel *et al.*, 2016]. The field of statistical-relational learning (SRL) aims to generalize single-table machine learning methods for multi-relational data; this is called *upgrading* the method [Getoor and Taskar, 2007; Laer and de Raedt, 2001]. Application domains for statistical-relational models include natural language processing, 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.*, 2001; Wang *et al.*, 2008]. This lecture addresses the important SRL task of learning the structure of a first-order Bayesian structure from a relational dataset. Our presentation describes several applications that leverage a learned model, such as modeling database statistics, exception mining, and extracting features for classification and anomaly detection.

The most common approach to BN structure learning is to search for a structure that maximizes a model selection score for a given dataset. Our companion paper [Schulte and Gholami, 2017] introduces a general method for upgrading BN model selection scores. This outline illustrates the method for the case of likelihood-based scores, which take the form (log-likelihood of data under model) - penalty(model, sample size, #number parameters). The full paper defines the method for BN scores in general, with more examples and references.

2 Background and Notation

We assume familarity with basic BN concepts such as DAGs and conditional probability tables. 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 $\boldsymbol{x}_i$.

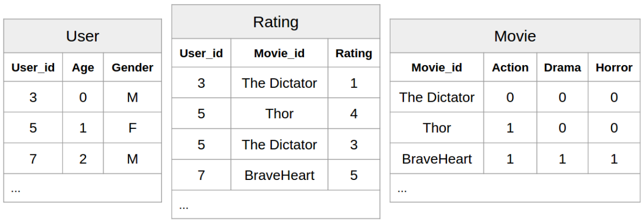
Relational Data A multi-relational model is typically a multi-population model. A **population** is a set of individuals of the same type (e.g., a set of $Users$, a set of $Movies$). Individuals are denoted by constants (e.g., $user_3$,$thor$). A k-ary **functor**, denoted f, f' etc., maps a tuple of k individuals to a value from the functor's **domain**. The arguments of a functor are restricted to appropriate types. Throughout the paper we assume complete data. A complete relational **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

[*] Supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada.



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.

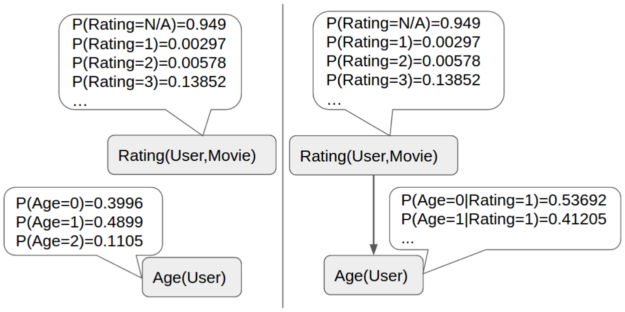


Figure 2: Example First-Order Bayesian networks: left = B_1 with graph G_1, right = B_1^+ with graph G_1^+.

the appropriate type. A first-order random variable (FORV) is a term with at least one population variable [Wang *et al.*, 2008]. A first-order Bayesian network (FOB) [Wang *et al.*, 2008], aka Parametrized BN [Kimmig *et al.*, 2014], is a BN whose nodes are FORVs. Figure 2 shows two 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.

The **database frequency** [Halpern, 1990] of an assignment $\boldsymbol{X} = \boldsymbol{x}$ is the number of satisfying groundings over the number of possible groundings:

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

where $\mathrm{n}\left[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}\right]$ denotes the **number of satisfying groundings** that satisfy the assignment in database $\mathcal{D}$, and $N\left[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}\right]$ denotes the total number of possible groundings of the variables in the list $\boldsymbol{X}$.

Using the standard BN product formula, a FOB B represents a joint distribution over assignments to first-order random variables, written $P_B(\boldsymbol{X} = \boldsymbol{x})$. A model selection score measures how well the model distribution P_B fits the empirical or *database distribution* $P_{\mathcal{D}}(\boldsymbol{X} = \boldsymbol{x})$. Table 1 compares database frequencies using the IMDb dataset to BN model probabilities. The expanded BN B_1^+ matches the database distribution perfectly but at the cost of more parameters.

3 Relational Likelihood Score

A fundamental model score is the class likelihood function, which measures how likely the data is given the model. In previous work on parameter learning, [Xiang and Neville, 2011; Schulte, 2011], the log-likelihood score LL for i.i.d.

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.

$\boldsymbol{X} = \boldsymbol{x}$	$Age(User) = 0$	$Age(User) = 0,$ $Rating(User, Movie) = 1$
$\mathrm{n}\left[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}\right]$	376	2,524
$N\left[\boldsymbol{X} = \boldsymbol{x}; \mathcal{D}\right]$	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	$0.00297 \cdot 0.53692 \approx 0.0016$

data was upgraded by the **normalized log-likelihood score** NLL. The NLL score can be computed in closed-form given the BN **sufficient statistics**, which we denote as follows. 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.

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

In relational sufficient statistics *the local sample size* $\mathrm{n}_i^G(\mathcal{D})$ *depends on the graph structure* whereas in i.i.d. data, the number of data points defines a global sample size that is the same for all nodes and all graph structures. The NLL score is defined as

$$\overline{LL}_i(G, \mathcal{D}) \equiv \sum_i \frac{1}{\mathrm{n}_i^G(\mathcal{D})} \sum_j \sum_k \mathrm{n}_{ijk}^G(\mathcal{D}) \cdot \log_2 \left(\frac{\mathrm{n}_{ijk}^G(\mathcal{D})}{\mathrm{n}_{ij}^G(\mathcal{D})} \right)$$

The normalization $1/\mathrm{n}_i^G(\mathcal{D})$ converts different sufficient statistics to proportions and therefore the same [0,1] scale. Table 2 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.

4 Relational Likelihood-Based Scores

In i.i.d. data, a likelihood-based score S subtracts a model complexity term from the log-likelihood score. We subtract a model complexity term from the NLL score to compare two BN structures G and G^+. Assuming that G^+ adds edges to G, our method is to compute, the improvement or **normalized gain** of G^+ over G as follows:

$$\left[\overline{LL}_i(G^+, \mathcal{D}) - \frac{f_i(G^+)}{\mathrm{n}_i^{G^+}(\mathcal{D})}\right] - \left[\overline{LL}_i(G, \mathcal{D}) - \frac{f_i(G)}{\mathrm{n}_i^{G^+}(\mathcal{D})}\right] \qquad (2)$$

where for the smaller BN B, the local complexity $f_i(B)$ is computed using the sample size for the *larger* structure. Normalizing *both* penalty terms by the same sample size measures them on the same scale. Table 3 gives the formulas for

Family Configuration	n_{ijk}	n_{ij}	n_i	n_{ijk}/n_i	*CP*	$LL_i(\cdot, \mathbf{n}_{ijk}^{\cdot}(\mathcal{D}))$	$\frac{LL_i(\cdot,\mathbf{n}_{ijk}^{\cdot}(\mathcal{D}))}{\mathrm{n}_i^{G+}(\mathcal{D})}$
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 2: 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^+.

the AIC and BIC penalty terms. Table 4 shows example values for the gains.

Since the normalized gain term for the smaller structure G depends on the sufficient statistics for the comparison structure G^+, *the normalized gain cannot be represented as the differential of two single-model scores.* Our baseline single-model scores extend the normalized log-likelihood score $\overline{LL}$ with a penalty term. The **count method** simply adds the penalty term $f_i(G)$; the **normalized method** divides the penalty term by the local sample size (i.e., $f_i(G)/\mathrm{n}_i^G(\mathcal{D})$).

AIC_i	BIC_i
$\frac{(\#pars_i^{G+} - \#pars_i^{G})}{\mathrm{n}_i^{G+}(\mathcal{D})}$	$\frac{(\#pars_i^{G+} - \#pars_i^{G})\log_2(\mathrm{n}_i^{G+}(\mathcal{D}))}{2\mathrm{n}_i^{G+}(\mathcal{D})}$

Table 3: Relational Local Penalty Terms for the AIC and BIC scores. $\#pars_i^{G+}$ is the number of parameters for node i. The normalized gain adds the penalty term to the normalized log-likelihood differential (2).

5 Theoretical Analysis

We formalize consistency for relational data following previous work [Sakai and Yamanishi, 2013; Xiang and Neville, 2011]. The notation $\boldsymbol{N}(\mathcal{D}) \rightarrow \infty$ from denotes that each population size $\mathcal{I}_i$ goes to infinity. Chickering and Meek 2002 introduced the concept of **local consistency**, which we adapt for gain functions. Let p be the data generating distribution. A gain function is **locally consistent** if the following hold as $\boldsymbol{N}(\mathcal{D}) \rightarrow \infty$, for any graph G and expansion G_+ that adds a single edge $X_+ \rightarrow X_i$ to G: The gain of a DAG model is (1) positive for any edge that is necessary for representing the generative distribution p, and (2) is negative for any edge that is unnecessary. These clauses ensure statistical consistency—necessary edges are learned—and optimality—only necessary edges are learned . A relational upgrade method **preserves local consistency** if local consistency for a single-table gain function entails local consistency for its upgrade.

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

6 Empirical Results

We learn Bayesian network structures with the three model comparison criteria shown in Table 3 on 6 benchmark datasets. The count score selects the *empty graph* on all. The normalized score selects graphs with many edges (almost complete). The normalized gain selects informative structures that strike a desirable balance between overly sparse and overly dense graphs.

7 Conclusion

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

References

[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].

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

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

[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.

[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.

				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 4: 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 $-9.79 < 0.020684 < 0.2090$.

[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.

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

[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, 2013 IEEE International Conference on*, pages 421–429. IEEE, 2013.

[Schulte and Gholami, 2017] Oliver Schulte and Sajjad Gholami. Locally consistent bayesian network scores for multi-relational data. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[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.

[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 *International Conference on Artificial Intelligence and Statistics*, pages 779–788, 2011.