Causal Modelling for Relational Data

Oliver Schulte
School of Computing Science
Simon Fraser University
Vancouver, Canada
Outline

- Relational Data vs. Single-Table Data
- Two key questions
  - Definition of Nodes (Random Variables)
  - Measuring Fit of Model to Relational Data

Previous Work
- Parametrized Bayes Nets (Poole 2003), Markov Logic Networks (Domingos 2005).
- The Cyclicity Problem.

New Work
- The Learn-and-Join Bayes Net Learning Algorithm.
- A Pseudo-Likelihood Function for Relational Bayes Nets.
Single Data Table Statistics

Traditional Paradigm Problem

- Single population
- Random variables = attributes of population members.
- “flat” data, can be represented in single table.

<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>

Causal Modelling for Relational Data - CFE 2010
Organizational Database/Science

- Structured Data.
- Multiple Populations.
- Taxonomies, Ontologies, nested Populations.
- **Relational Structures.**
Relational Databases

- Input Data: A finite (small) model/interpretation/possible world.

⇒ Multiple Interrelated Tables.

<table>
<thead>
<tr>
<th>s-id</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>c-id</th>
<th>Rating</th>
<th>Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>102</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p-id</th>
<th>Popularity</th>
<th>Teaching-a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oliver</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Jim</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>s-id</th>
<th>p-id</th>
<th>Salary</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>Oliver</td>
<td>High</td>
<td>3</td>
</tr>
<tr>
<td>Kim</td>
<td>Oliver</td>
<td>Low</td>
<td>1</td>
</tr>
<tr>
<td>Paul</td>
<td>Jim</td>
<td>Med</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>s-id</th>
<th>c-id</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>Paul</td>
<td>101</td>
<td>B</td>
<td>1</td>
</tr>
</tbody>
</table>
Link based Classification

- \( P(\text{diff}(101)) \)?

<table>
<thead>
<tr>
<th>Student</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>Course</th>
<th>c-id</th>
<th>Rating</th>
<th>Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>101</td>
<td>3</td>
<td>???</td>
</tr>
<tr>
<td></td>
<td>102</td>
<td>2</td>
<td>RA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Professor</th>
<th>p-id</th>
<th>Popularity</th>
<th>Teaching-a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oliver</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Jim</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Registration</th>
<th>s-id</th>
<th>c.id</th>
<th>Grade</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Jack</td>
<td>101</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Jack</td>
<td>102</td>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Kim</td>
<td>102</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Paul</td>
<td>101</td>
<td>B</td>
<td>1</td>
</tr>
</tbody>
</table>
Link prediction

- $P(\text{Registered}(\text{jack,101}))$?
Relational Data: what are the random variables (nodes)?

- A **functor** is a function symbol with 1st-order variables $f(X)$, $g(X, Y)$, $R(X, Y)$.
- Each variable ranges over a **population** or domain.
- A Parametrized Bayes Net (PBN) is a BN whose nodes are functors (Poole UAI 2003).
- Single-table data = all functors contain the same single free variable $X$. 
Example: Functors and Parametrized Bayes Nets

- Parameters: conditional probabilities $P(\text{child} \mid \text{parents})$.
- e.g., $P(\text{wealth}(Y) = T \mid \text{wealth}(X) = T, \text{Friend}(X,Y) = T)$
- defines joint probability for every conjunction of value assignments.
Domain Semantics of Functors

• Halpern 1990, Bacchus 1990
• Intuitively, $P(\text{Flies}(X) \mid \text{Bird}(X)) = 90\%$ means “the probability that a randomly chosen bird flies is 90\%”.
• Think of a variable $X$ as a random variable that selects a member of its associated population with uniform probability.
• Then functors like $f(X), g(X, Y)$ are functions of random variables, hence themselves random variables.
Domain Semantics: Examples

• \( P(S = jack) = \frac{1}{3} \).
• \( P(\text{age}(S) = 20) = \sum_{s: \text{age}(s) = 20} \frac{1}{|S|} \).
• \( P(\text{Friend}(X, Y) = T) = \sum_{x,y: \text{friend}(x,y)} \frac{1}{(|X| \cdot |Y|)} \).
• In general, the domain frequency is the number of satisfying instantiations or groundings, divided by the total possible number of groundings.
• The database tables define a set of populations with attributes and links \( \Rightarrow \text{database distribution} \) over functor values.
Defining Likelihood Functions for Relational Data

- Need a quantitative measure of how well a model fits the data.
- Single-table data consists of identically and independently structured entities (IID).
- **Relational data is not IID.**
  - Likelihood function ≠ simple product of instance likelihoods.
Knowledge-based Model Construction

- 1st-order model = template.
- Instantiate with individuals from database (fixed!) $\rightarrow$ ground model.
- Isomorphism DB facts $\leftrightarrow$ assignment of values $\rightarrow$ likelihood measure for DB.

Class-level Template
with 1st-order Variables

Instance-level Model w/
domain(S) = \{jack, jane\}
domain(C) = \{100, 200\}
The Combining Problem

- How do we combine information from different related entities (courses)?
- Aggregate properties of related entities (PRMs; Getoor, Koller, Friedman).
- Combine probabilities. (BLPs; Poole, deRaedt, Kersting.)
The Cyclicality Problem

Class-level model (template)

\[ \text{Rich}(X) \]
\[ \text{Friend}(X,Y) \]
\[ \text{Rich}(Y) \]

Ground model

\[ \text{Rich}(a) \]
\[ \text{Friend}(a,b) \]
\[ \text{Friend}(b,c) \]
\[ \text{Friend}(c,a) \]

\[ \text{Rich}(b) \]
\[ \text{Rich}(c) \]
\[ \text{Rich}(a) \]

• 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.”
• 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.
Undirected Models Avoid Cycles

Class-level model (template)

Ground model
Markov Network Example

- **Undirected** graphical model

- Potential functions defined over cliques

\[ P(x) = \frac{1}{Z} \prod_{c} \Phi_{c}(x_{c}) \]

\[ Z = \sum_{x} \prod_{c} \Phi_{c}(x_{c}) \]

<table>
<thead>
<tr>
<th>Smoking</th>
<th>Cancer</th>
<th>( \Phi(S,C) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>4.5</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>4.5</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>2.7</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>4.5</td>
</tr>
</tbody>
</table>
Markov Logic Networks

- Domingos and Richardson ML 2006
- An MLN is a set of formulas with weights.
- Graphically, a Markov network with functor nodes.
  - Solves the combining and the cyclicity problems.
- For every functor BN, there is a predictively equivalent MLN (the moralized BN).

![Graphical representation of Markov Logic Networks](image)

Causal Modelling for Relational Data - CFE 2010
New Proposal

• Causality at token level (instances) is underdetermined by type level model.
  • Cannot distinguish whether wealth(jane) causes wealth(jack), wealth (jack) causes wealth(jane) or both (feedback).

➢ Focus on type-level causal relations.

• How? Learn model of Halpern’s database distribution.
• For token-level inference/prediction, convert to undirected model.
The Learn-and-Join Algorithm (AAAI 2010)

- Required: single-table BN learner \(L\). Takes as input \((T,RE,FE)\):
  - Single data table.
  - A set of edge constraints (forbidden/required edges).
- Nodes: Descriptive attributes (e.g. `intelligence(S)`)
  - Boolean relationship nodes (e.g., `Registered(S,C)`).

1. \(\text{RequiredEdges}, \text{ForbiddenEdges} := \text{emptyset}\).
2. For each entity table \(E_i\):
   a) Apply \(L\) to \(E_i\) to obtain BN \(G_i\). For two attributes \(X,Y\) from \(E_i\),
   b) If \(X \rightarrow Y\) in \(G_i\), then \(\text{RequiredEdges} \leftarrow X \rightarrow Y\).
   c) If \(X \rightarrow Y\)\ not in \(G_i\), then \(\text{ForbiddenEdges} \leftarrow X \rightarrow Y\).
3. For each relationship table join (= conjunction) of size \(s = 1,..k\)
   a) Compute \(R\text{table join, join with entity tables} := J_i\).
   b) Apply \(L\) to \((J_i, RE, FE)\) to obtain BN \(G_i\).
   c) Derive additional edge constraints from \(G_i\).
4. Add relationship indicators: If edge \(X \rightarrow Y\) was added when analyzing join \(R_1 \Join R_2 \Join \ldots \Join R_m\), add edges \(R_i \rightarrow Y\).
Phase 1: Entity tables

**Students**

<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>

**Course**

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

BN learner $L$

- $intelligence(S)$
- $ranking(S)$
- $diff(C)$
- $teach-ability(p(C))$
- $popularity(p(C))$
Phase 2: relationship tables

<table>
<thead>
<tr>
<th>S.Name</th>
<th>C.number</th>
<th>grade</th>
<th>satisfaction</th>
<th>intelligence</th>
<th>ranking</th>
<th>Course</th>
<th>difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>101</td>
<td>A</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

BN learner $L$

intelligence(S) → ranking(S)

$\text{diff}(C)$ → teach-ability(p(C))

$\text{rating}(C)$ → popularity(p(C))

grade(S,C) → $\text{diff}(C)$ → teach-ability(p(C))
Phase 3: add Boolean relationship indicator variables

- intelligence(S)
- ranking(S)
- satisfaction(S,C)
- Registered(S,C)
- grade(S,C)
- diff(C)
- rating(C)
- teach-ability(p(C))
- popularity(p(C))
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 + 0.032</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.
- \(x + y = \text{structure learning + parametrization.}\)
- **JBN**: Our join-based algorithm.
- **MLN, CMLN**: standard programs from the U of Washington (Alchemy)
Accuracy

![Bar chart showing accuracy for different datasets: University, MovieLens Subsample 1, MovieLens Subsample 12, Mutagenesis Subsample 1, Mutagenesis Subsample 12. The x-axis represents the datasets, and the y-axis represents accuracy. The chart compares accuracy for JBN, MLN, and CMLN models.]
Pseudo-likelihood for Functor Bayes Nets

- What likelihood function $P(\text{database, graph})$ does the learn-and-join algorithm optimize?

1. Moralize the BN (causal graph).
2. Use the Markov net likelihood function for moralized BN---without the normalization constant.
   - $\prod_{\text{families}} P(\text{child} | \text{parent})^{\#\text{child-parent instances}}$
   - pseudo-likelihood.

![Diagram of Relational Causal Graph, Markov Logic Network, and Likelihood Function]
Features of Pseudo-likelihood P*

✓ Tractability: maximizing estimates = empirical conditional database frequencies!

• Similar to pseudo-likelihood function for Markov nets (Besag 1975, Domingos and Richardson 2007).

• Mathematically equivalent but conceptually different interpretation: expected log-likelihood for randomly selected individuals.
1. Randomly select instances $X_1 = x_1, \ldots, X_n = x_n$, for each variable in BN.
2. Look up their properties, relationships.
3. Compute log-likelihood for the BN assignment obtained from the instances.
4. $L^H = \text{average log-likelihood over uniform random selection of instances.}$

**Proposition** $L^H(D,B) = \ln(P*(D,B) \times c$  
where $c$ is a (meaningful) constant.  
No independence assumptions!
Summary of Review

- Two key conceptual questions for relational causal modelling.
  1. What are the random variables (nodes)?
  2. How to measure fit of model to data?
  1. Nodes = functors, open function terms (Poole).
  2. Instantiate type-level model with all possible tokens. Use instantiated model to assign likelihood to the totality of all token facts.
- Problem: instantiated model may contain cycles even if type-level model does not.
- One solution: use undirected models.
Summary of New Results

New algorithm for learning causal graphs with functors.
+ Fast and scalable (e.g., 5 min vs. 21 hr).
+ Substantial Improvements in Accuracy.

New pseudo-likelihood function for measuring fit of model to data.
- Tractable parameter estimation.
- Similar to Markov network (pseudo)-likelihood.
- New semantics: expected log-likelihood of the properties of randomly selected individuals.
Open Problems

Learning

• Learn-and-Join learns dependencies among attributes, not dependencies among relationships.
• Parameter learning still a bottleneck.

Inference/Prediction

• Markov logic likelihood does not satisfy Halpern’s principle:
  if \( P(\varphi(X)) = p \), then \( P(\varphi(a)) = p \)
  where \( a \) is a constant.
  (Related to Miller’s principle).
• Is this a problem?
Thank you!

- Any questions?
Choice of Functors

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

- In remainder of this talk, use functors corresponding to
  - Attributes (columns), e.g., \( \text{intelligence}(S) \), \( \text{grade}(S,C) \)
  - Boolean Relationship indicators, e.g. \( \text{Friend}(X,Y) \).
Typical Tasks for Statistical-Relational Learning (SRL)

- **Link-based Classification**: given the links of a target entity and the attributes of related entities, predict the class label of the target entity.

- **Link Prediction**: given the attributes of entities and their other links, predict the existence of a link.