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

Oliver Schulte

School of Computing Science

Simon Fraser University

Vancouver, Canada

SFU

Machine Learning for Relational Databases

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

Measuring Model Fit

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).
3. *Pseudo-likelihood function for 1+2.*
4. *Random Selection Semantics.*
5. Parameter Learning.
6. Structure Learning.

Database Instance based on Entity-Relationship (ER) Model

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

Professor		
<u>Name</u>	popularity	teaching Ability
Oliver	3	1
David	2	1

Registration			
<u>S.name</u>	<u>C.number</u>	grade	satisfaction
Jack	101	A	1
Jack	102	B	2
Kim	102	A	1
Kim	103	A	1
Paul	101	B	1
Paul	102	C	2

Course			
<u>Number</u>	Prof	rating	difficulty
101	Oliver	3	1
102	David	2	2
103	Oliver	3	2

<u>Key fields</u> are underlined.

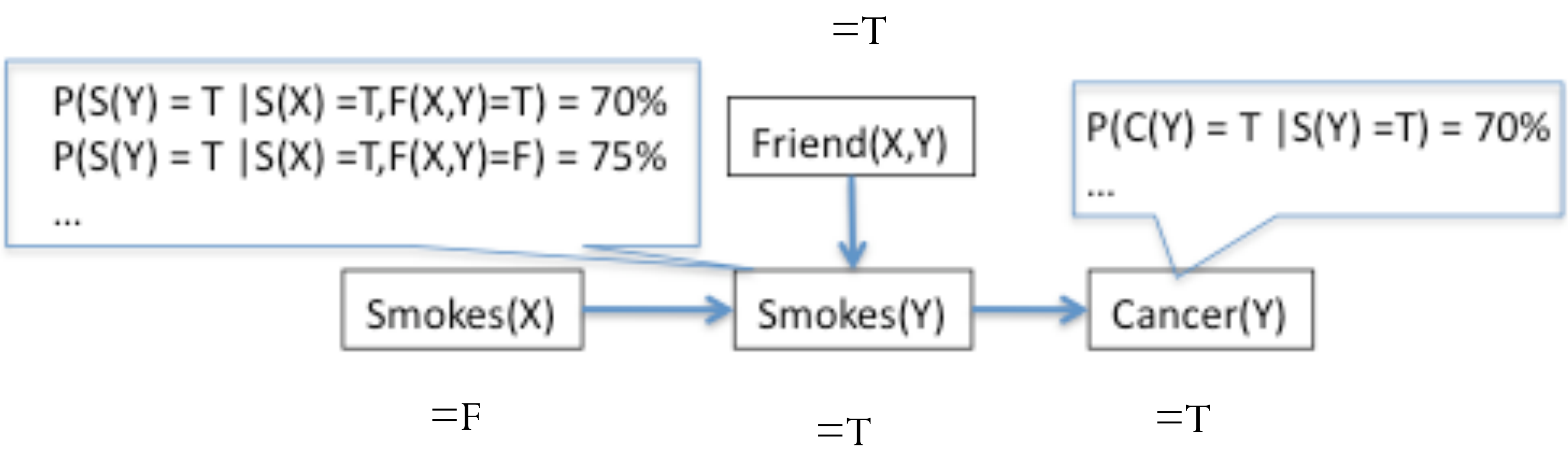
Nonkey fields are deterministic **functions of key fields**.

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 1st-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).

*David Poole, "First-Order Probabilistic Inference", IJCAI 2003.
Originally: Parametrized Bayes Net.

Example: Functor Bayes Nets

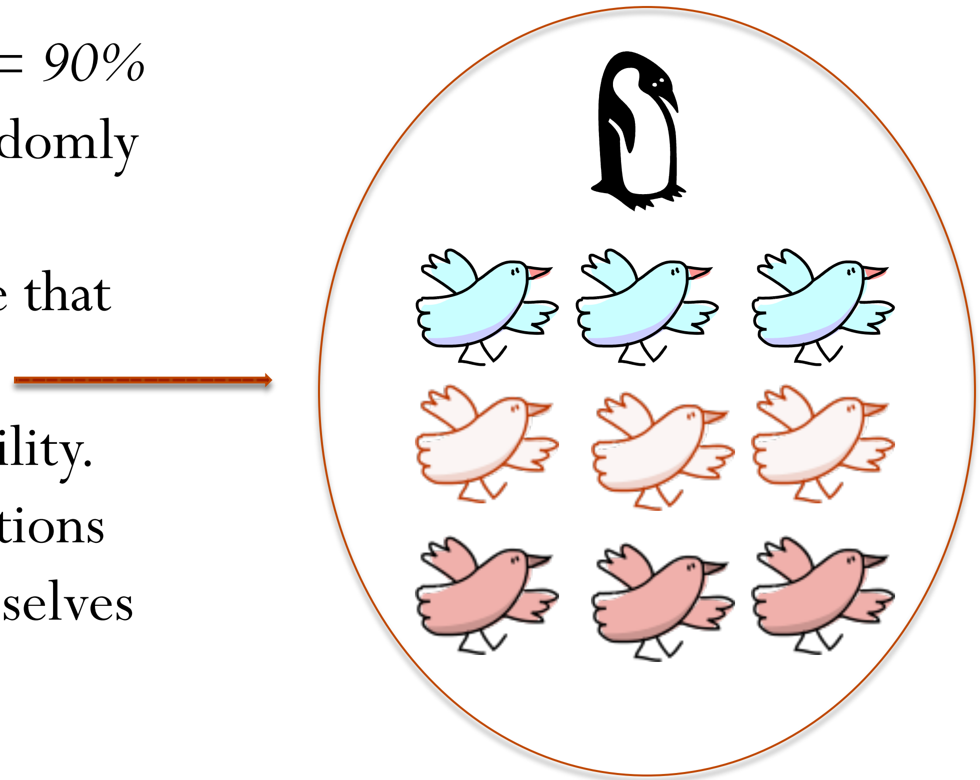


- Parameters: conditional probabilities *P(child|parents)*.
- Defines joint probability for every conjunction of value assignments.

What is the <u>interpretation</u> of the joint probability?

Random Selection Semantics of Functors

- Intuitively, $P(Flies(X) \mid Bird(X)) = 90\%$ 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.



Halpern, "An analysis of first-order logics of probability", AI Journal 1990.
Bacchus, "Representing and reasoning with probabilistic knowledge", MIT Press 1990.

Random Selection Semantics: Examples

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

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

Users

Name	Smokes	Cancer
Anna	T	T
Bob	T	F

Friend

Name1	Name2
Anna	Bob
Bob	Anna

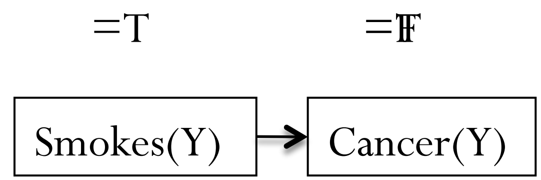
Likelihood Function for Single-Table Data

decomposed (local) data log-likelihood

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

Table *T* count of co-occurrences of child node value and parent state

Parameter of Bayes net *B*



Users

Name	Smokes	Cancer	P_B	$\ln(P_B)$
Anna	T	T	0.36	-1.02
Bob	T	F	0.14	-1.96

	Π≈ 0.05	Σ= -2.98
Likelihood/Log-likelihood	$P(T\|B)$	$\ln P(T\|B)$

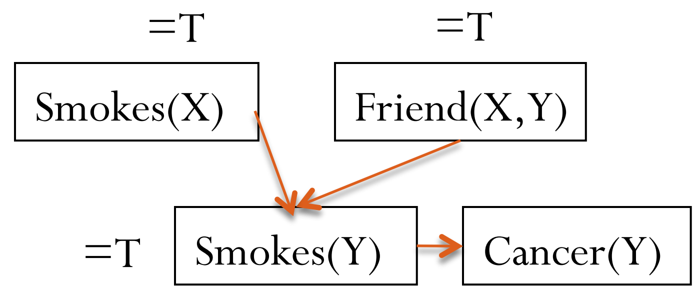
Proposed Pseudo Log-Likelihood

For database D:

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

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

Parameter of Bayes net



Users

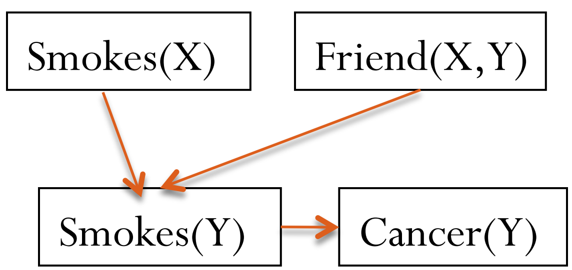
Name	Smokes	Cancer
Anna	T	T
Bob	T	F

Friend

Name1	Name2
Anna	Bob
Bob	Anna

Semantics: Random Selection Log-Likelihood

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



	Hyperentity		Hyperfeatures					
Γ	X	Y	F(X,Y)	S(X)	S(Y)	C(Y)	P_B^γ	$ln(P_B^\gamma)$
γ_1	Anna	Bob	T	T	T	F	0.105	-2.254
γ_2	Bob	Anna	T	T	T	T	0.245	-1.406
γ_3	Anna	Anna	F	T	T	T	0.263	-1.338
γ_4	Bob	Bob	F	T	T	F	0.113	-2.185

$$L^R = -(2.254+1.406+1.338+2.185)/4 \approx -1.8$$

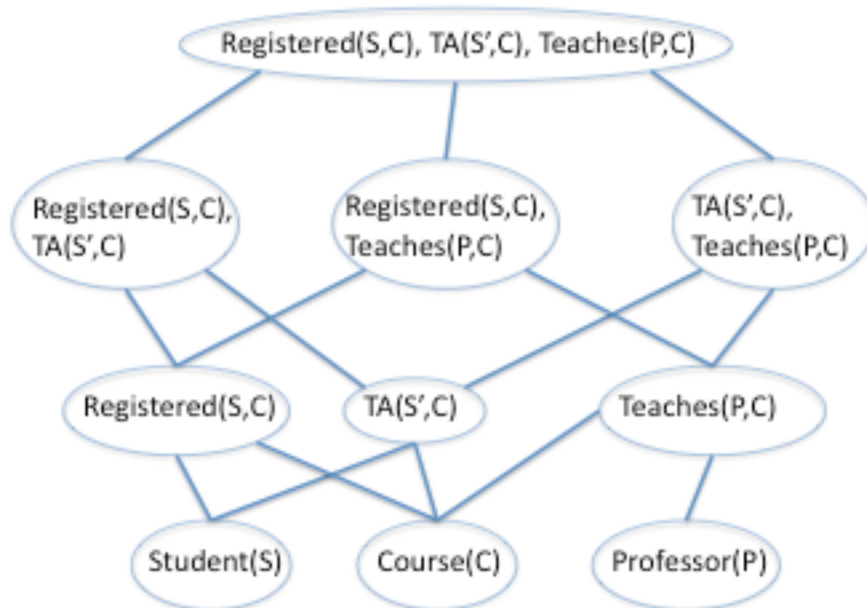
Proposition The random selection log-likelihood equals the pseudo log-likelihood.

Parameter Learning Is Tractable

Proposition For a given database D, the parameter values that maximize the pseudo likelihood are the empirical conditional frequencies in the database.

Structure Learning

- 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).

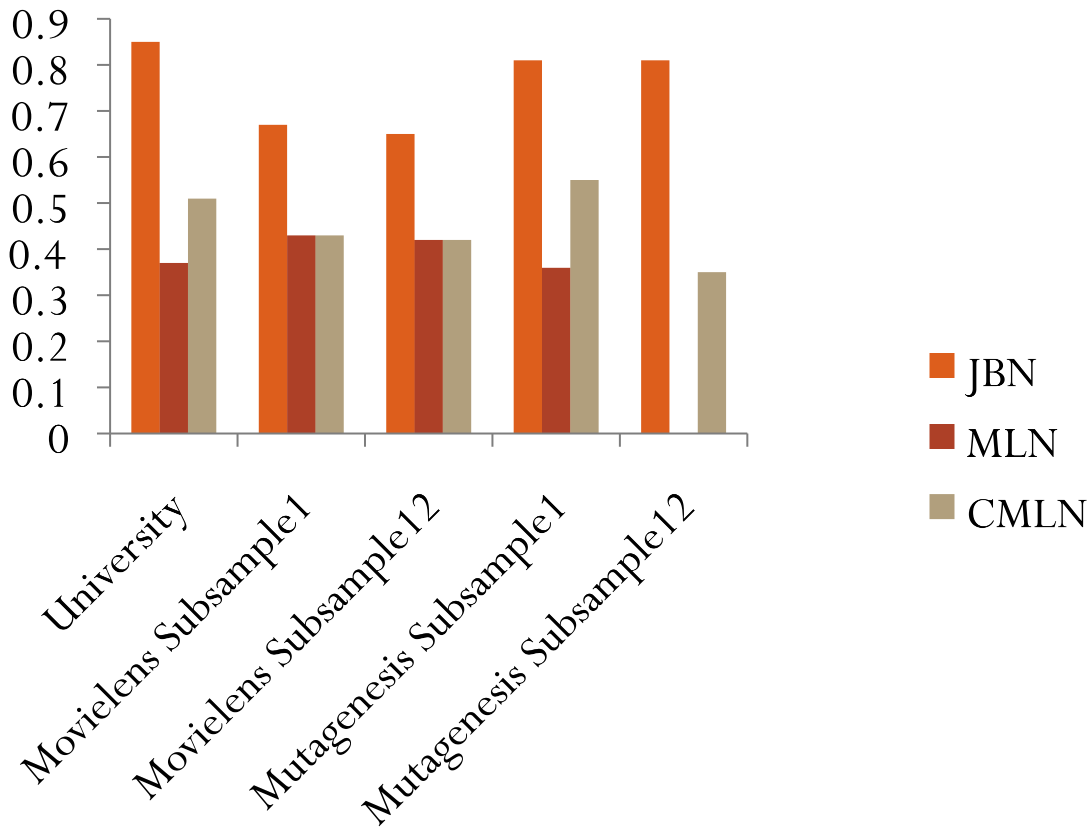


Running time on benchmarks

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 (with Markov net methods).
- JBN: Our join-based algorithm.
- MLN, CMLN: standard programs from the U of Washington (Alchemy)

Accuracy



- 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?

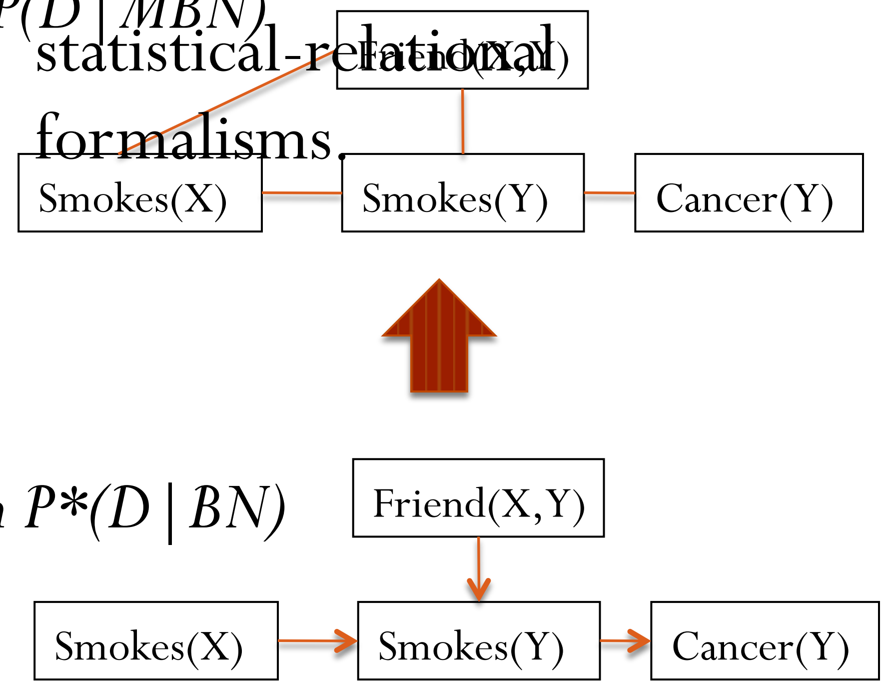
Comparison With Markov Logic Networks (MLNs)

- 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)*

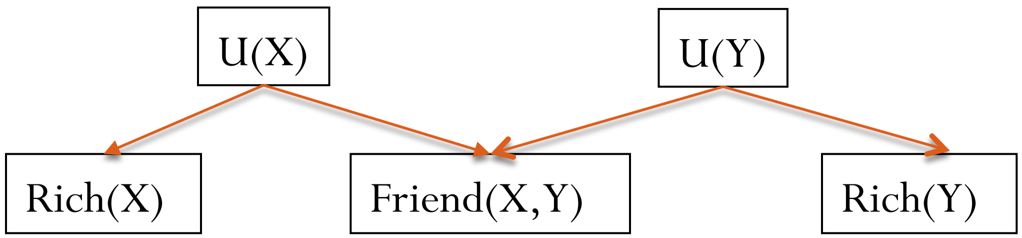
$$\ln(P(D \mid MBN) = \ln P^*(D \mid BN) + \ln(Z)$$

"Markov Logic: An Interface Layer for Artificial Intelligence". Domingos and Lowd 2009.

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----Netflix 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.
  3. *Undirected* models (Taskar, Abeel, Koller UAI 2002, Domingos and Richardson ML 2006).

Hidden Variables Avoid Cycles

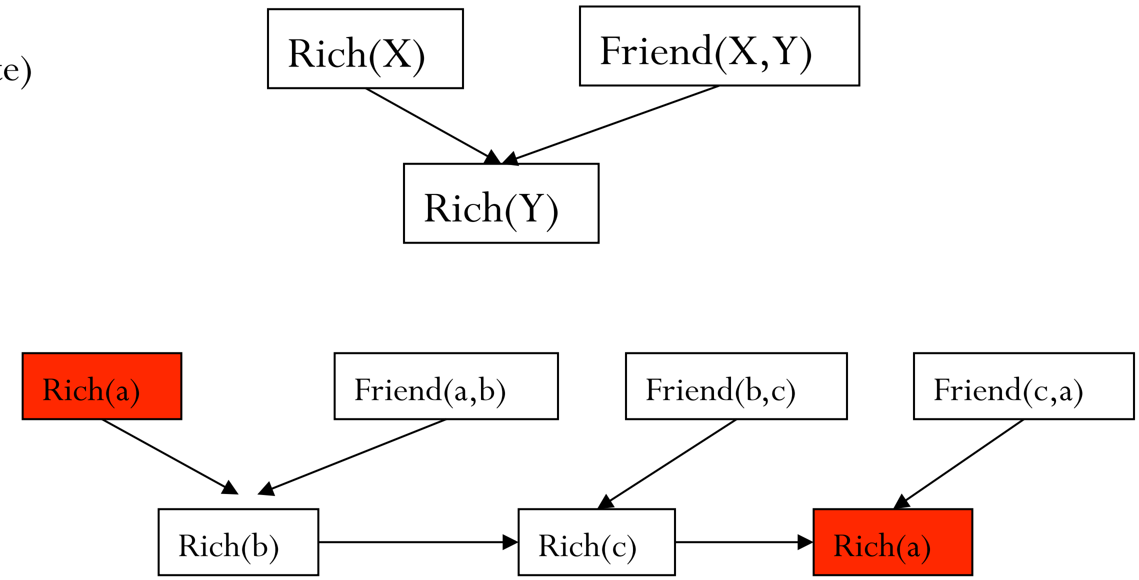


- Assign unobserved values *u(jack), u(jane)*.
- Probability that Jack and Jane are friends depends on their unobserved "type".
- In ground model, *rich(jack)* and *rich(jane)* are correlated given that they are friends, but neither is an ancestor.
- Common in social network analysis (Hoff 2001, Hoff and Rafferty 2003, Fienberg 2009).
- $1M prize in Netflix challenge.
- Also for multiple types of relationships (Kersting et al. 2009).
- Computationally demanding.

The Cyclicity Problem

Class-level model (template)

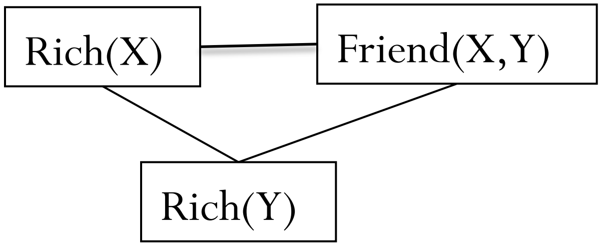
Ground model



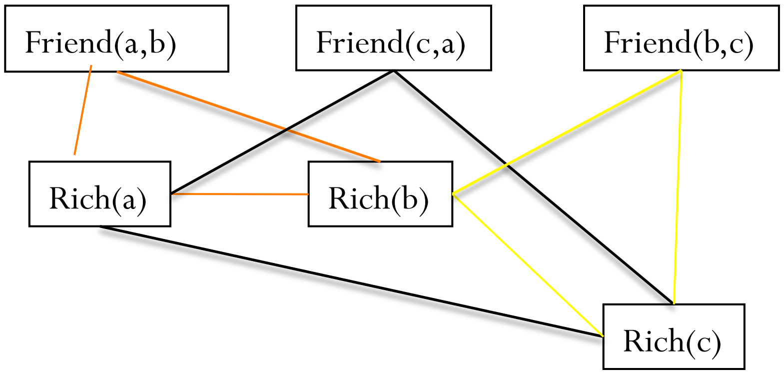
- With recursive relationships, get cycles in ground model even if none in 1st-order model.
- Jensen and Neville 2007: "The acyclicity constraints of directed models severely constrain their applicability to relational data."

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)*.