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

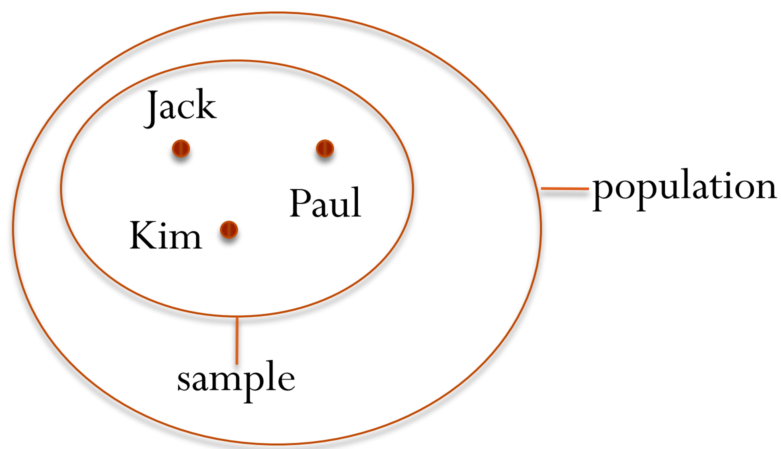
- The Learn-and-Join Bayes Net Learning Algorithm.
- A Pseudo-Likelihood Function for Relational Bayes Nets.

Single Data Table Statistics

Traditional Paradigm Problem

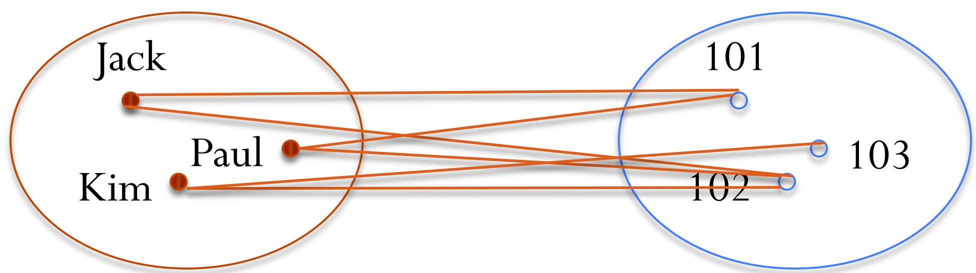
- Single population
- Random variables = attributes of population members.
- “flat” data, can be represented in single table.

Students		
<u>**Name**</u>	**intelligence**	**ranking**
Jack	3	1
Kim	2	1
Paul	1	2



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.

Student

s-id	Intelligence	Ranking
Jack	3	1
Kim	2	1
Paul	1	2

Course

c-id	Rating	Difficulty
101	3	1
102	2	2

Professor

p-id	Popularity	Teaching-a
Oliver	3	1
Jim	2	1

RA

s-id	p-id	Salary	Capability
Jack	Oliver	High	3
Kim	Oliver	Low	1
Paul	Jim	Med	2

Registration

s-id	c.id	Grade	Satisfaction
Jack	101	A	1
Jack	102	B	2
Kim	102	A	1
Paul	101	B	1

Link based Classification

- P(diff(101))?

Student

s-id	Intelligence	Ranking
Jack	3	1
Kim	2	1
Paul		

Course

c-id	Rating	Difficulty
101	3	???

Professor

p-id	Popularity	Teaching-a
Oliver	3	1
		1

RA

s-id	p-id	Salary	Capability
Jack	Oliver	High	3
Kim	Oliver	Low	1
Paul	Jim	Med	2

Registration

s-id	c.id	Grade	Satisfaction
Jack	101	A	1
Jack	102	B	2
Kim	102	A	1
Paul	101	B	1

Link prediction

- P(Registered(jack,101))?

Student

s-id	Intelligence	Ranking
Jack	3	1
Kim	2	1
Paul	1	

Course

c-id	Rating	Difficulty
101	3	1

Professor

p-id	Popularity	Teaching-a
Oliver	3	1
		1

RA

s-id	p-id	Salary	Capability
Jack	Oliver	High	3
Kim	Oliver	Low	1
Paul	Jim	Med	2

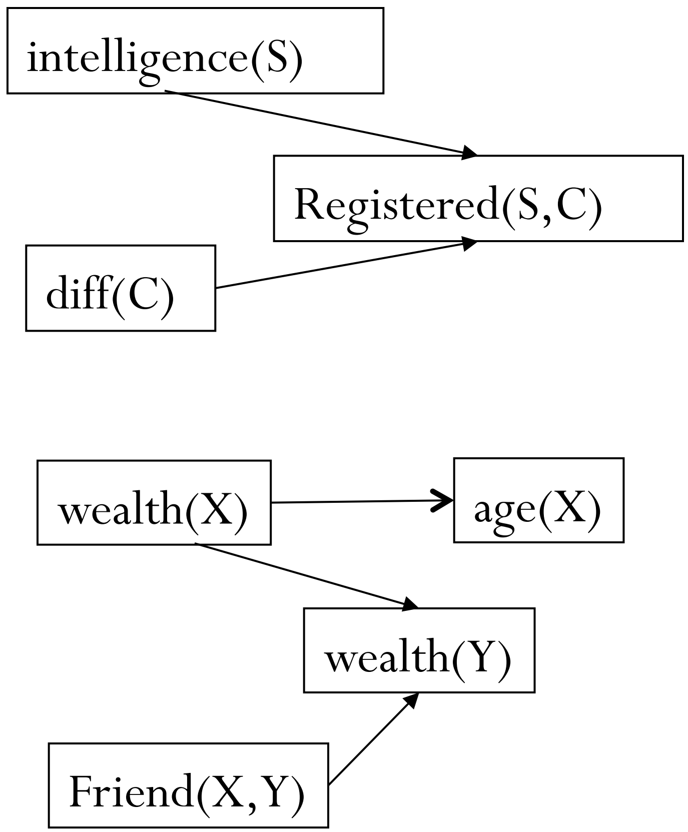
Registration

s-id	c.id	Grade	Satisfaction
Jack	101	A	1
Jack	102	B	2
Kim	102	A	1
Paul	101	B	1

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(child | parents).
- e.g., P(wealth(Y) = T | wealth (X) = T, Friend(X,Y) = T)
- defines joint probability for every conjunction of value assignments.

Domain Semantics of Functors

- Halpern 1990, Bacchus 1990
- Intuitively, $P(Flies(X) \mid Bird(X)) = 90\%$ 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

- $P(S = jack) = 1/3$.
- $P(age(S) = 20) = \Sigma_{s:age(s)=20} 1/|S|$.
- $P(Friend(X,Y) = T) = \Sigma_{x,y:friend(x,y)} 1/(|X||Y|)$.
- 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 ➔ **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.

Student

s-id	Intelligence	Ranking
Jack	3	1
Kim	2	1
Paul	1	[illegible]

Course

c-id	Rating	Difficulty
101	3	1
[illegible]	[illegible]	[illegible]

Professor

p-id	Popularity	Teaching-a
Oliver	3	1
[illegible]	[illegible]	1

RA

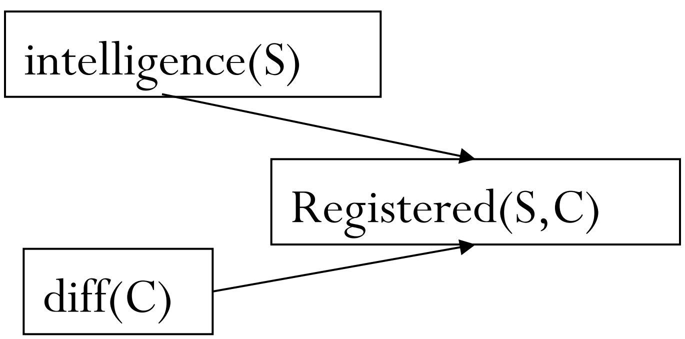
s-id	p-id	Salary	Capability
Jack	Oliver	High	3
Kim	Oliver	Low	1
Paul	Jim	Med	2

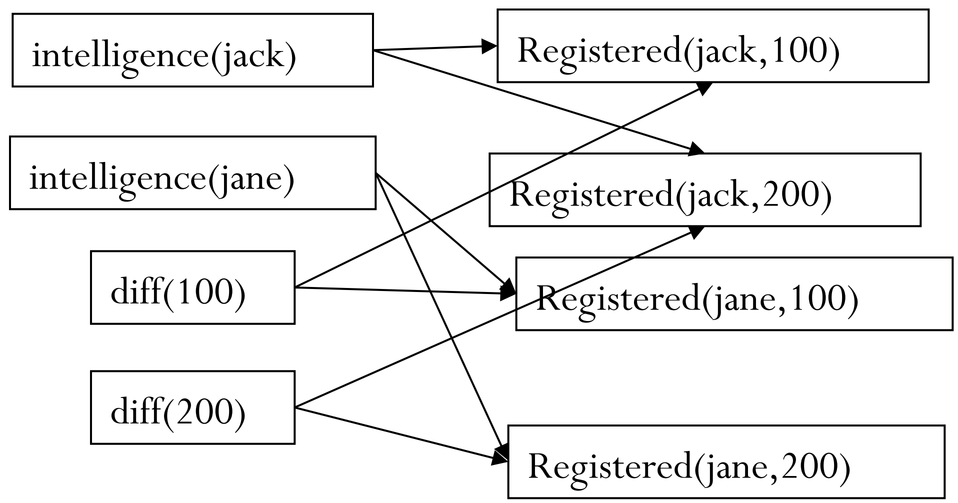
Registration

s-id	c.id	Grade	Satisfaction
Jack	101	A	1
Jack	102	B	2
Kim	102	A	1
Paul	101	B	1

Knowledge-based Model Construction

- Ngo and Haddaway, 1997; Koller and Pfeffer, 1997; Haddaway, 1999.
- 1^{st}-order model = template.
- Instantiate with individuals from database (fixed!) → ground model.
- Isomorphism DB facts ↔ assignment of values → **likelihood measure** for DB.

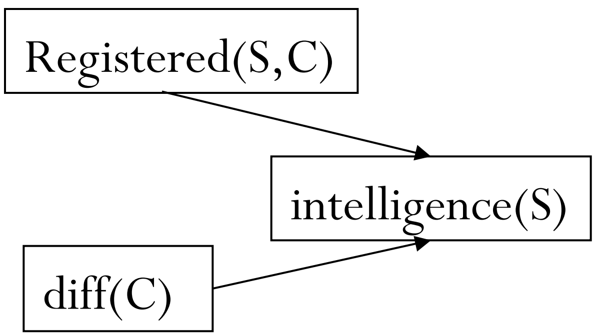


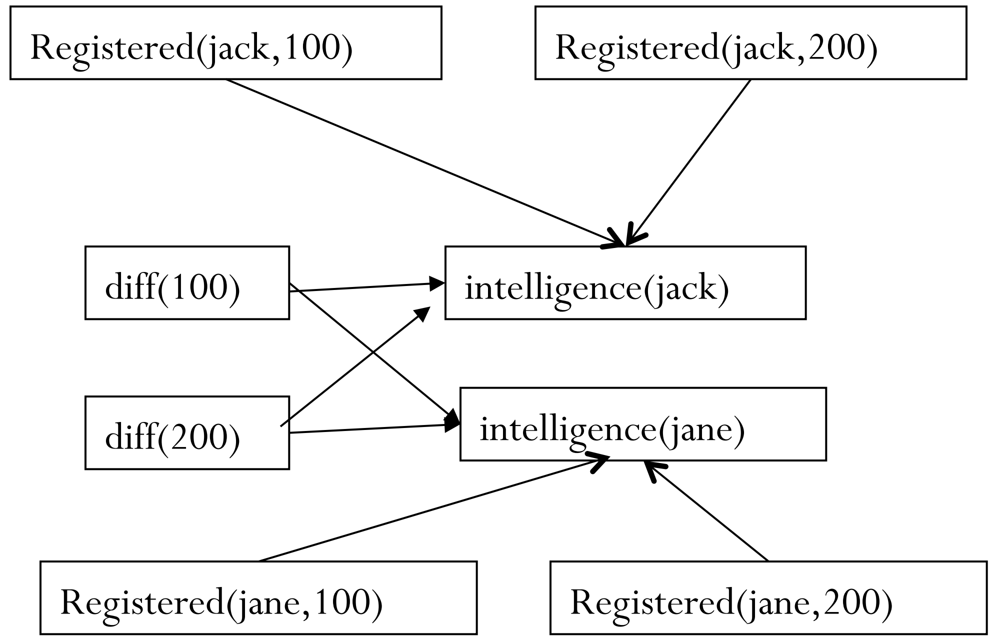


Class-level Template
with 1st-order Variables

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

The Combining Problem

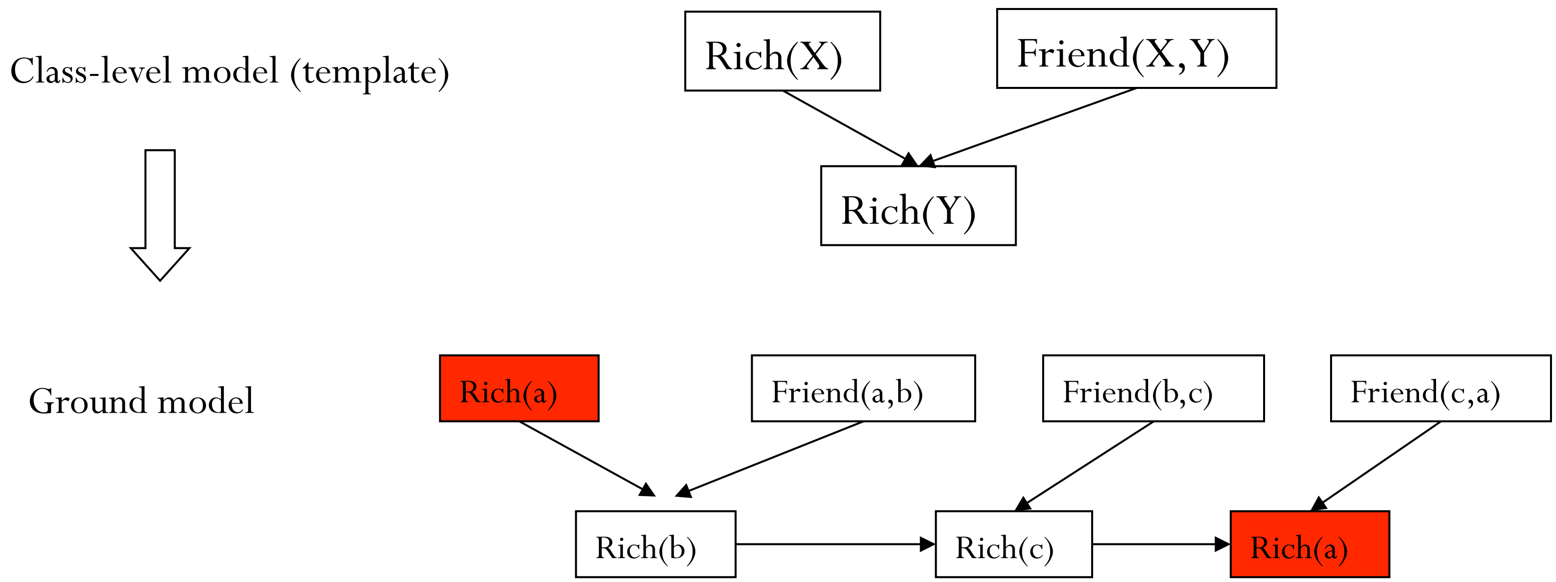




- How do we combine information from different related entities (courses)?

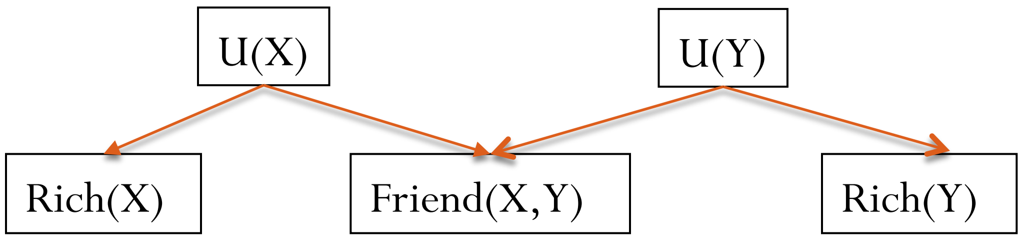
- Aggregate properties of related entities (PRMs; Getoor, Koller, Friedman).
- Combine probabilities. (BLPs; Poole, deRaedt, Kersting.)

The Cyclicity Problem



- 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.”

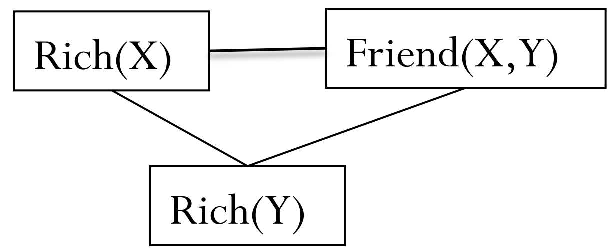
Hidden Variables Avoid Cycles



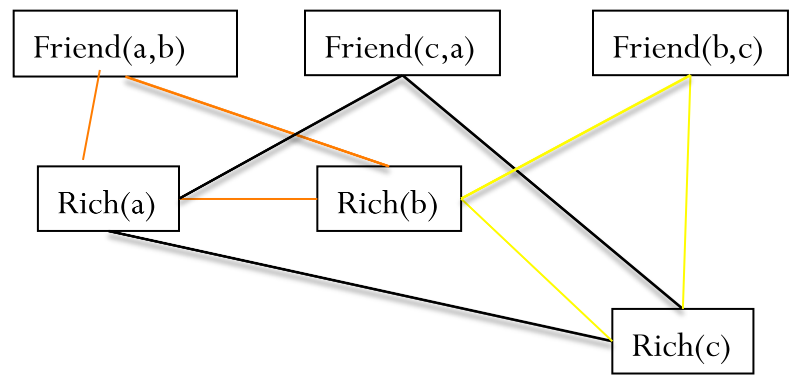
- Assign unobserved values *u(jack), u(jane)*.
- Probability that Jack and Jane are friends depends on their unobserved "type".
- In ground model, *rich(jack)* and *rich(jane)* are correlated given that they are friends, but neither is an ancestor.
- Common in social network analysis (Hoff 2001, Hoff and Rafferty 2003, Fienberg 2009).
- \$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)

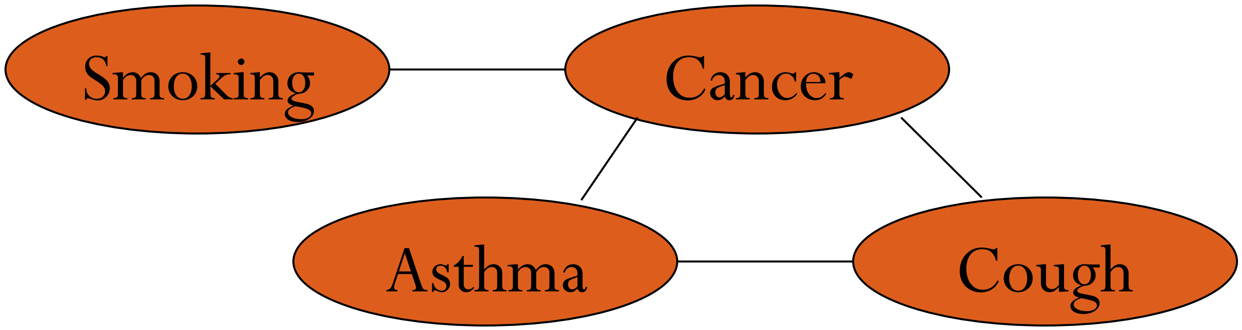


Ground model



Markov Network Example

- **Undirected** graphical model



- Potential functions defined over cliques

$$P(x) = \frac{1}{Z} \prod_c \Phi_c(x_c)$$

$$Z = \sum_x \prod_c \Phi_c(x_c)$$

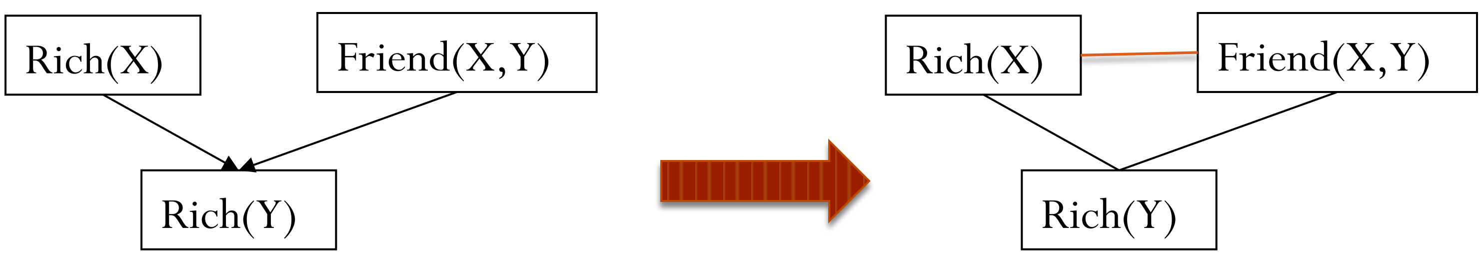
Smoking	Cancer	Φ(S,C)
False	False	4.5
False	True	4.5
True	False	2.7
True	True	4.5

Markov Logic Networks

- Domingos and Richardson ML 2006
- An MLN is a set of formulas with weights.
- Graphically, a Markov network with functor nodes.

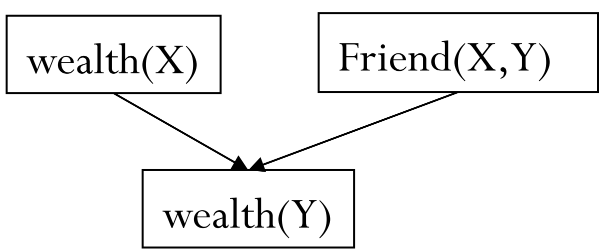
☑ **Solves the combining and the cyclicity problems.**

- For every functor BN, there is a predictively equivalent MLN (the moralized BN).



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)

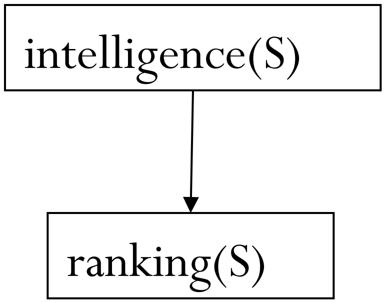
- Required: single-table BN learner L. Takes as input *(T,RE,FE)*:
  - Single data table.
  - A set of edge constraints (forbidden/required edges).
- Nodes: Descriptive attributes (e.g. *intelligence(S))*
  Boolean relationship nodes (e.g., *Registered(S,C))*.

1. RequiredEdges, ForbiddenEdges := 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 RequiredEdges += $X \rightarrow Y$.
   c) If $X \rightarrow Y$ *not* in G_i, then ForbiddenEdges += $X \rightarrow Y$.
3. For each relationship table join (= conjunction) of size $s = 1,..k$
   a) Compute Rtable 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* R_m, add edges $R_i \rightarrow Y$.

Phase 1: Entity tables

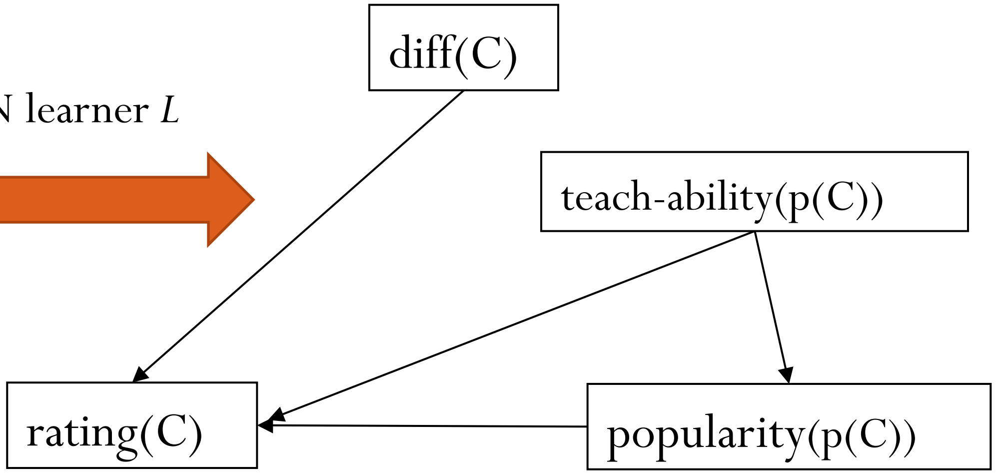
Students		
Name	**intelligence**	**ranking**
Jack	3	1
Kim	2	1
Paul	1	2

BN learner *L*



Course			
Number	**Prof**	**rating**	**difficulty**
101	Oliver	3	1
102	David	2	2
103	Oliver	3	2

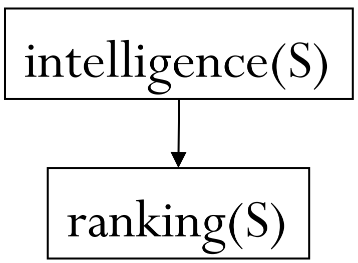
BN learner *L*

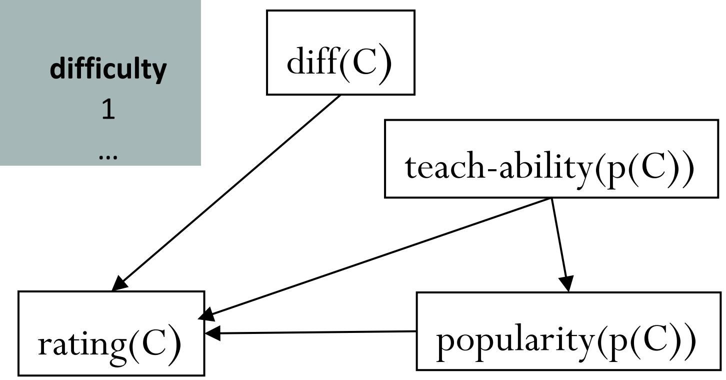


Phase 2: relationship tables

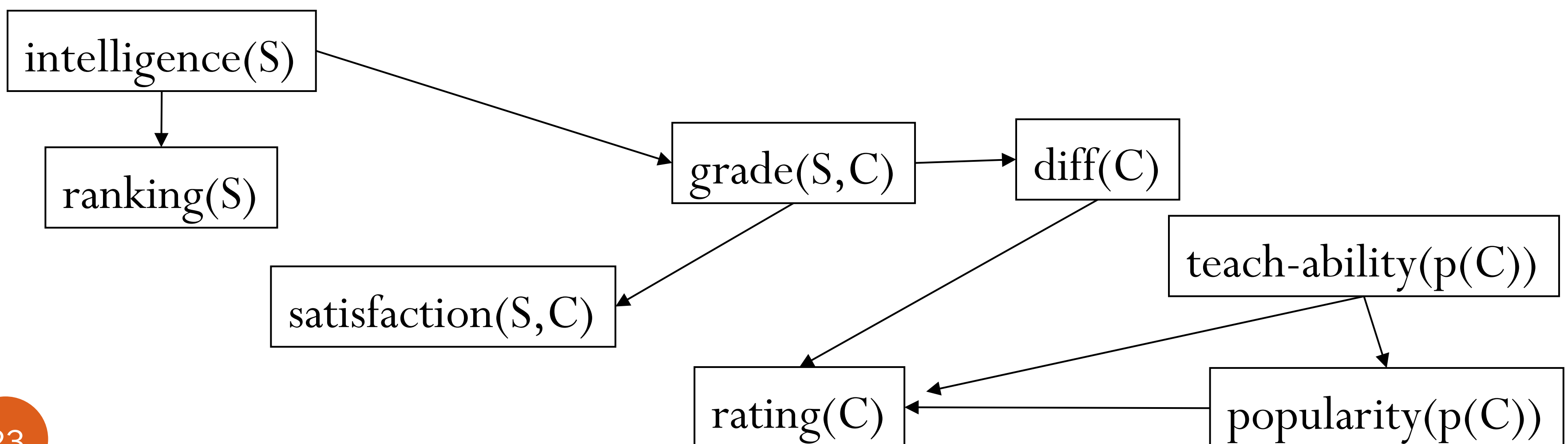
Registration

S.Name	**C.number**	**grade**	**satisfaction**	Student **intelligence**	**ranking**	Course **rating**	**difficulty**
Jack	101	A	1	3	1	3	1
….	….	…	…	…	…	…	…

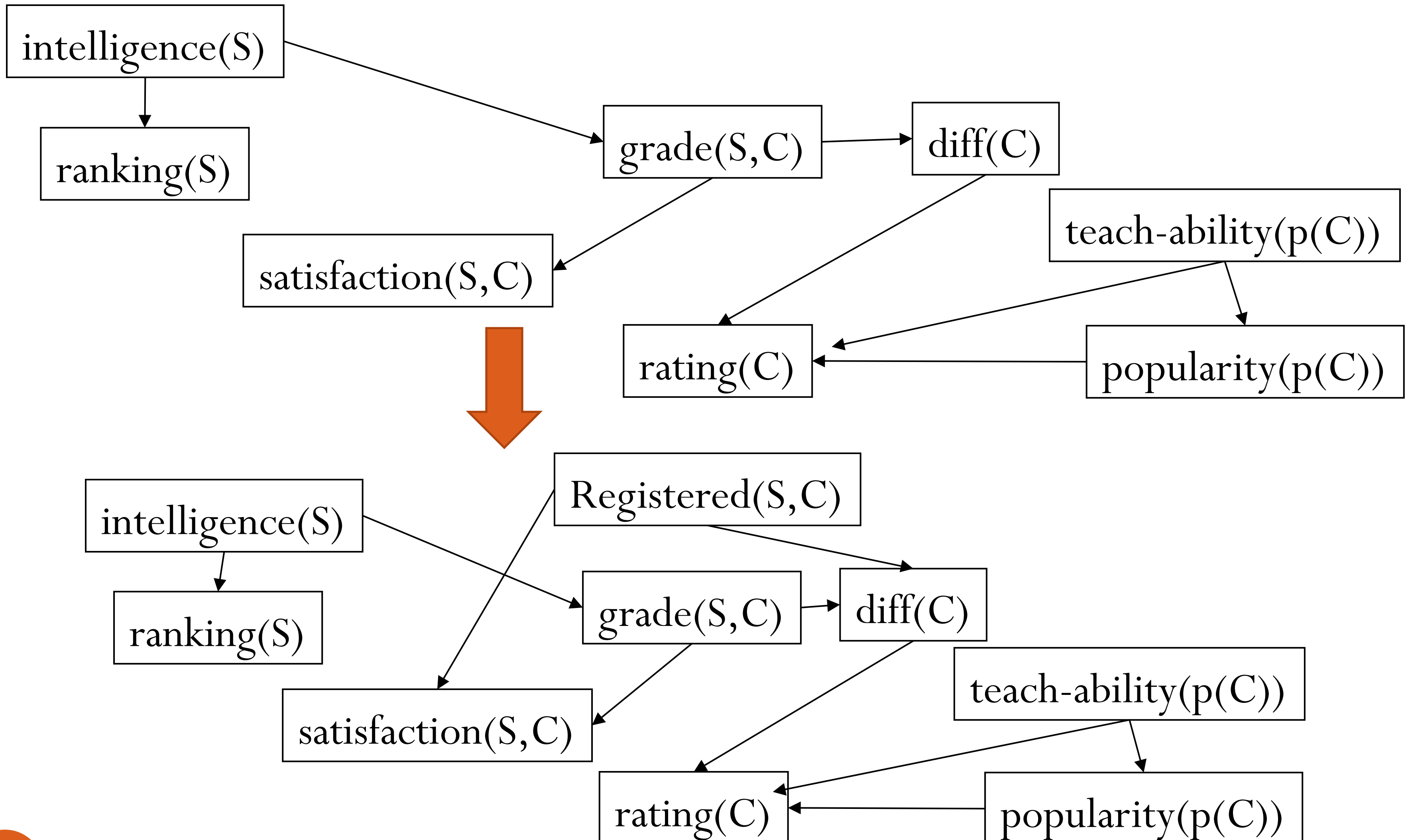




BN learner *L*



Phase 3: add Boolean relationship indicator variables

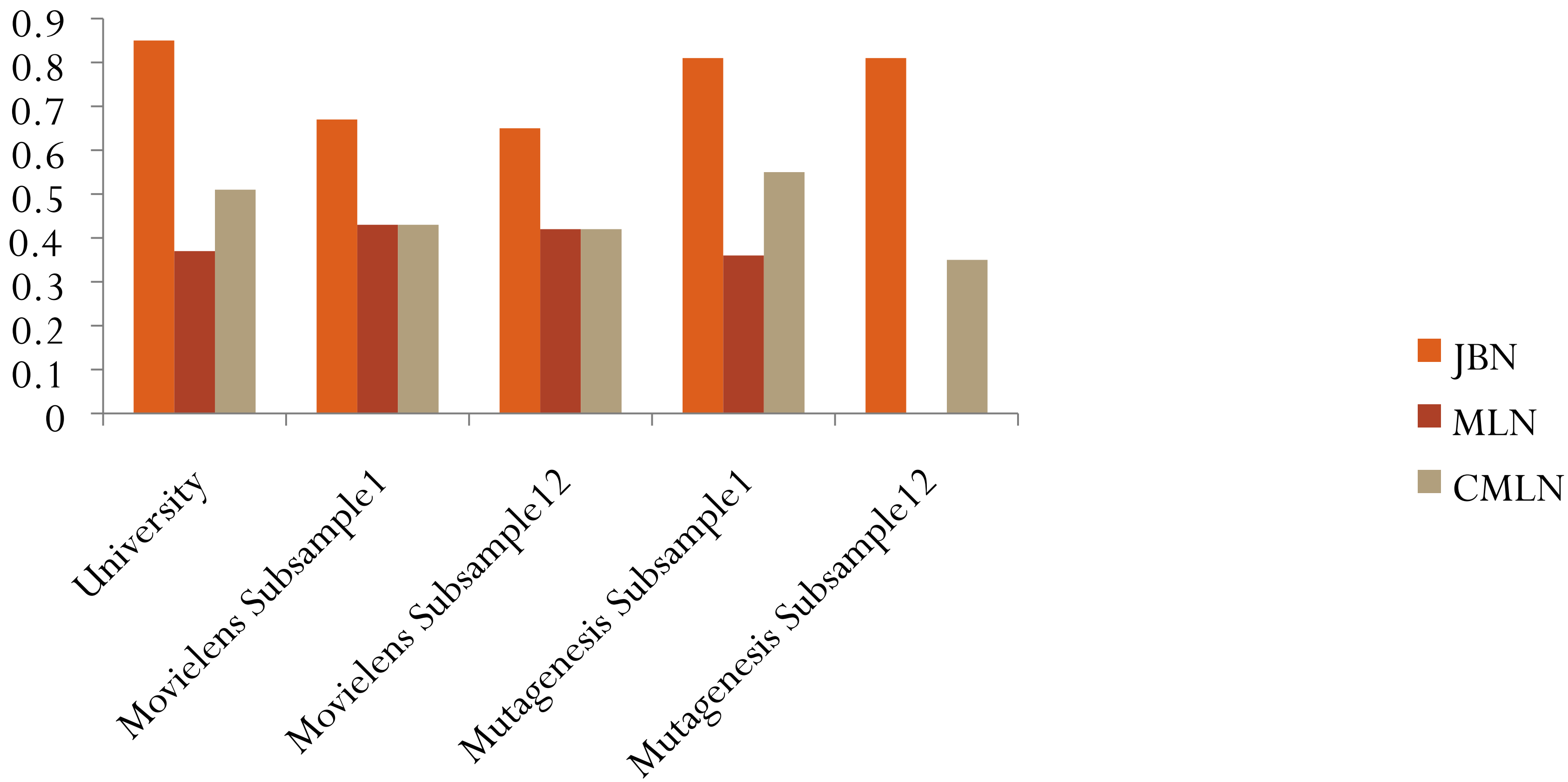


Running time on benchmarks

Dataset	JBN	MLN	CMLN
University	0.03+0.032	5.02	11.44
MovieLens	1.2+120	NT	NT
MovieLens Subsample 1	0.05 + 0.33	44	121.5
MovieLens Subsample 2	0.12 + 5.10	2760	1286
Mutagenesis	0.5 +NT	NT	NT
Mutagenesis subsample 1	0.1 + 5	3360	900
Mutagenesis subsample 2	0.2 +12	NT	3120

- Time in Minutes. NT = did not terminate.
- x + y = structure learning + parametrization.
- JBN: Our join-based algorithm.
- MLN, CMLN: standard programs from the U of Washington (Alchemy)

Accuracy


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
University
Movielens Subsample1
Movielens Subsample12
Mutagenesis Subsample1
Mutagenesis Subsample12
JBN
MLN
CMLN

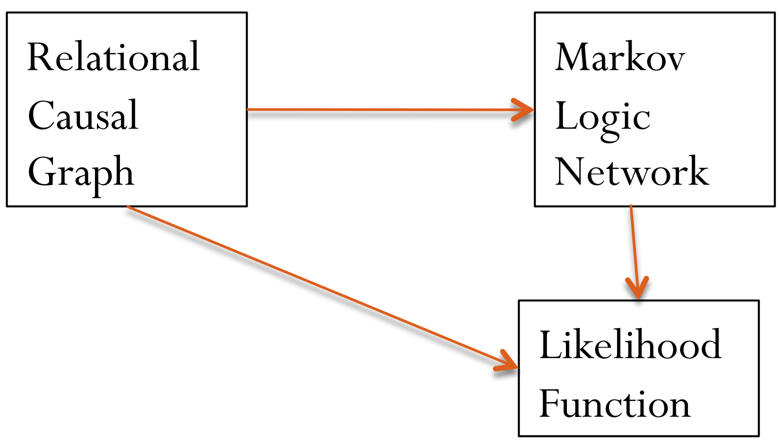
Pseudo-likelihood for Functor Bayes Nets

- What likelihood function P(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.*

➢ $\Pi_{families} \cdot P(child \mid parent)^{\#child\text{-}parent\ instances}$

➢ pseudo-likelihood.

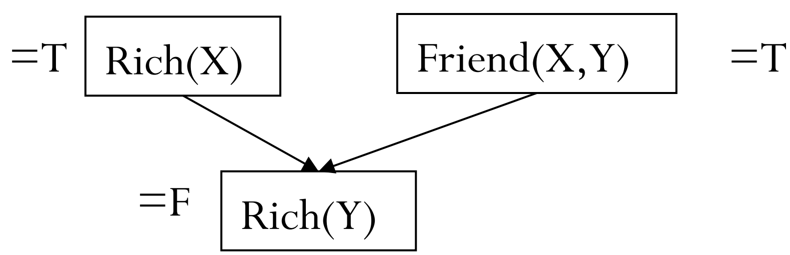


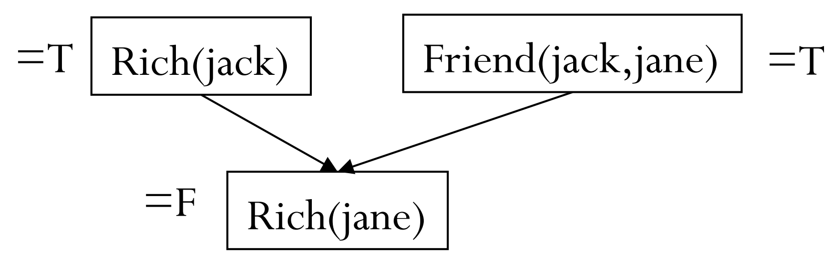
Features of Pseudo-likelihood P*

- ✓ 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.

Halpern Semantics for Functor Bayes Nets (new)

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 = average log-likelihood over uniform random selection of instances.





Proposition $L^H(D,B) = ln(P^*(D,B)\ x\ c$

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

<u>New algorithm for learning causal graphs with functors.</u>

+ Fast and scalable (e.g., 5 min vs. 21 hr).
+ Substantial Improvements in Accuracy.

<u>New pseudo-likelihood function for measuring fit of model to data.</u>

- Tractable parameter estimation.
- Similar to Markov network (pseudo)-likelihood.
- New semantics: expected log-likelihood of the properties of randomly selected individuals.

Open Problems

<u>Learning</u>

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

<u>Inference/Prediction</u>

- 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: *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)*.

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.