A Tractable Pseudo-Likelihood for Bayes Nets Applied To Relational Data

Oliver Schulte
School of Computing Science
Simon Fraser University
Vancouver, Canada
Relational Databases dominate in practice.
• Want to apply Machine Learning ➔ **Statistical-Relational Learning**.
• Fundamental issue: how to combine logic and probability?

Typical SRL Tasks
• **Link-based Classification**: predict the class label of a target entity, given the links of a target entity and the attributes of related entities.
• **Link Prediction**: predict the existence of a link, given the attributes of entities and their other links.
• **Generative Modelling**: represent the joint distribution over links and attributes. ★Today
Statistical Learning requires a **quantitative measure** of data fit.  
   e.g., BIC, AIC: log-likelihood of data given model + complexity penalty.

- In relational data, *units are interdependent*  
  ⇒ no product likelihood function for model.

- Proposal of this talk: use **pseudo likelihood**.
  - *Unnormalized* product likelihood.
  - Like independent-unit likelihood, but with event frequencies instead of event counts.
Outline

1. Relational databases.
2. Bayes Nets for Relational Data (Poole IJCAI 2003).
4. Random Selection Semantics.
5. Parameter Learning.
Database Instance based on Entity-Relationship (ER) Model

<table>
<thead>
<tr>
<th>Name</th>
<th>intelligence</th>
<th>ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Kim</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Paul</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>popularity</th>
<th>Ability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oliver</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>David</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S.name</th>
<th>C.number</th>
<th>grade</th>
<th>satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>101</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>Jack</td>
<td>102</td>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>Kim</td>
<td>102</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>Kim</td>
<td>103</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>Paul</td>
<td>101</td>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>Paul</td>
<td>102</td>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number</th>
<th>Prof</th>
<th>rating</th>
<th>difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Oliver</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>102</td>
<td>David</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>103</td>
<td>Oliver</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Key fields are underlined.
Nonkey fields are deterministic functions of key fields.
Relational Data: what are the random variables (nodes)?

- A functor is a function or predicate symbol (Prolog).
- A **functor random variable** is a functor with 1\(^{st}\)-order variables \(f(X), g(X, Y), R(X, Y)\).
- Each variable \(X, Y, \ldots\) ranges over a **population** or domain.
- A **Functor Bayes Net**\(^*\) (FBN) is a Bayes Net whose nodes are functor random variables.
- Highly expressive (Domingos and Richardson MLJ 2006, Getoor and Grant MLJ 2006).

Example: Functor Bayes Nets

- Parameters: conditional probabilities $P(\text{child} | \text{parents})$.
- Defines joint probability for every conjunction of value assignments.

What is the interpretation of the joint probability?
Random Selection Semantics of Functors

• Intuitively, \( P(\text{Flies}(X) \mid \text{Bird}(X)) = 90\% \) means “the probability that a randomly chosen bird flies is 90%”.
• Think of \( X \) as a random variable that selects a member of its associated population with uniform probability.
• Nodes like \( f(X), g(X, Y) \) are functions of random variables, hence themselves random variables.

Random Selection Semantics: Examples

- $P(X = \text{Anna}) = 1/2$.
- $P(\text{Smokes}(X) = T) = \sum_{x:\text{Smokes}(x) = T} 1 / |X| = 1$.
- $P(\text{Friend}(X, Y) = T) = \sum_{x,y:\text{Friend}(x,y)} 1 / (|X| |Y|)$.

- The database frequency of a functor assignment is the number of satisfying instantiations or groundings, divided by the total possible number of groundings.

<table>
<thead>
<tr>
<th>Users</th>
<th>Smokes</th>
<th>Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anna</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bob</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Friend</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name1</td>
<td>Name2</td>
</tr>
<tr>
<td>Anna</td>
<td>Bob</td>
</tr>
<tr>
<td>Bob</td>
<td>Anna</td>
</tr>
</tbody>
</table>
Likelihood Function for Single-Table Data

\[
\ln P(T | B) = \sum_{\text{nodes } i} \sum_{\text{values } k} \sum_{\text{parent-states } j} n_T(v_i = k, pa_i = j) \ln P_B(v_i = k | pa_i = j)
\]

Table \( T \) count of co-occurrences of child node value and parent state

Parameter of Bayes net \( B \)

Users

<table>
<thead>
<tr>
<th>Name</th>
<th>Smokes</th>
<th>Cancer</th>
<th>( P_B )</th>
<th>( \ln(P_B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anna</td>
<td>T</td>
<td>T</td>
<td>0.36</td>
<td>-1.02</td>
</tr>
<tr>
<td>Bob</td>
<td>T</td>
<td>F</td>
<td>0.14</td>
<td>-1.96</td>
</tr>
</tbody>
</table>

Likelihood/Log-likelihood

\[
\prod \approx 0.05 \quad \Sigma = -2.98
\]

\[
P(T | B) \quad \ln P(T | B)
\]
Proposed Pseudo Log-Likelihood

For database D:

\[
\ln P^*(D|B) = \sum_{\text{nodes}} \sum_{i \text{ values}} \sum_{k \text{ parent-states}} \sum_{j} P_D(v_i = k, pa_i = j) \ln P_B(v_i = k|pa_i = j)
\]

**Database D**

**frequency** of co-occurrences of child node value and parent state

<table>
<thead>
<tr>
<th>Name</th>
<th>Smokes</th>
<th>Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anna</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bob</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

**Friend**

<table>
<thead>
<tr>
<th>Name1</th>
<th>Name2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anna</td>
<td>Bob</td>
</tr>
<tr>
<td>Bob</td>
<td>Anna</td>
</tr>
</tbody>
</table>

**Users**

\[\begin{align*}
\text{Smokes}(X) = &\ T \\
\text{Friend}(X,Y) = &\ T \\
\text{Smokes}(Y) = &\ T \\
\text{Cancer}(Y) = &\ T
\end{align*}\]
1. Randomly select instances $X_1 = x_1, \ldots, X_n = x_n$ for each variable in FBN.
2. Look up their properties, relationships in database.
3. Compute log-likelihood for the FBN assignment obtained from the instances.
4. $L^R = \text{expected log-likelihood over uniform random selection of instances.}$

**Proposition** The random selection log-likelihood equals the pseudo log-likelihood.

\[
L^R = -(2.254 + 1.406 + 1.338 + 2.185)/4 \approx -1.8
\]
Proposition For a given database D, the parameter values that maximize the pseudo likelihood are the empirical conditional frequencies in the database.
In principle, just replace single-table likelihood by pseudo likelihood.

Efficient new algorithm (Khosravi, Schulte et al. AAAI 2010). Key ideas:

- Use single-table BN learner as black box module.
- Level-wise search through table join lattice. Results from shorter paths are propagated to longer paths (think APRIORI).
### Running time on benchmarks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>JBN</th>
<th>MLN</th>
<th>CMLN</th>
</tr>
</thead>
<tbody>
<tr>
<td>University</td>
<td>0.03 + 0.32</td>
<td>5.02</td>
<td>11.44</td>
</tr>
<tr>
<td>MovieLens</td>
<td>1.2 + 120</td>
<td>NT</td>
<td>NT</td>
</tr>
<tr>
<td>MovieLens Subsample 1</td>
<td>0.05 + 0.33</td>
<td>44</td>
<td>121.5</td>
</tr>
<tr>
<td>MovieLens Subsample 2</td>
<td>0.12 + 5.10</td>
<td>2760</td>
<td>1286</td>
</tr>
<tr>
<td>Mutagenesis</td>
<td>0.5 + NT</td>
<td>NT</td>
<td>NT</td>
</tr>
<tr>
<td>Mutagenesis subsample 1</td>
<td>0.1 + 5</td>
<td>3360</td>
<td>900</td>
</tr>
<tr>
<td>Mutagenesis subsample 2</td>
<td>0.2 + 12</td>
<td>NT</td>
<td>3120</td>
</tr>
</tbody>
</table>

- Time in Minutes. NT = did not terminate.
- \( x + y = \) structure learning + parametrization (with Markov net methods).
- JBN: Our join-based algorithm.
- MLN, CMLN: standard programs from the U of Washington (Alchemy)
• Inference: use MLN algorithm after moralizing.
• Task (Kok and Domingos ICML 2005):
  • remove one fact from database, predict given all others.
  • report average accuracy over all facts.
Summary: Likelihood for relational data.

- Combining relational databases and statistics.
  - Very important in practice.
  - Combine logic and probability.
- Interdependent units → hard to define model likelihood.
- Proposal: Consider a randomly selected small group of individuals.
- Pseudo log-likelihood = expected log-likelihood of randomly selected group.
Summary: Statistics with Pseudo-Likelihood

- **Theorem**: Random pseudo log-likelihood equivalent to standard single-table likelihood, replacing table counts with database frequencies.
- Maximum likelihood estimates = database frequencies.
- Efficient Model Selection Algorithm based on lattice search.
- In simulations, very fast (minutes vs. days), much better predictive accuracy.
Thank you!

- Any questions?
Comparison With Markov Logic Networks (MLNs)

- MLNs are basically undirected graphs with functor nodes.
  - Let MBN = Bayes net converted to MLN.
  - Log-likelihood of MBN = pseudo log-likelihood of B + normalization constant.

One of the most successful statistical-relational formalisms

\[
\ln P(D | MBN) = \ln P*(D | BN) + \ln(Z)
\]

---

Likelihood Functions for Parametrized Bayes Nets

Problem: Given a database $D$ and an FBN model $B$, how to define model likelihood $P(D | B)$?

Fundamental Issue: interdependent units, not iid.

Previous approaches:

1. Introduce latent variables such that units are independent conditional on hidden “state” (e.g., Kersting et al. IJCAI 2009).
   - Different model class, computationally demanding.
   - Related to nonnegative matrix factorization----Netlix challenge.

2. Grounding, or Knowledge-based Model Construction (Ngo and Haddaway, 1997; Koller and Pfeffer, 1997; Haddaway, 1999; Poole 2003).
   - Can lead to cyclic graphs.

• Assign unobserved values \( u(jack) \), \( u(jane) \).
• Probability that Jack and Jane are friends depends on their unobserved “type”.
• In ground model, \( rich(jack) \) and \( rich(jane) \) are correlated given that they are friends, but neither is an ancestor.
• $1M prize in Netflix challenge.
• Also for multiple types of relationships (Kersting et al. 2009).
• Computationally demanding.
The Cyclicity Problem

Class-level model (template)

Ground model

- With recursive relationships, get cycles in ground model even if none in 1st-order model.
- Jensen and Neville 2007: “The acyclicity constraints of directed models severely constrain their applicability to relational data.”
Undirected Models Avoid Cycles

Class-level model (template)

Ground model
Choice of Functors

- Can have complex functors, e.g.
  - Nested: $wealth(father(father(X)))$.
  - Aggregate: $AVG_C \{ grade(S, C) : Registered(S, C) \}$.

- In remainder of this talk, use functors corresponding to
  - Attributes (columns), e.g., $intelligence(S)$, $grade(S, C)$
  - Boolean Relationship indicators, e.g. $Friend(X, Y)$.