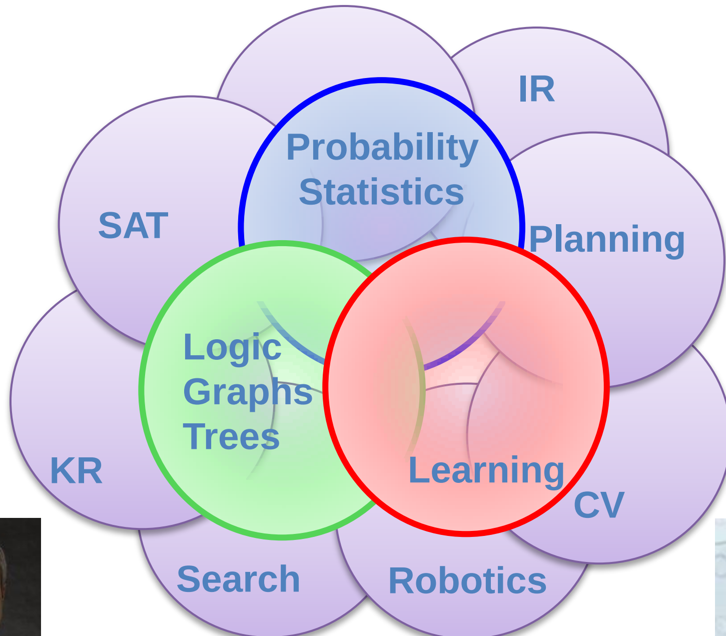
Combining Logic and Probability

Languages, Algorithms and Applications



Pedro Domingos

University of Washington, USA

Kristian Kersting

Fraunhofer IAIS & Univ. of Bonn, Germany

uai2011
Conference on Uncertainty in Artificial Intelligence
Barcelona, Spain
July 14-17, 2011

Acknowledgements

- Statistical Relational Learning (SRL) and AI (StarAI) are a synthesis of ideas of many individuals who have participated in various SRL/StarAI events, workshops and classes.

- Thanks to all of you!

General Take-Away Message

- **Graphs are not enough**
- **We need logic**

Roadmap

1. Motivation
2. Statistical Relational Learning / AI: a short overview
3. Markov Logic Networks

MOTIVATION

[Hermann Rorschach (* Nov 8, 1884; † April 2 1922)]

Rorschach Test

Etzioni's Rorschach Test for Computer Scientists

Moore’s Law?

uai2011
Conference on Uncertainty in Artificial Intelligence
Barcelona, Spain
July 14-17, 2011

Storage Capacity?

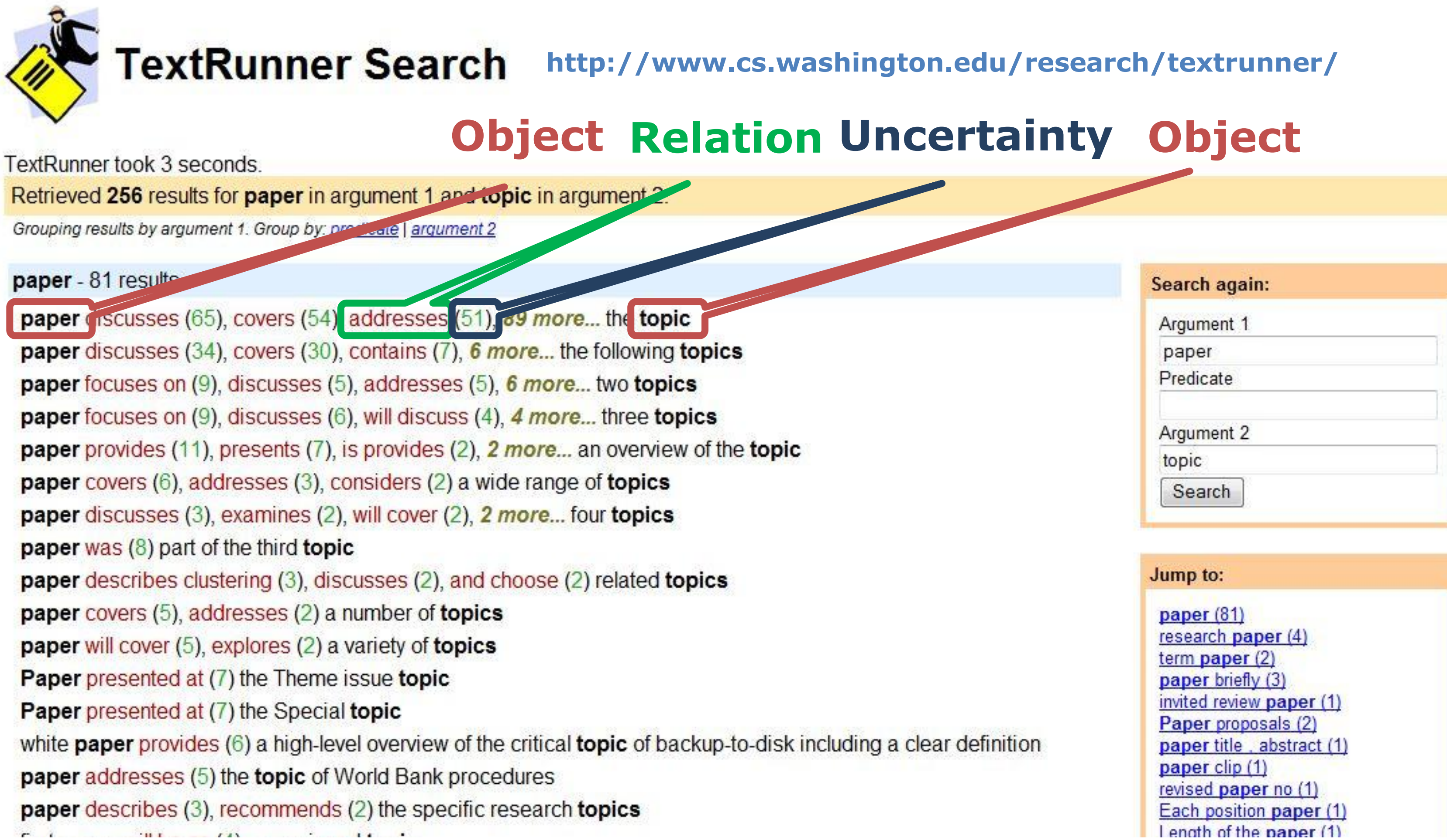
Number of Scientific Publications?

Number of Facebook Users?

Number of Web Pages?

[Weikum, Kasneci, Ramanath, Suchanek Commun. ACM 52(4): 56-64 (2009)]

The World-Wide Mind



So, Tasks Are Often Structural

- Objects are not just feature vectors
  - They have parts and subparts
  - Which have relations with each other
  - They can be trees, graphs, etc.
- Objects are seldom i.i.d. (independent and identically distributed)
  - They exhibit local and global dependencies
  - They form class hierarchies (with multiple inheritance)
  - Objects' properties depend on those of related objects
- Deeply interwoven with knowledge

How do computer systems deal with structural problems?

(First-order) Logic handles Structures

- Main theoretical foundation of computer science
- General language for describing complex structures and knowledge: trees, graphs, hierarchies, etc.
- Inference algorithms (satisfiability testing, resolution, theorem proving, etc.)

More compact knowledge representation. Consider e.g. classicial examples such as chess or wumpus:
FOL << PL << atomic

$\forall$ *x,y father-of(x,y)* $\wedge$ *female(y)*
$\Leftrightarrow$ *daughter-of(y,x)*

daugther-of(cecily,john)
daugther-of(lily,tom)
…

Explicit enumeration

Many types of entities
Relations between them
Arbitrary knowledge

Logic
true/false

5^{th} C B.C.

19^{th} C

atomic **propositional** **first-order/relational**

Tasks are also often Statistical

- Information are ambiguous
- Our information is always incomplete
- Our predictions are uncertain

How do computer systems deal with uncertainty?

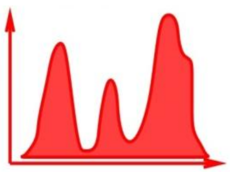
Probability handles Uncertainty

Mixture models
Hidden Markov models
Bayesian networks
Markov random fields
Maximum entropy models
Conditional random fields
…

Probability

17th C

20th C



Sensor noise
Human error
Inconsistencies
Unpredictability

Explicit enumeration

Many types of entities
Relations between them
Arbitrary knowledge

Logic
true/false

5th C B.C.

19th C

atomic **propositional** **first-order/relational**

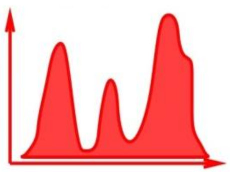
[slide inspired by Russell]

So, will traditional (U)AI scale ?

17th C

20th C

Probability



Sensor noise
Human error
Inconsistencies
Unpredictability

Explicit enumeration

Many types of entities
Relations between them
Arbitrary knowledge

Logic
true/false

5th C B.C.

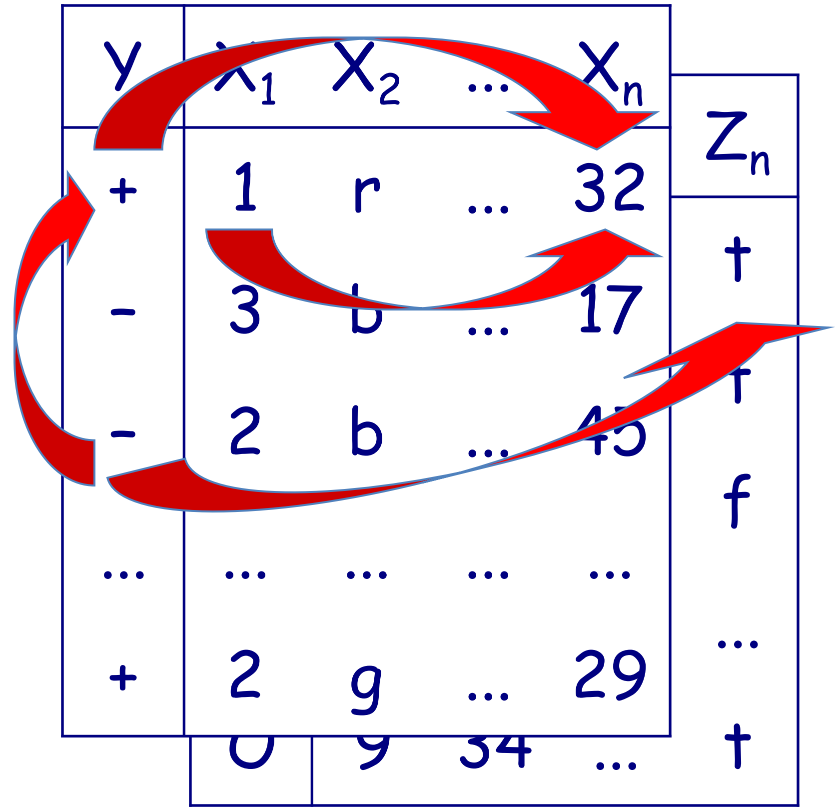
19th C

atomic | propositional | first-order/relational

uai2011
Conference on Uncertainty in Artificial Intelligence
Barcelona, Spain
July 14-17, 2011

Propositional vs. Relational Data

- Traditional work in robotics, machine learning and knowledge discovery assume data instances form a single table.



- Traditional statistical models assume independence among instances (rows) and find associations among the values of multiple variables within a single instance.

- Relational models assume *dependence* among instances in *different* rows and tables and find associations among these values.

[slide adapted from David Jensen]

uai2011
Conference on Uncertainty in Artificial Intelligence
Barcelona, Spain
July 14-17, 2011

[inspired by Friedman and Koller]

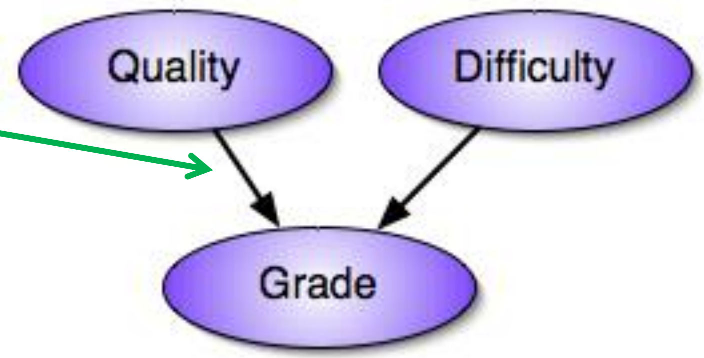
Let’s consider a simple relational domain: Reviewing Papers

- The grade of a paper at a conference depends on the paper’s quality and the difficulty of the conference.
  - **Good papers may get A’s at easy conferences**
  - **Good papers may get D’s at top conference**
  - **Weak papers may get B’s at good conferences**
  - **…**

(Reviewing) Bayesian Network

Random Variables

Direct Influence



P(Qual)		
low	middle	high
0.3	0.5	0.2

P(Diff)		
low	middle	high
0.2	0.3	0.5

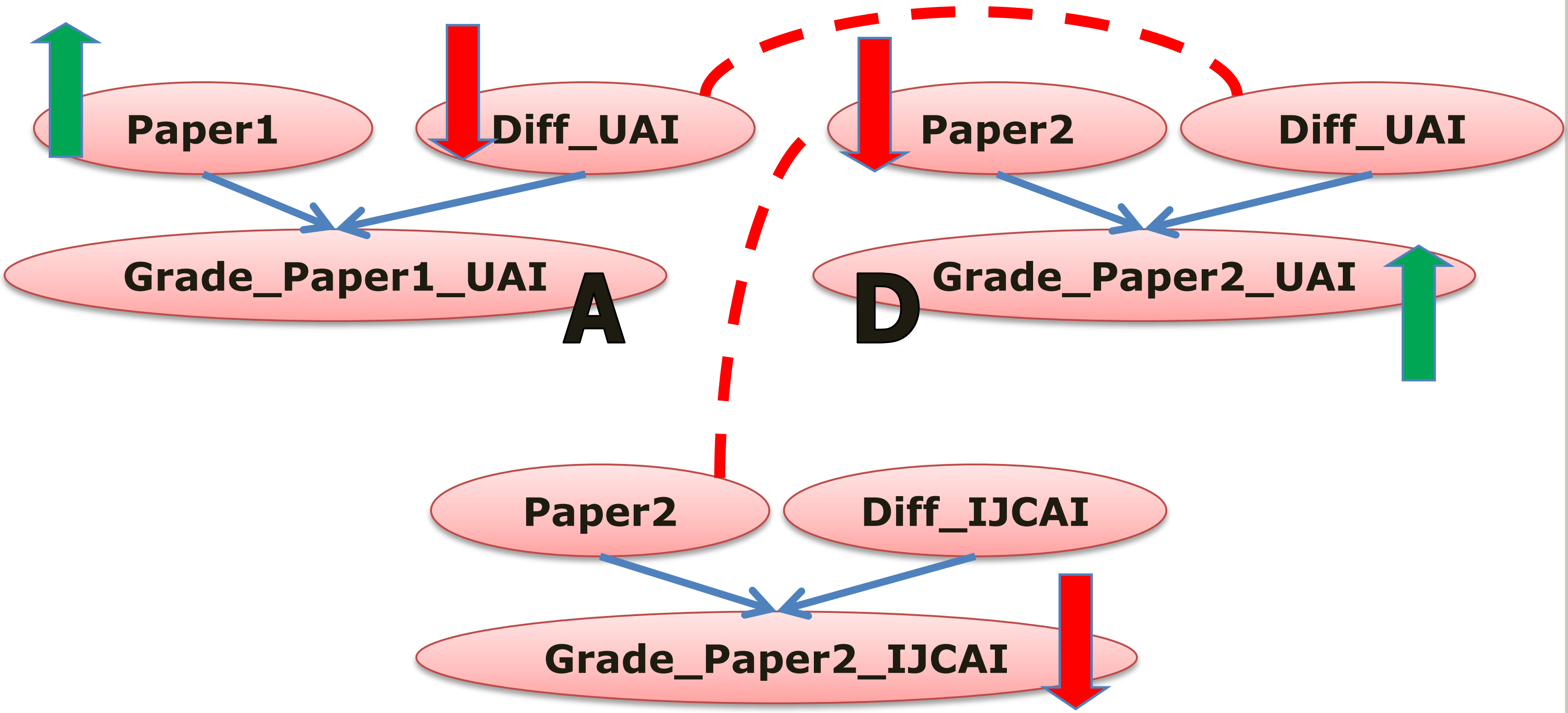
$$P(X_1,\ldots,X_n)=\prod_{i=1}^{n} P(X_i \mid X_{i-1},\ldots,X_1)$$

Qual	Diff	P(Grade) c	b	a
low	low	0.2	0.5	0.3
low	middle	0.1	0.7	0.2
...				

The real world, however, ...

... has inter-related objects

These 'instance' are not independent !

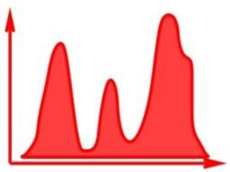


So, will traditional (U)AI scale ? No !

"Scaling up the environment will inevitably overtax the resources of the traditional (U)AI architecture."

MIND THE GAP

Probability



17th C

20th C

Sensor noise
Human error
Inconsistencies
Unpredictability

Explicit enumeration

Many types of entities
Relations between them
Arbitrary knowledge

Logic
true/false

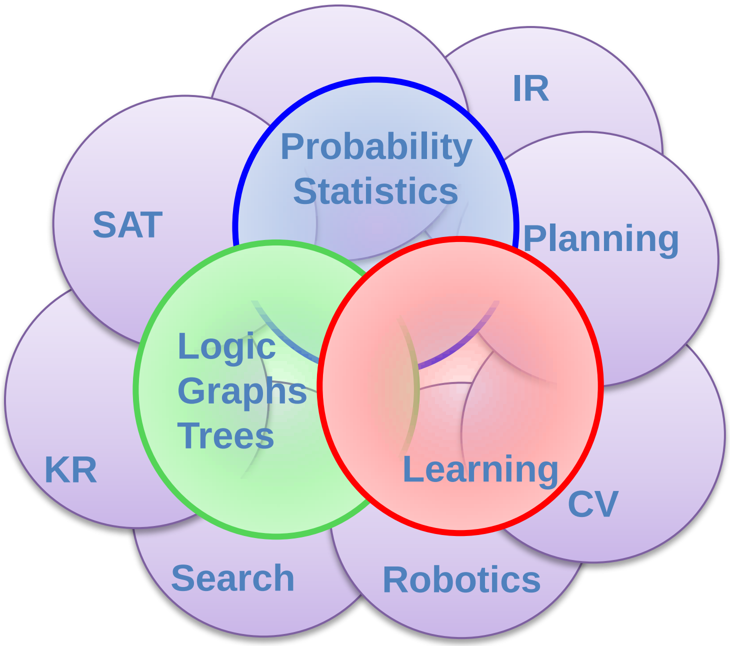
5th C B.C.

19th C

atomic — propositional — first-order/relational

Statistical Relational Learning and AI

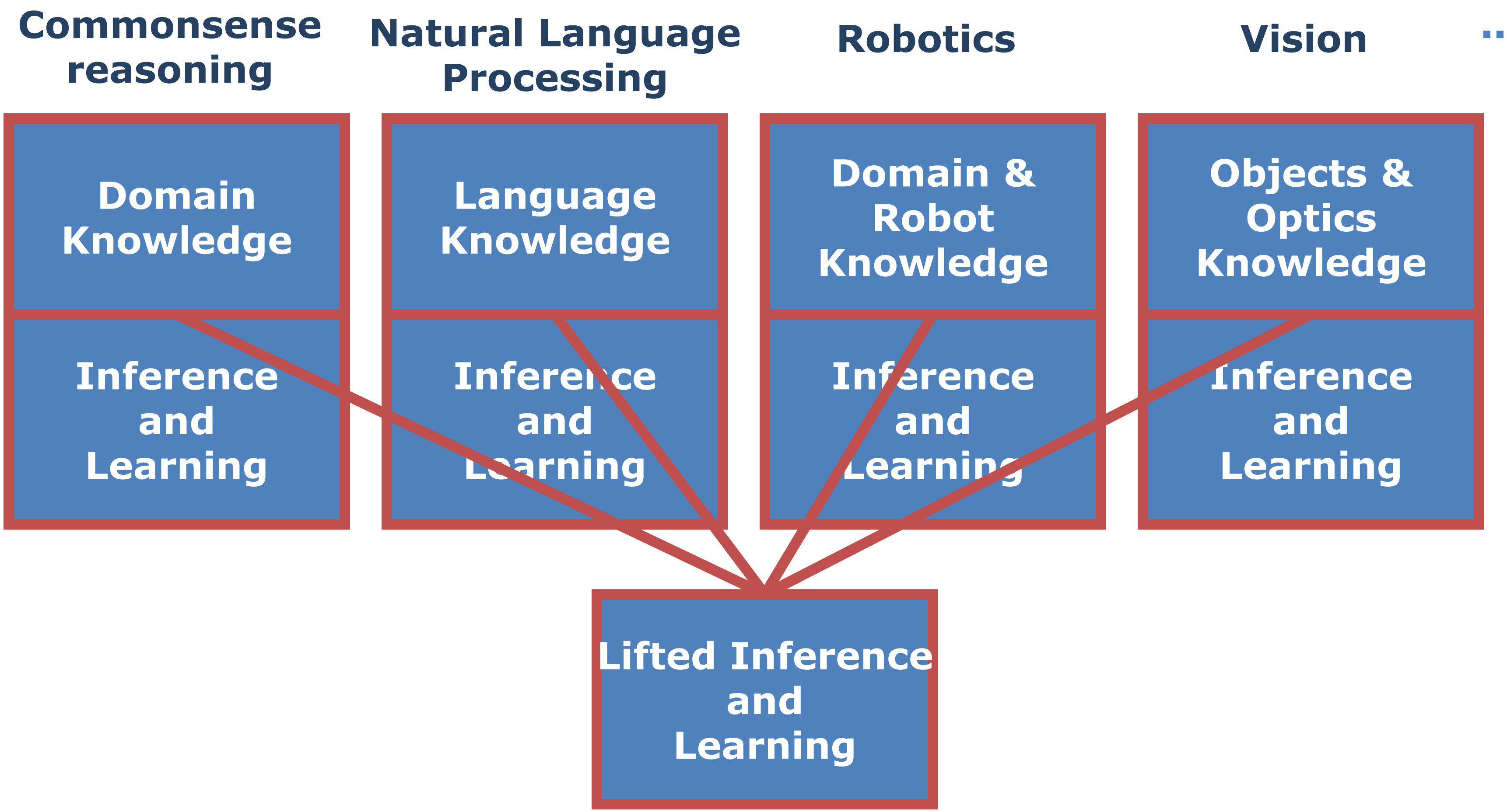
Let‘s deal with **uncertainty**, **objects**, **relations**, and **learning** jointly



The study and design of intelligent agents that act in noisy worlds composed of objects and relations among the objects

[slide inspired by de Salvo Braz]

The Big Picture on AI



Why the Tutorial?

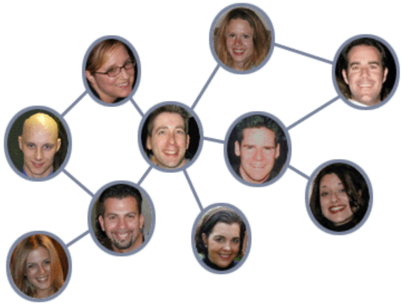
- **A very active, multi-disciplinary research area**
  - Involves all sub-disciplines of AI: *reasoning and acting under uncertainty, knowledge representation, constraint satisfaction, machine learning, …*
  - Unfortunately, can be hard to follow: ***they all speak a different language***
- **A success story**
  - Often outperforms state-of-the-art
  - Novel ways of *using the structure* for faster and/or more robust solutions
  - Growth path for (U)AI in general

STATISTICAL RELATIONAL LEARNING / AI: A SHORT OVERVIEW

Applications to Date

- Natural language processing
- Information extraction
- Entity resolution
- Link prediction
- Collective classification
- Social network analysis
- Robot mapping
- Activity recognition
- Scene analysis
- Computational biology
- Probabilistic Cyc
- Personal assistants
- Etc.






Information Extraction

Parag Singla and Pedro Domingos, “Memory-Efficient Inference in Relational Domains” (AAAI-06).

Singla, P., & Domingos, P. (2006).Memory-efficent inference in relatonal domains. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 500-505). Boston, MA: AAAI Press.

H. Poon & P. Domingos, Sound and Efficient Inference with Probabilistic and Deterministic Dependencies”, in Proc. AAAI-06, Boston, MA, 2006.

P. Hoifung (2006). Efficent inference. In Proceedings of the Twenty-First National Conference on Artificial Intelligence.

Information Extraction

Paper

Parag Singla and Pedro Domingos, "Memory-Efficient Inference in Relational Domains" (AAAI-06).

Singla, P., & Domingos, P. (2006).Memory-efficent inference in relatonal domains. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 500-505). Boston, MA: AAAI Press.

H. Poon & P. Domingos, Sound and Efficient Inference with Probabilistic and Deterministic Dependencies", in Proc. AAAI-06, Boston, MA, 2006.

P. Hoifung (2006). Efficent inference. In Proceedings of the Twenty-First National Conference on Artificial Intelligence.

Segmentation

Author

Title

Paper

Venue

Parag Singla and Pedro Domingos, "Memory-Efficient Inference in Relational Domains" (AAAI-06).

Singla, P., & Domingos, P. (2006). Memory-efficent inference in relatonal domains. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 500-505). Boston, MA: AAAI Press.

H. Poon & P. Domingos, Sound and Efficient Inference with Probabilistic and Deterministic Dependencies", in Proc. AAAI-06, Boston, MA, 2006.

P. Hoifung (2006). Efficent inference. In Proceedings of the Twenty-First National Conference on Artificial Intelligence.

Entity Resolution

Author
Title
Paper
Venue

Parag Singla and Pedro Domingos, "Memory-Efficient Inference in Relational Domains" (AAAI-06).

Singla, P., & Domingos, P. (2006). Memory-efficent inference in relatonal domains. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 500-505). Boston, MA: AAAI Press.

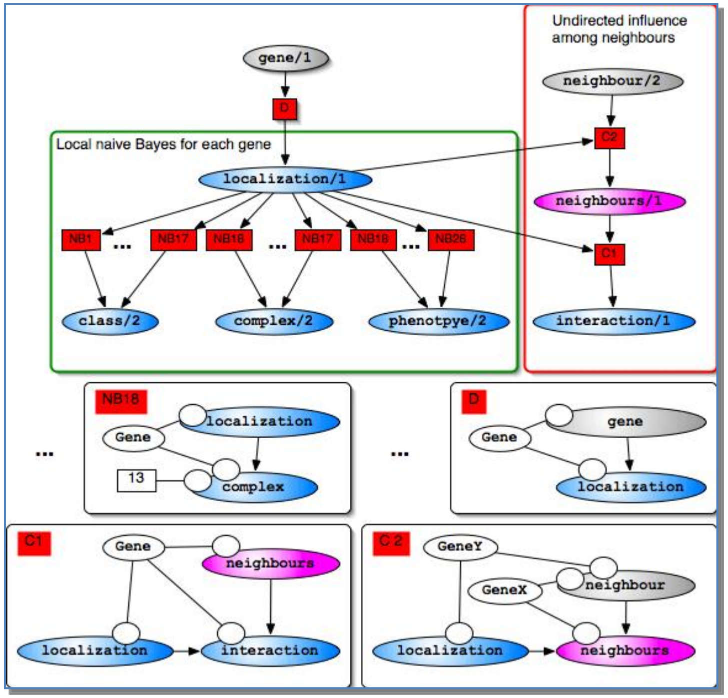
H. Poon & P. Domingos, Sound and Efficient Inference with Probabilistic and Deterministic Dependencies", in Proc. AAAI-06, Boston, MA, 2006.

P. Hoifung (2006). Efficent inference. In Proceedings of the Twenty-First National Conference on Artificial Intelligence.

Relations are at the heart of entity resolution

Gene Localization

- Predict the localization of a given gene in a cell among 15 distinct positions
- Relations important as sequence similarity does not help

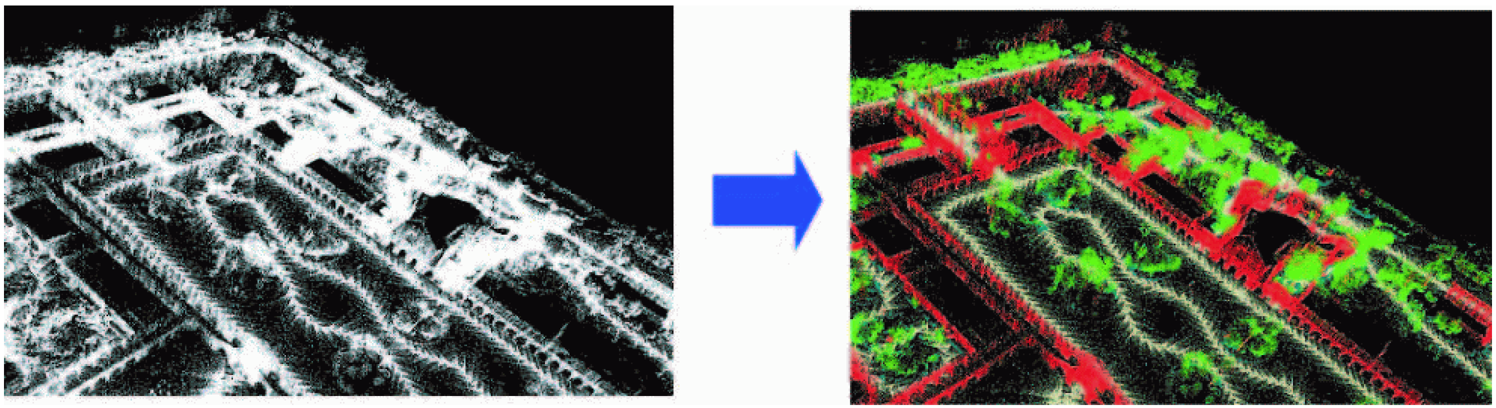


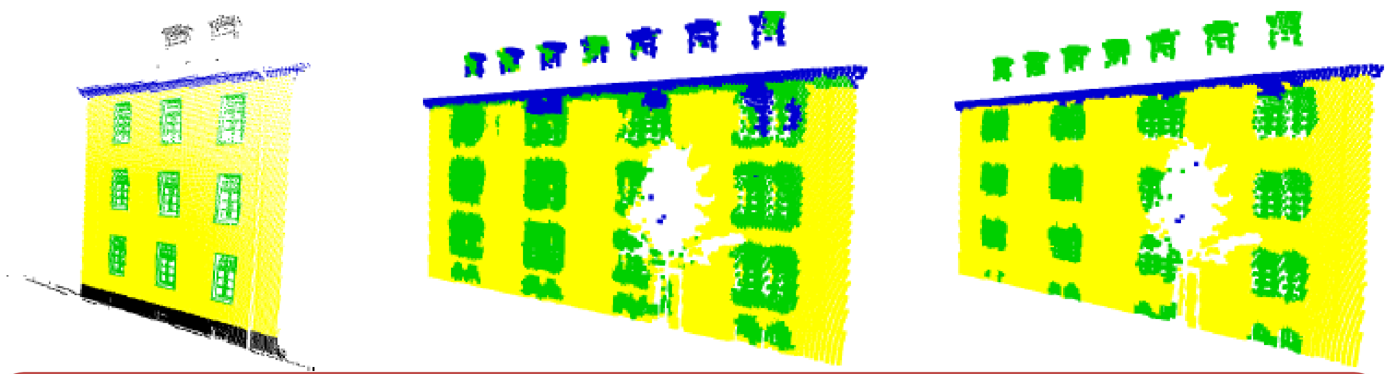
Relational Kernels better then Hayashi et al.'s KDD Cup 2001 winning approach

[Anguelov et al. CVPR05, Triebel et al. ICRA06, …]

Semantic Labeling of 3D Scan Data

- Neighbouring pixels/voxels have the same semantic label





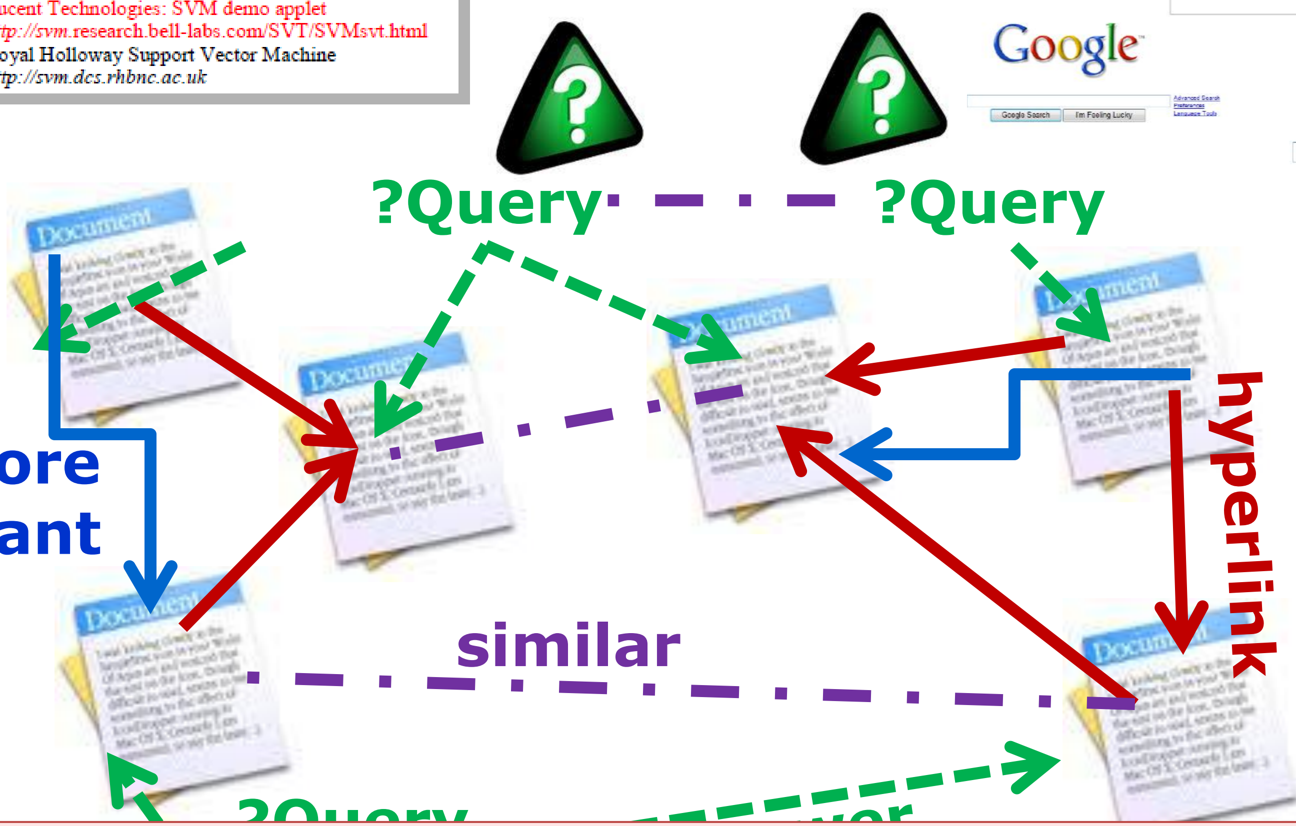
Relations as constraints

Web Search

(3 < 2),
(7 < 2),
(7 < 4),
(7 < 5),
(7 < 6)

1. Kernel Machines
   http://svm.first.gmd.de/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm light/
4. An Introduction to Support Vector Machines
   http://www.support-vector.net/
5. *Support Vector Machine and Kernel ... References*
   http://svm.research.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES ...
   http://www.jiscmail.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsvt.html
8. Royal Holloway Support Vector Machine
   http://svm.dcs.rhbnc.ac.uk

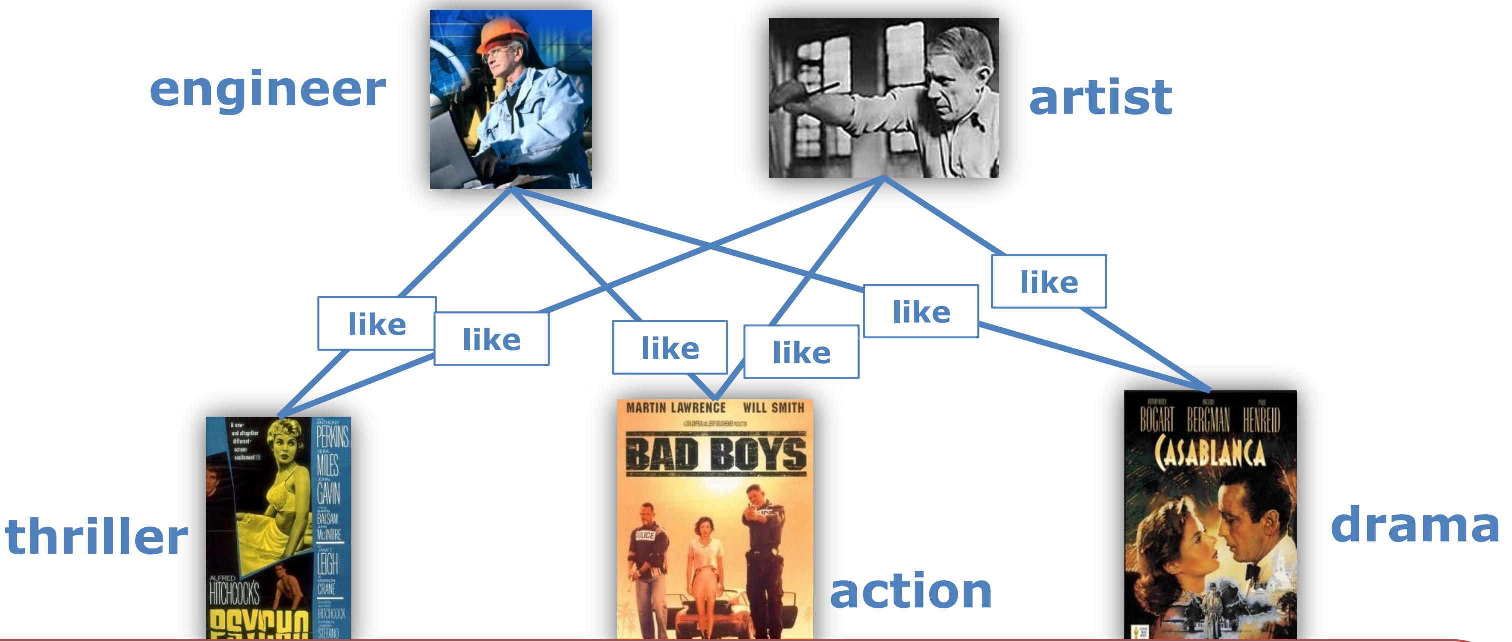
Rel(1),
NotRel(2),
Rel(3),
NotRel(4),
NotRel(5),
NotRel(6),
Rel(7)



Relational approaches outperfom traditional ranking approaches

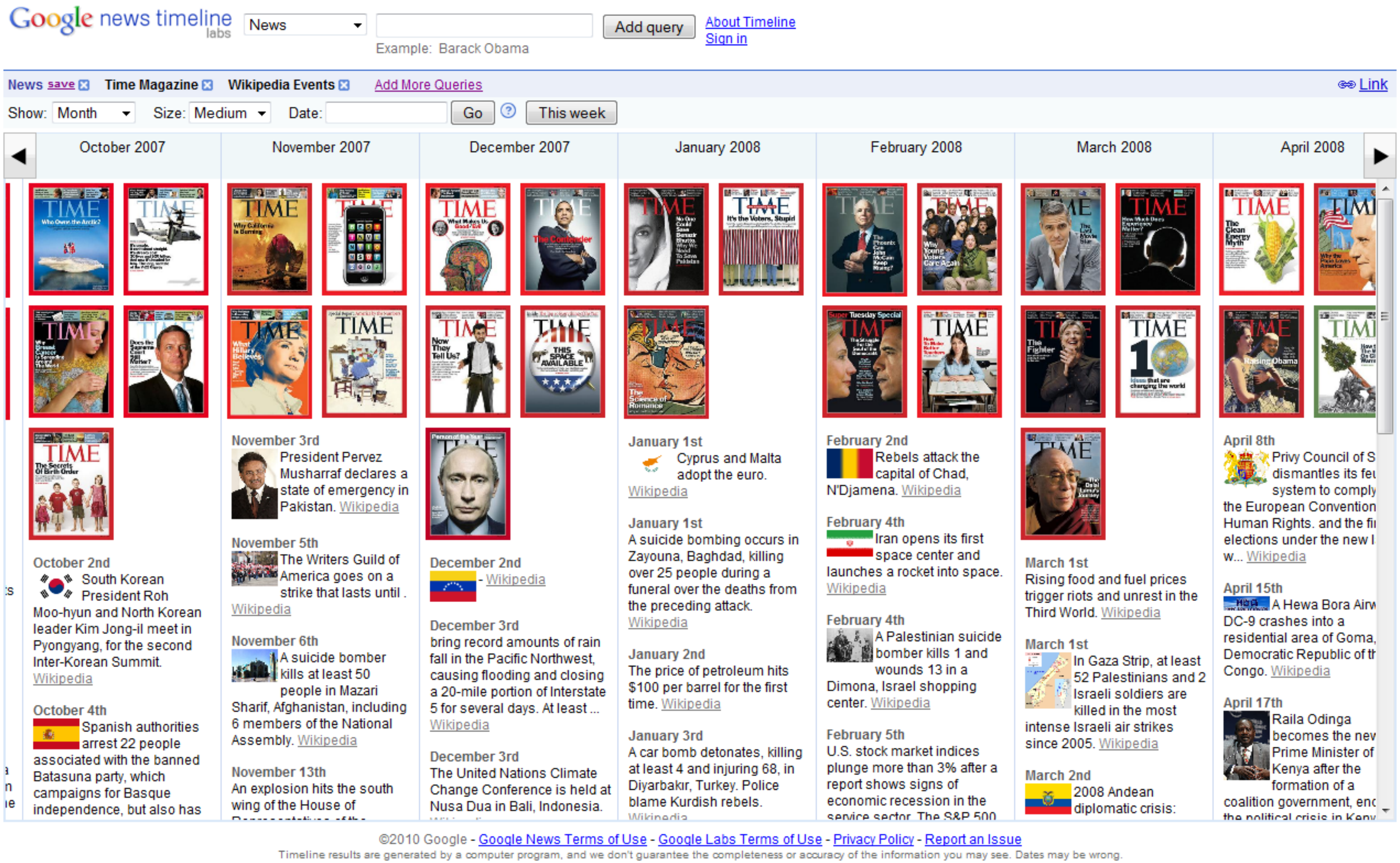
Social Recommendation / Collaborative Filtering

- Predict whether a user **likes** a movie given attributes of users and movies, as well as known ratings and **complex link structures**

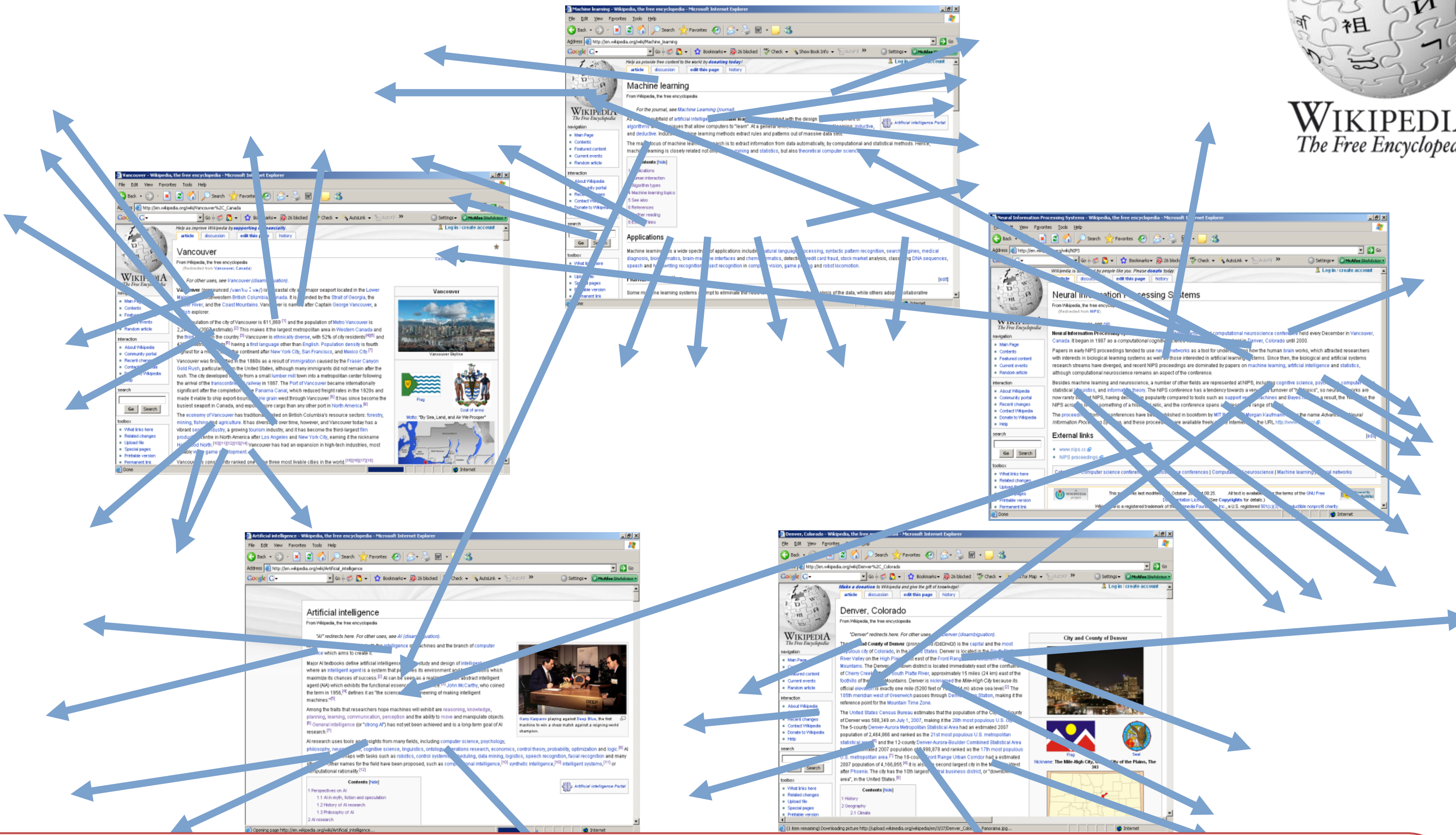


Relational approaches outperfom set-based recommendation systems

What is the world talking about ?



Topic Models



Relational approaches estimate better low-dimensional embeddings

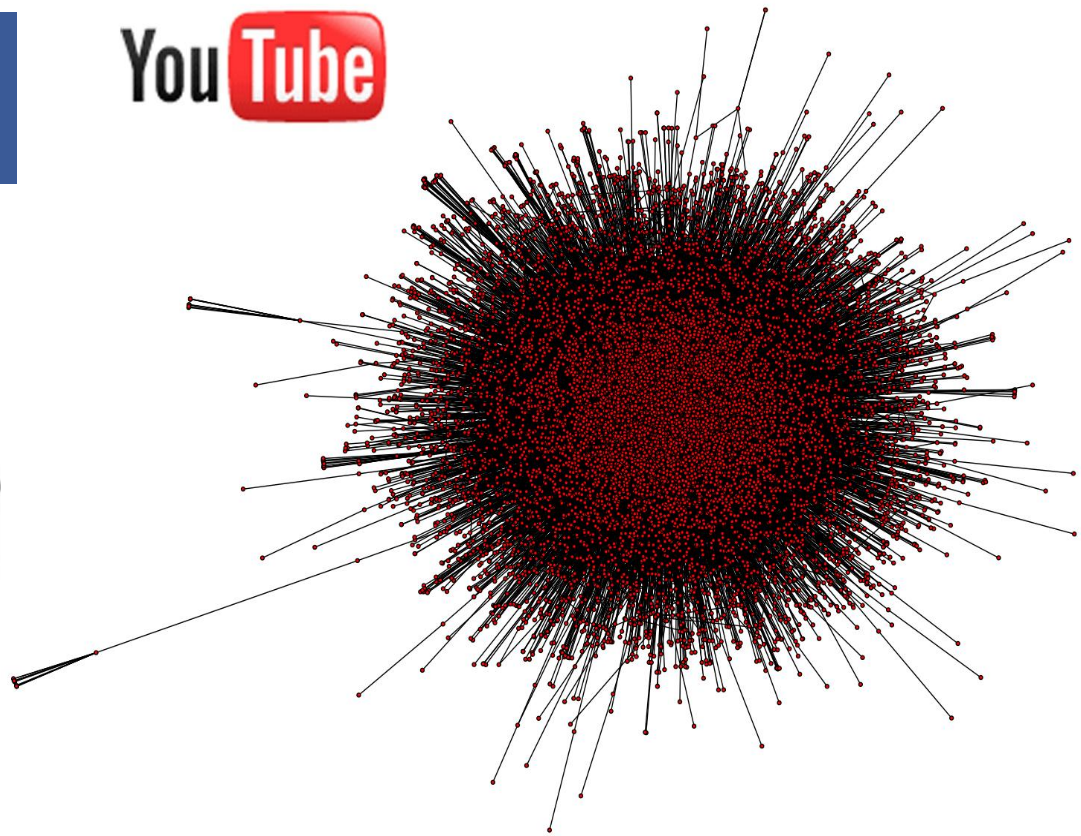
How do you spend your spare time?

YouTube like media portals have changed the way users access media content in the Internet

Every day, millions of people visit social media sites such as Flickr, YouTube, and Jumpcut, among others, to share their photos and videos, …

while others enjoy themselves by searching, watching, commenting, and rating the photos and videos; what your friends like will bear great significance for you.

How do you efficiently broadcast information?

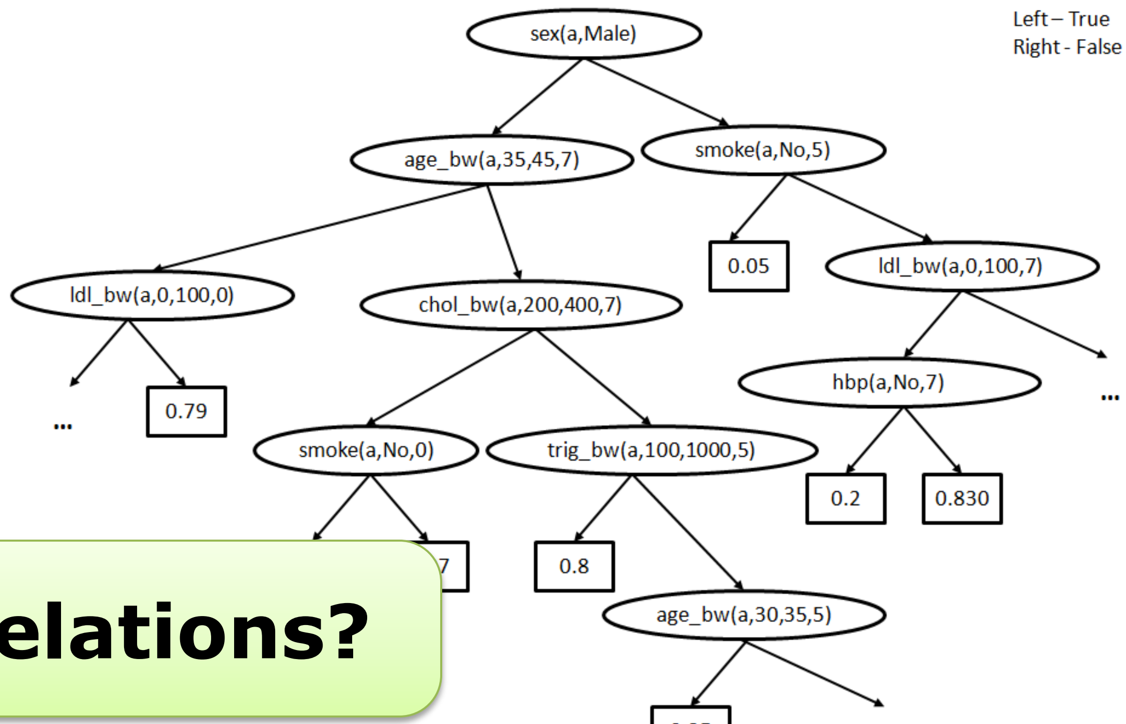


Lifted inference faster than belief propagation

Predicting Coronary Artery Calcification Levels

- Cardiovascular disease cost the EU EURO169 billion in 2003 and the USA about EURO310.23 billion in direct and indirect annual costs.
- By comparison, the estimated cost of all cancers is EURO146.19 billion and HIV infections EURO22.24 billion.

Algorithm	Accuracy	AUC-ROC
J48	0.667	0.607
SVM	0.667	0.5
AdaBoost	0.667	0.608
Bagging	0.677	0.613
NB	0.75	0.653
RPT	0.669*	0.778
RFGB	0.667*	0.819



So, what are relations?

Relational models provide new insights

What are Relations?

- There are several types of relations and in turn there are several views on what (statitical) relational learning is

1. **Relations provide additional correlations/ regularization**
   - If two words occure frequently in the same context (page, paragraph, sentence, …) then there must be some semantic relation between them

2. **Often extensional (data) only, for one relation**
   - Covariance function, distance functions, kernel functions, graphs, tensors, random walks with restarts…

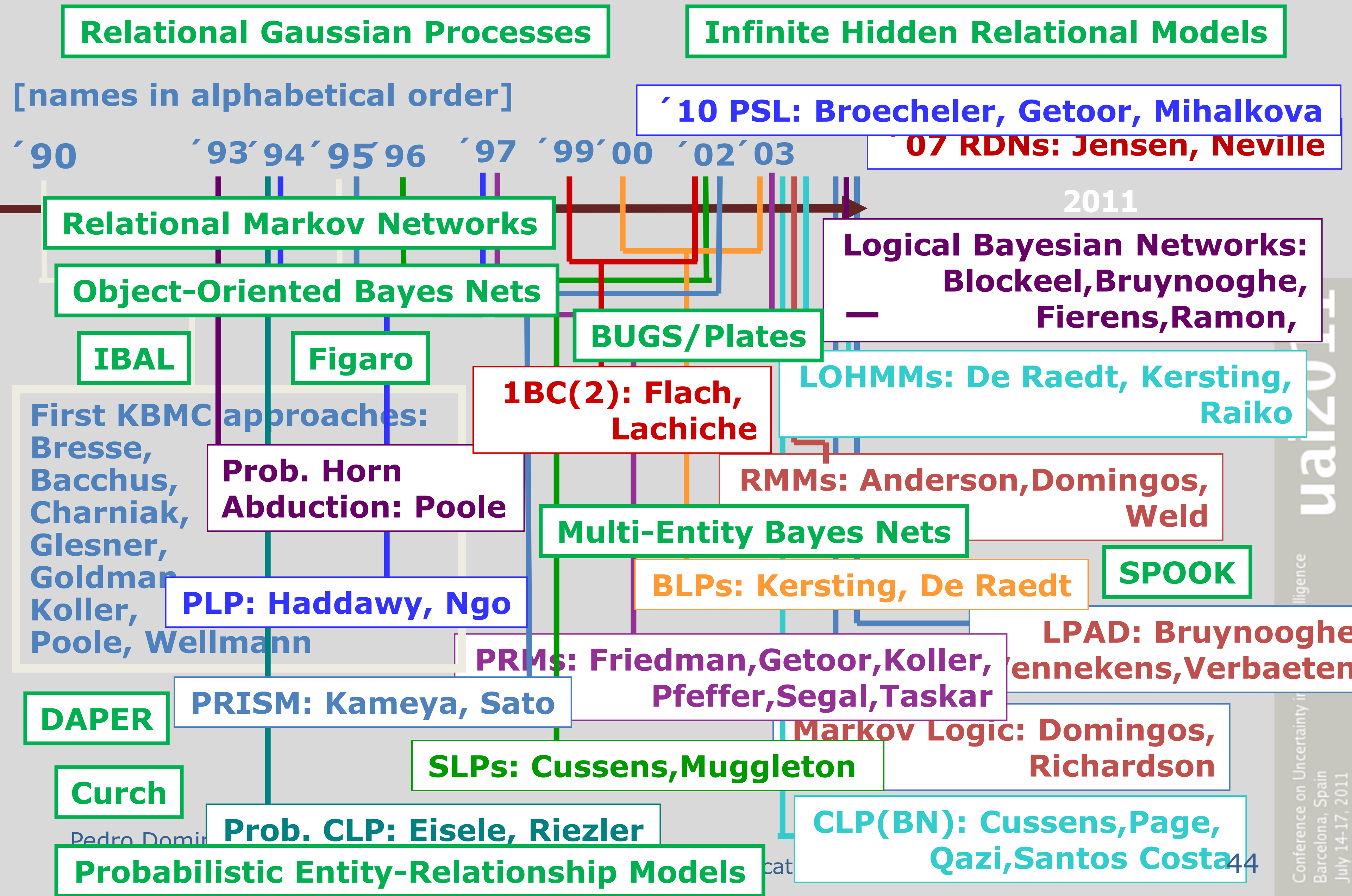
What are Relations?

3. **Relations are symmetries/redundancies in the model**
   - E.g. lifted inference based on bisimulation

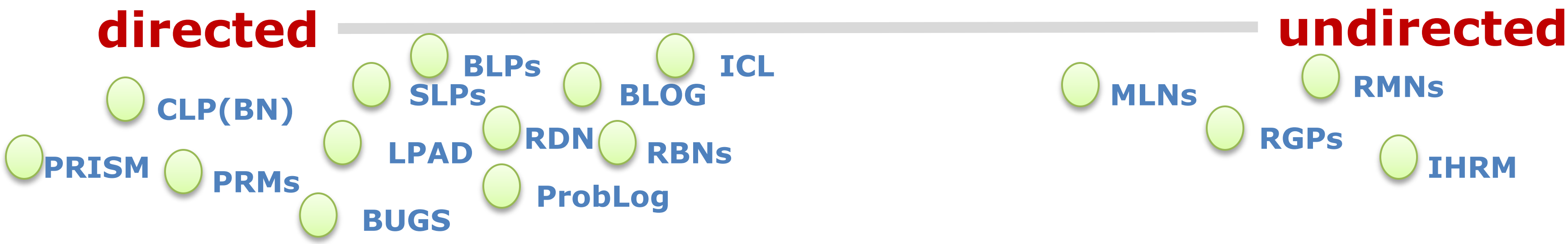
4. **Hypergraph representations of data**
   - Multiple (extensional) relations
   - Random walks with restarts as similarity measure or to produce path features

5. **Full-fledged relational (or logical) knoweldge as considered in this tutorial**
   - Multiple (often typed) relations
   - Intensional formulas (often Horn clauses)
     ancestor(X,Z) ^ parent(Z,Y) ⇒ ancestor(X,Y)

The SRL Alphabet Soup



Key Dimensions with some prototypes



[Getoor et al. 2002; Kersting De Raedt 2007]

Directed: Probabilistic Relational Models (PRMs) Bayesian logic Programs (BLPs)

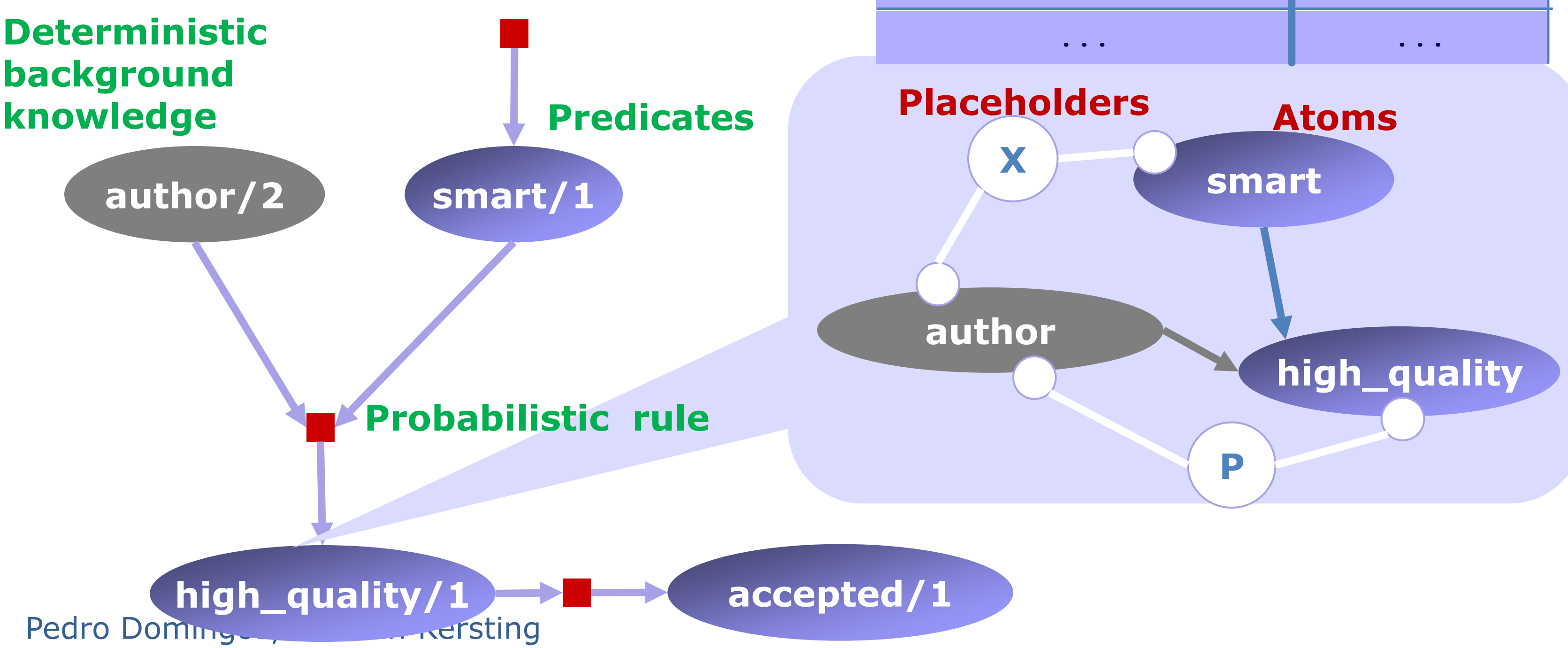
$$\forall x\; author(x,p) \wedge smart(x) \Rightarrow high\_quality(p)$$

$$\forall x\; high\_quality(p) \Rightarrow accepted(p)$$

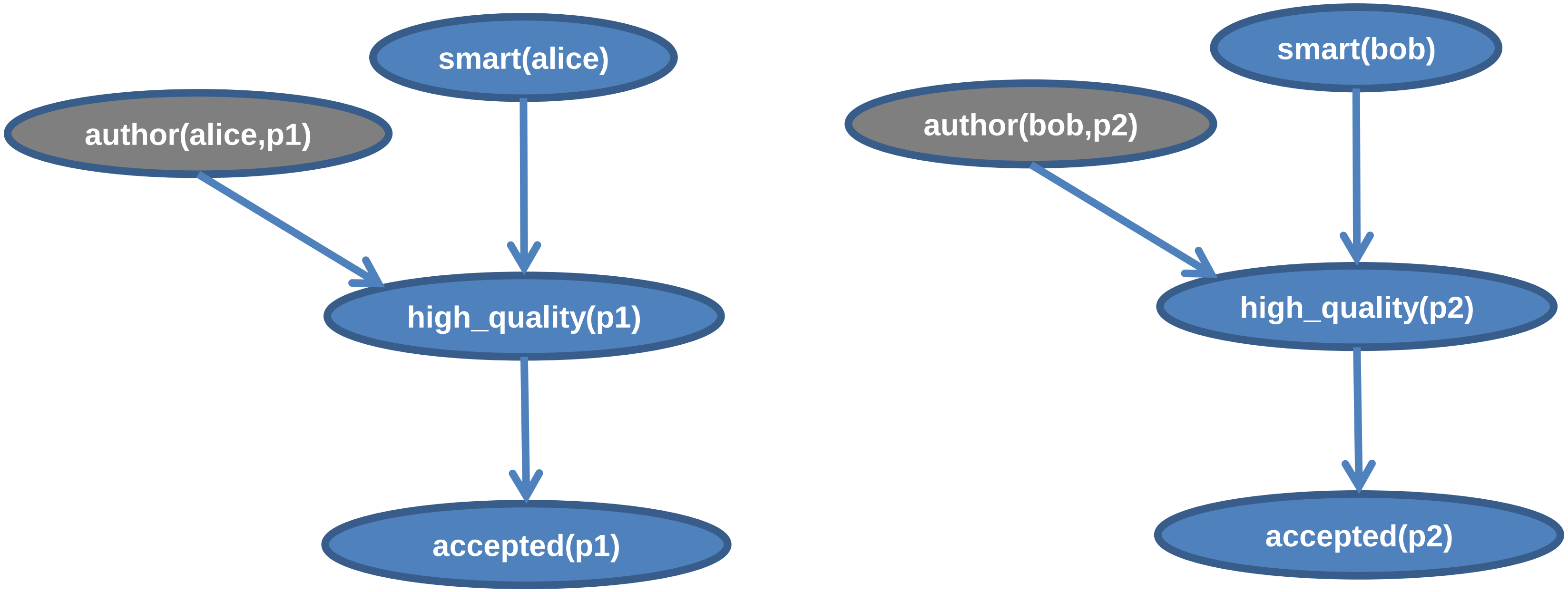
Macro for conditional probability table

high_quality(Y)	smart(X)
(0.9,0.1)	yes
...	...

Rule Graph



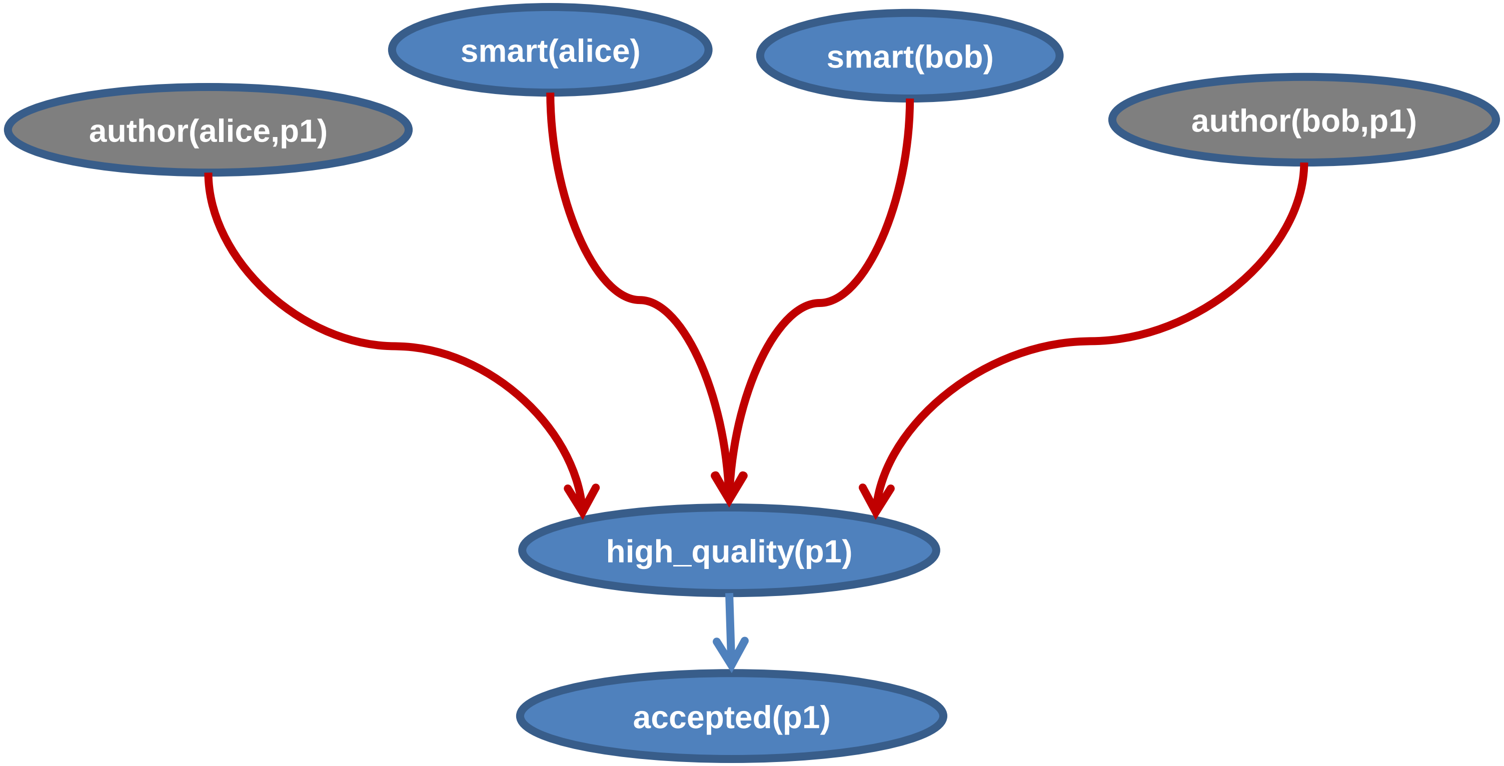
Inference on BN constructed by instantiating the rules/ macros using back- or forward chaining



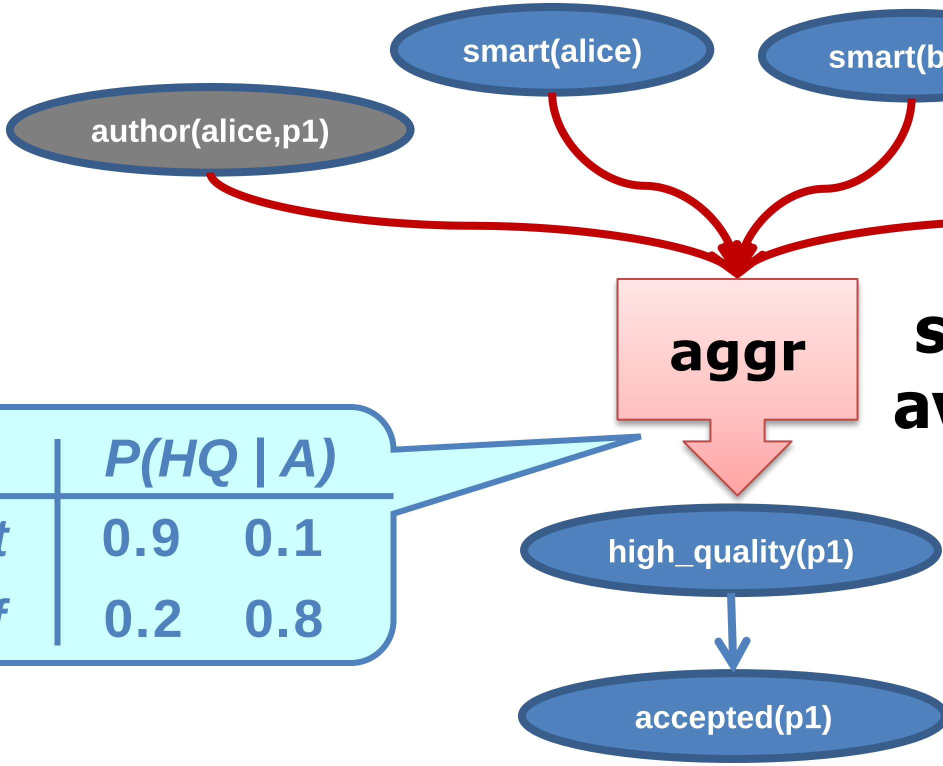
But what happens if instead we have author(bob,p1)?

So, we can deal with a variable number of objects. The induced BN depends on the domain elements and the background knowledge we have.

Directed: Aggregate Dependencies



Directed: Aggregate Dependencies



A	P(HQ | A)	
t	0.9	0.1
f	0.2	0.8

sum, min, max, avg, mode, count

Still, the induced model is assumed to be acyclic

[Neville, Jensen 2007]

Option 1 : Relational Dependency Networks (RDNs)

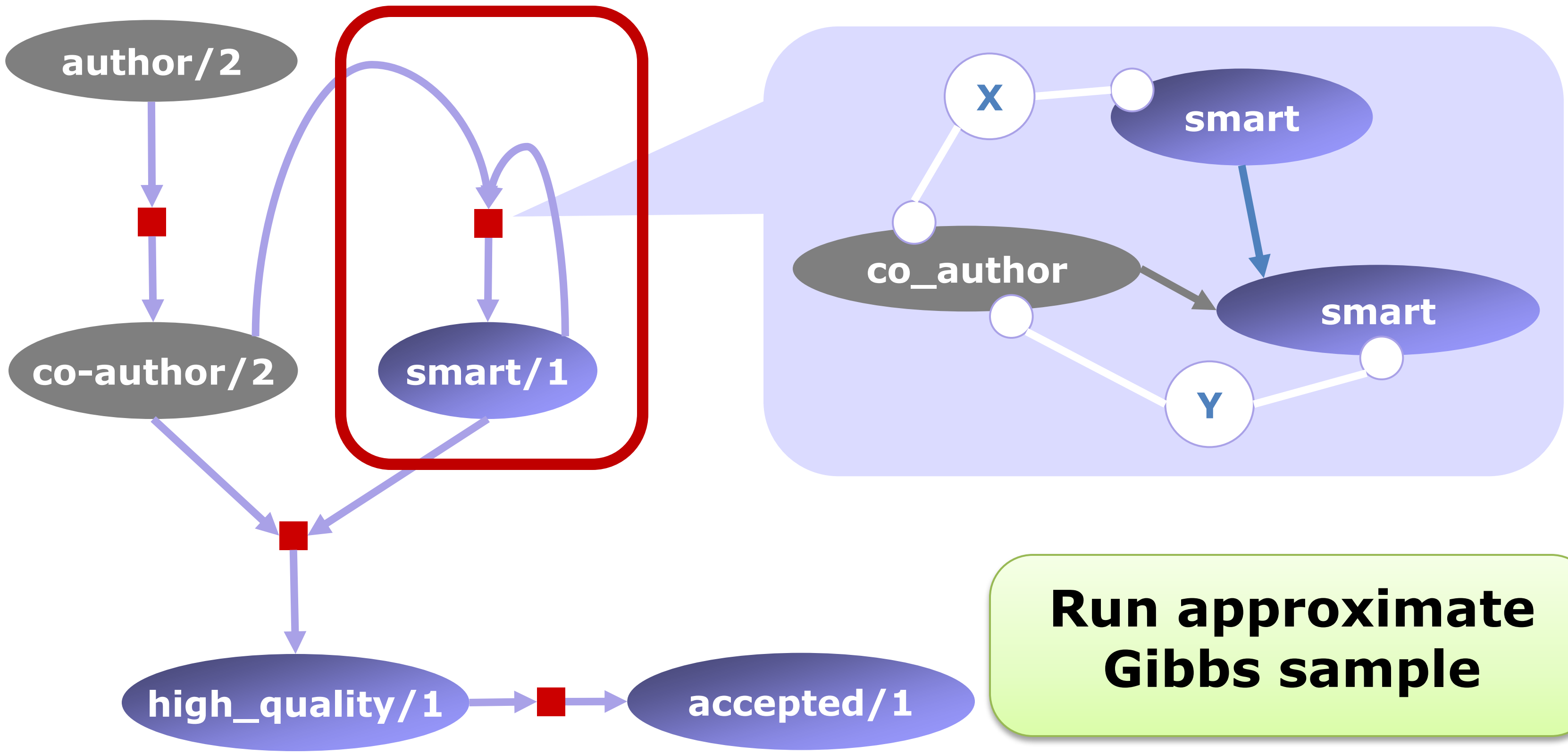
$$\forall x\ author(x,p) \wedge smart(x) \Rightarrow high\_quality(p)$$

$$\forall x\ high\_quality(p) \Rightarrow accepted(p)$$

cyclic dependency

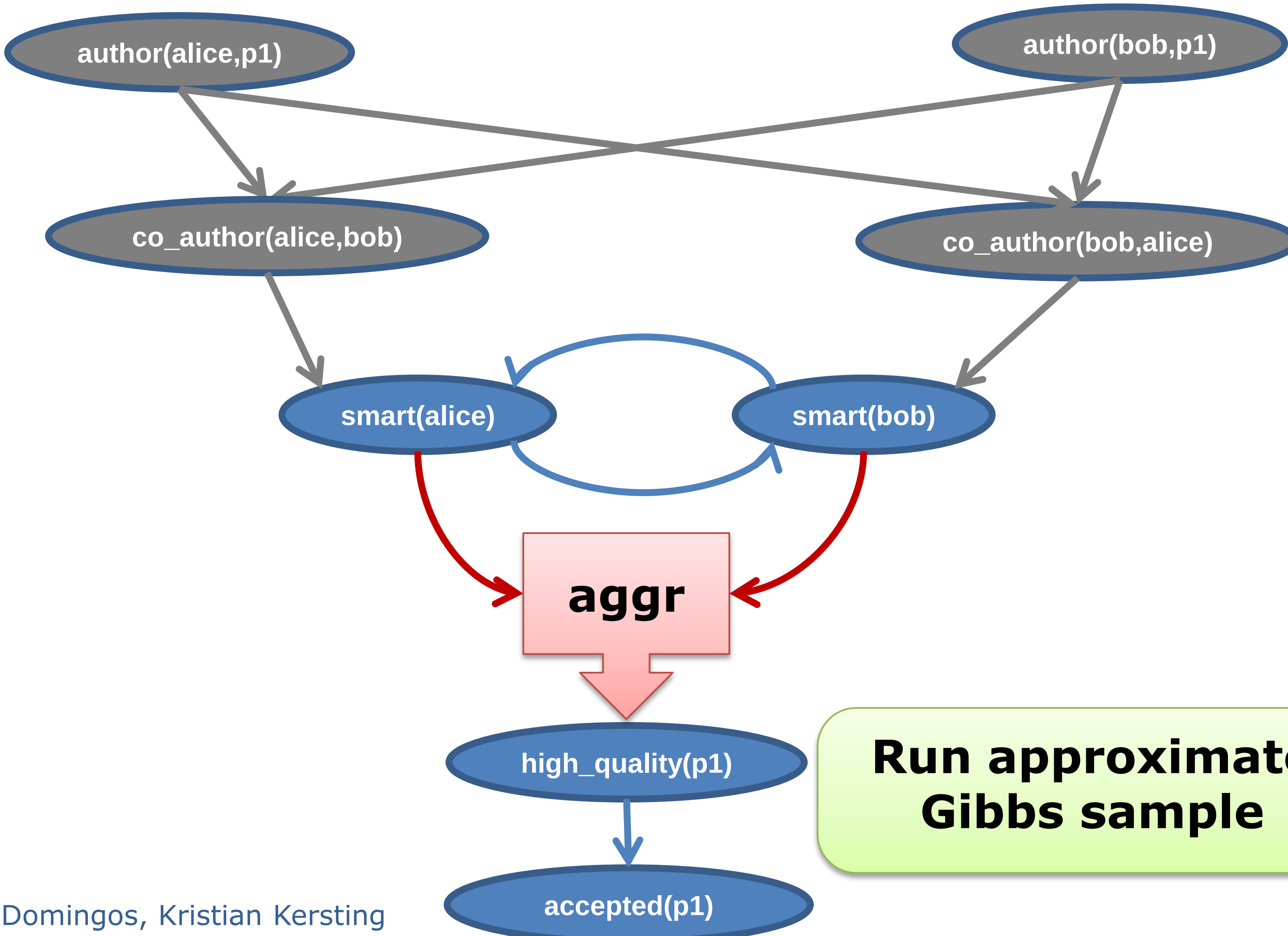
$$\boxed{\forall x,y\ co\_author(x,y) \wedge smart(x) \Rightarrow smart(y)}$$

$$\forall x,y\ \exists p\ author(x,p) \wedge author(y,p) \Rightarrow co\_author(x,y)$$



Run approximate Gibbs sample

Relational Dependency Networks



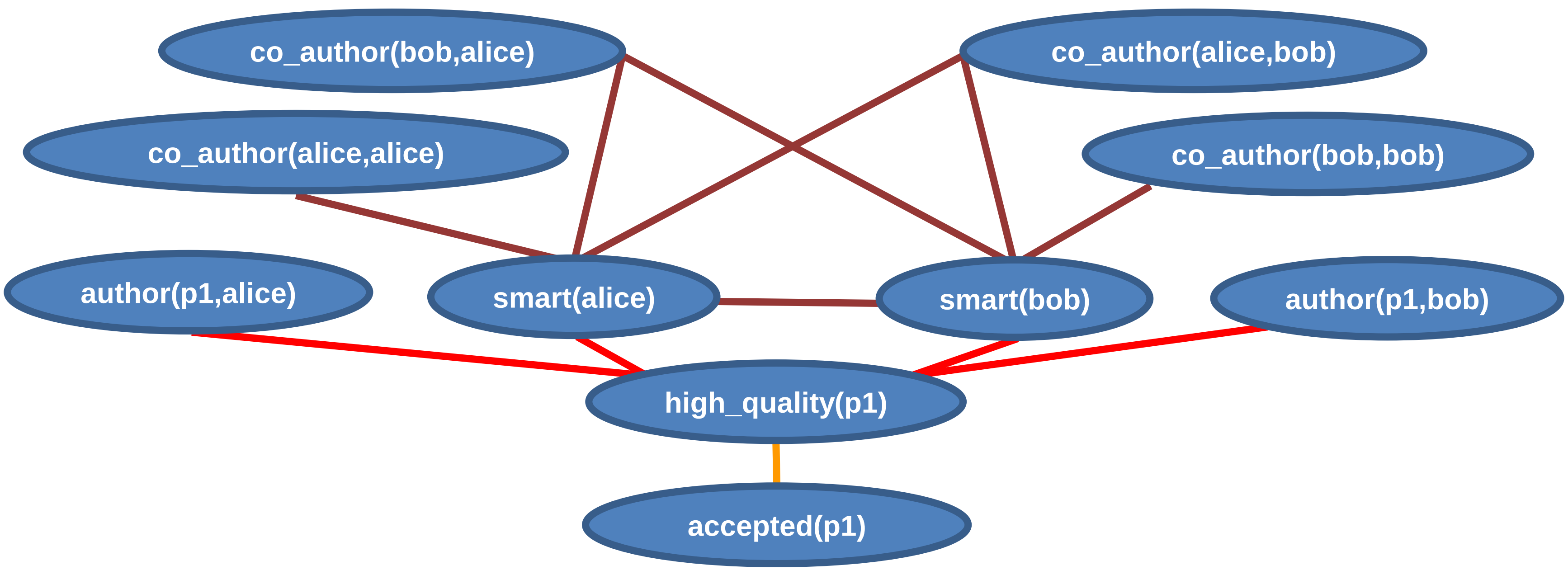
Run approximate Gibbs sample

[Richardson, Domingos MLJ 62(1-2): 107-136, 2006]

Option 2: Markov Logic Networks

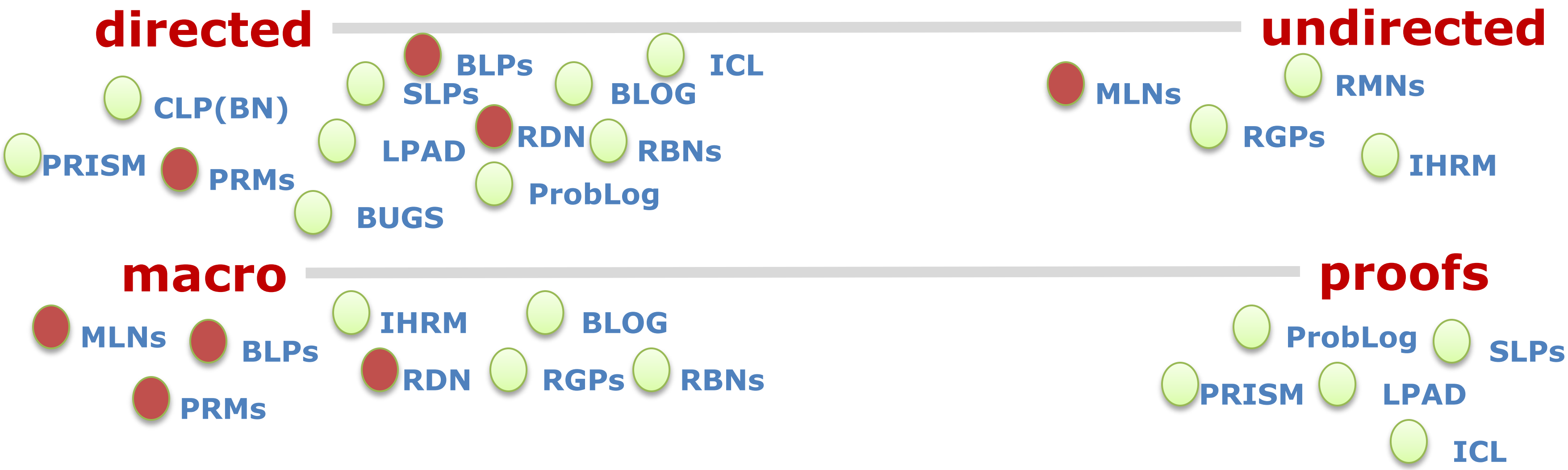
Suppose we have constants: alice, bob and p1

1.5	$\forall x\ author(x,p) \wedge smart(x) \Rightarrow high\_quality(p)$
1.1	$\forall x\ high\_quality(p) \Rightarrow accepted(p)$
1.2	$\forall x,y\ co\_author(x,y) \Rightarrow (smart(x) \Leftrightarrow smart(y))$
∞	$\forall x,y\ \exists p\ author(x,p) \wedge author(y,p) \Rightarrow co\_author(x,y)$

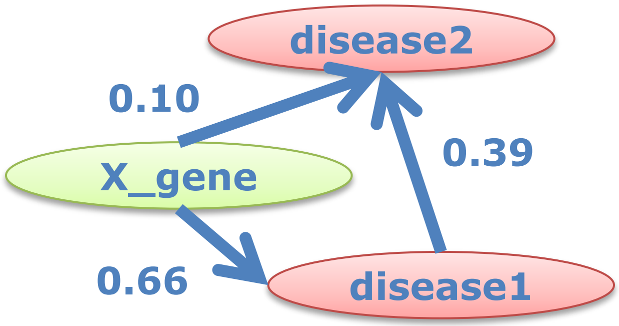


Compile to an undirected model

Key Dimensions with some prototypes



ProbLog

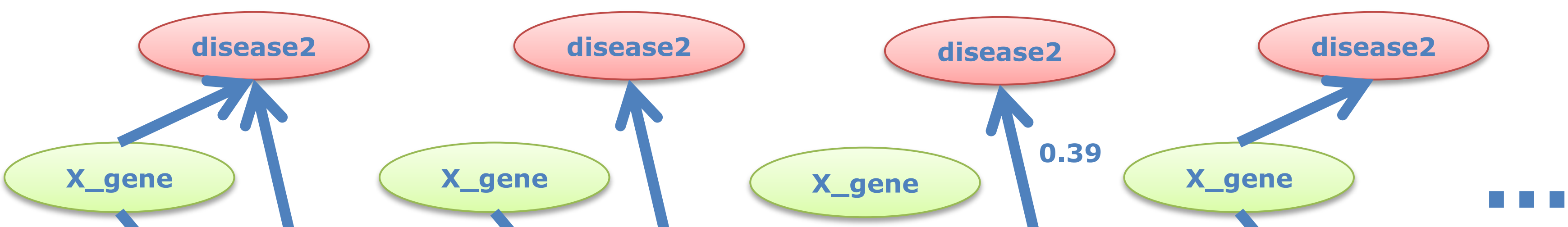



```
0.10 :: edges(x_gene, disease2)
0.66 :: edge(x_gene, disease1)
0.39 :: edges(disease1,disease2)

path(X,Y) :- edge(X,Y)
path(X,Y) :- edges(X,Z), path(Z,Y)
```

- Label of a clause/fact c is the probability that c belongs to the target program; Facts/clauses independent of each other
- Defines a distribution over programs $P(L|Program) = \prod_{c_i \in L} p_i \prod_{c_j \notin L} (1 - p_j)$

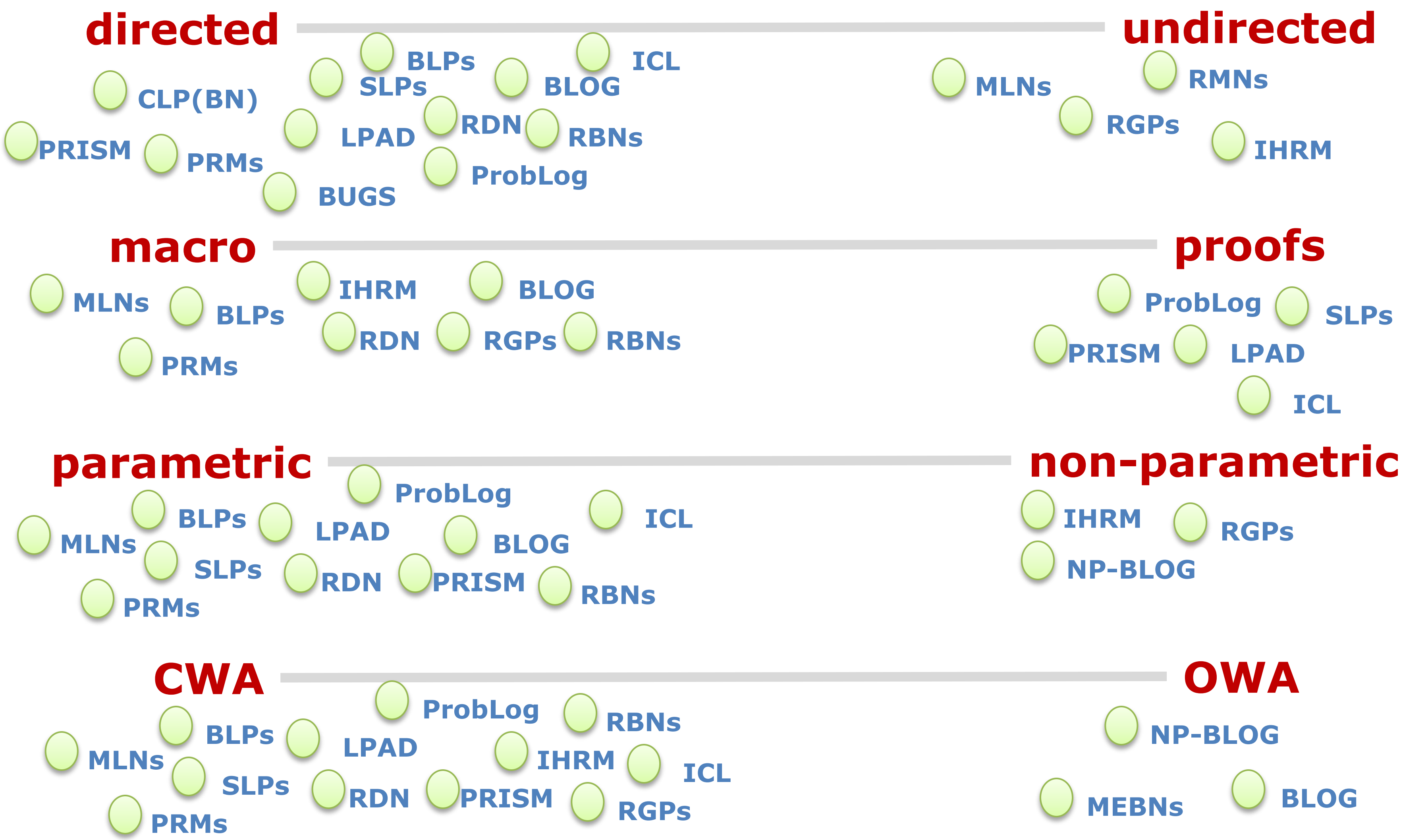
P(path(x_gene,disease2))= sum of probs of all programs that entail the query

P=0.1*0.66*0.39 + P=(1-0.1)*0.66*0.39 + P=0.1*0.66*(1-0.39)



Exponentially many subprograms! To avoid explosion, consider proofs/paths only + store them in a BDD in order to count correctly

Many other approaches !!



And actually they span the whole AI spectrum

- Relational topic models
- Mixed-membership models
- Relational Gaussian processes
- Relational reinforcement learning
- (Partially observable) MDPs
- Systems of linear equations
- Kalman filters
- Declarative information networks

No, this is very much like in the early days of UAI !

So, should we worry about the soup?

The early days of UAI

Maximum entropy inference

Odds-likelihood updating

Dempster-Shafer Belief Functions

Mycin's Certainty Factors

Bayesian Networks

Expert-rating

Decision-theoretic metrices

Belief Maintenance System

Bayes' Theorem

Prospector

Probabilistic Logic

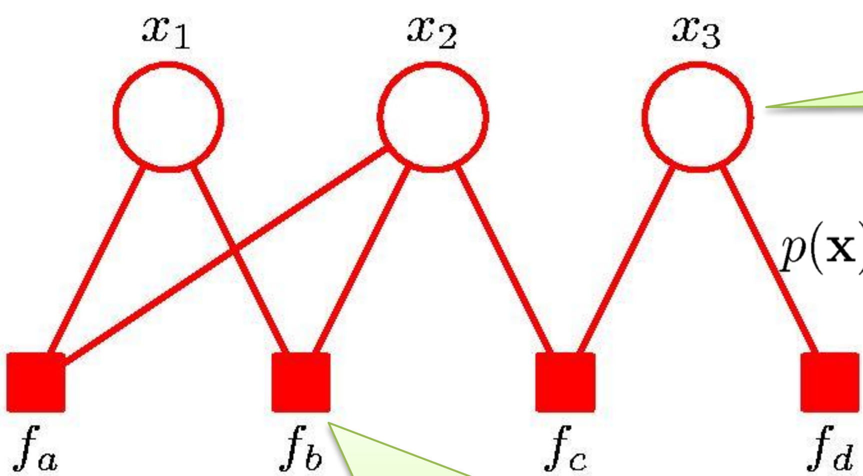
Fuzzy Set Theory

Incidence Calculus

[B. Wise, M. Henrion. A Framework for Comparing Uncertain Inference Systems to Probability. UAI-85]
[A. Bundy. Incidence Calculus: A Mechanism for Probabilistic Reasoning. UAI-85]
[D. Hunter. Uncertain. Reasoning Using Maximum Entropy Inference. UAI-85]
[D. Heckerman. Probabilistic Interpretations for MYCIN's Certainty Factors. UAI-85]
[S. Ursic. Generalizing Fuzzy Logic Probabilistic Inferences. UAI-86]
[N.J. Nilsson. Probabilistic Logic. *Artificial Intelligence* 28(1): 71-87, 1986]
[B. Falkenheiner. Towards a General-Purpose Belief Maintenance System. UAI-86]
[D. Heckerman. An Empirical Comparison of Three Inference Methods. UAI-88]

This soup boiled down to Graphical Models as intermediate representation

Distributions can naturally be represented as Factor Graphs



Random variable

Factor resp. potential

$$p(\mathbf{x}) = f_a(x_1, x_2) f_b(x_1, x_2) f_c(x_2, x_3) f_d(x_3)$$

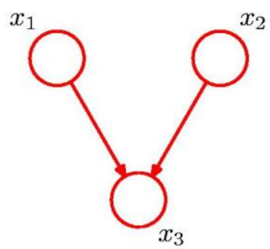
$$p(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s)$$

unnormalized !

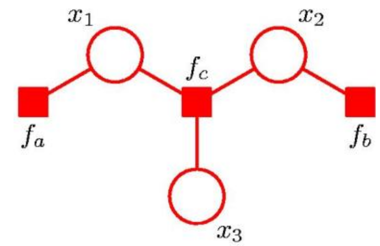
- There is an edge between a circle and a box if the variable is in the domain/scope of the factor

uai2011
Conference on Uncertainty in Artificial Intelligence
Barcelona, Spain
July 14-17, 2011

Factor Graphs from Graphical Models



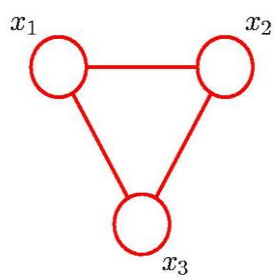
$$p(\mathbf{x}) = p(x_1)p(x_2) p(x_3|x_1,x_2)$$



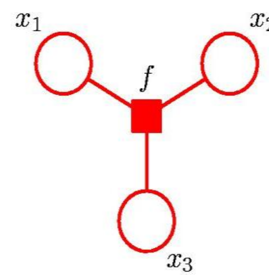
$$f_a(x_1) = p(x_1)$$

$$f_b(x_2) = p(x_2)$$

$$f_c(x_1,x_2,x_3) = p(x_3|x_1,x_2)$$



$$\psi(x_1,x_2,x_3)$$



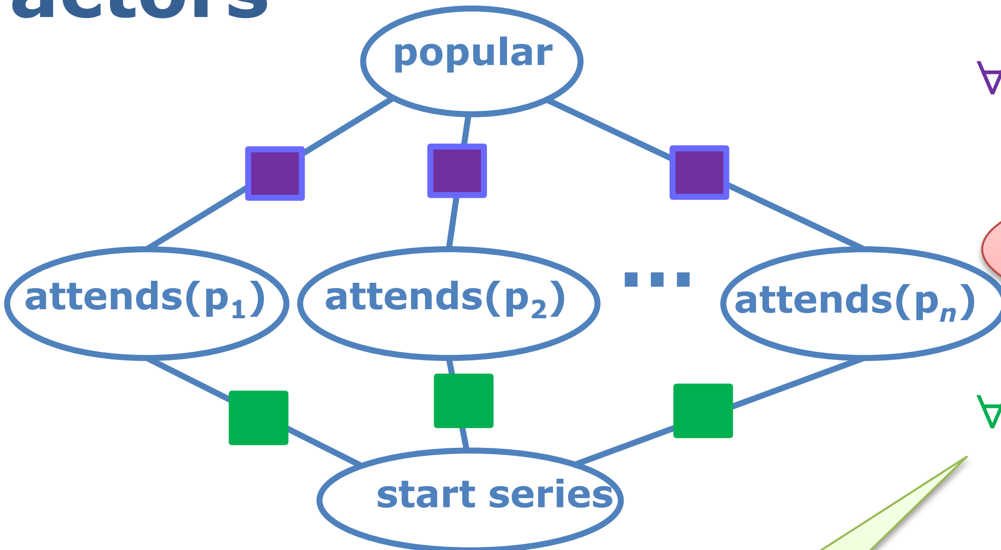
$$f(x_1,x_2,x_3) = \psi(x_1,x_2,x_3)$$

Similar "boiling down" process is going on in SRL!

Boiled-Down SRL Alphabet Soup

- Given a relational model in your language of choice, a set of constants and a query, compile everything into an intermediate respresentation
  - Factor graphs
  - BDDs, Artihmetic Circuits, d-DNNFs, …
  - Weighted CNFs
- Run inference

Rules + Potential: Logically Parameterized Factors



$\forall X.\ \phi_1(\text{popular}, \text{attends}(X))$

Logical Variables parameterize RV

$\forall X.\ \phi_2(\text{attends}(X), \text{series})$

Atoms represent a set of random variables

Parfactors parameterized factors

There can also be **contraints** to logical variables such as **X=/=UAI11**

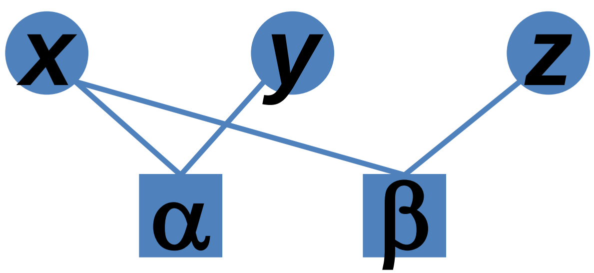
Rules + Weights: Weighted CNF

- Weighted MAX-SAT as mode finding for log-linear distributions
- Each configuration has a cost: the sum of the weights of the unsatisfied (ground) clauses.
- An infinite cost gives a 'hard' clause.
- Goal: find an assignment with minimal cost.

Weigthed CNF

$$\overset{\color{red}{w1}}{\underbrace{(x \vee y)}_{\alpha}} \wedge \overset{\color{purple}{w2}}{\underbrace{(\neg x \vee z)}_{\beta}}$$

Factor Graph:



x	y	$f_\alpha(x,y)$
0	0	0
0	1	w1
1	0	w1
1	1	w1

x	z	$f_\beta(x,z)$
0	0	w2
0	1	w2
1	0	0
1	1	w2

ILP= Machine Learning + Logic Programming

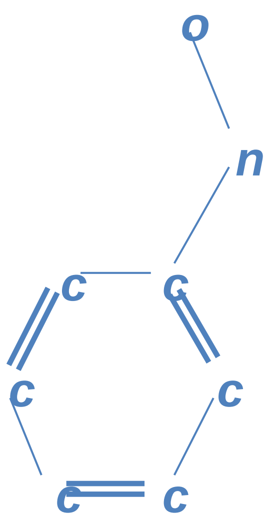
[Muggleton, De Raedt JLP96]

Find a set of general rules

mutagenic(X) :- atom(X,A,c),charge(X,A,0.82)
mutagenic(X) :- atom(X,A,n),...

Examples E

pos(mutagenic(m_1))
neg(mutagenic(m_2))
pos(mutagenic(m_3))
...



Background Knowledge B

molecule(m_1)	**molecule(m_2)**
atom(m_1,a_{11},c)	**atom(m_2,a_{21},o)**
atom(m_1,a_{12},n)	**atom(m_2,a_{22},n)**
bond(m_1,a_{11},a_{12})	**bond(m_2,a_{21},a_{22})**
charge(m_1,a_{11},0.82)	**charge(m_2,a_{21},0.82)**
...	**...**

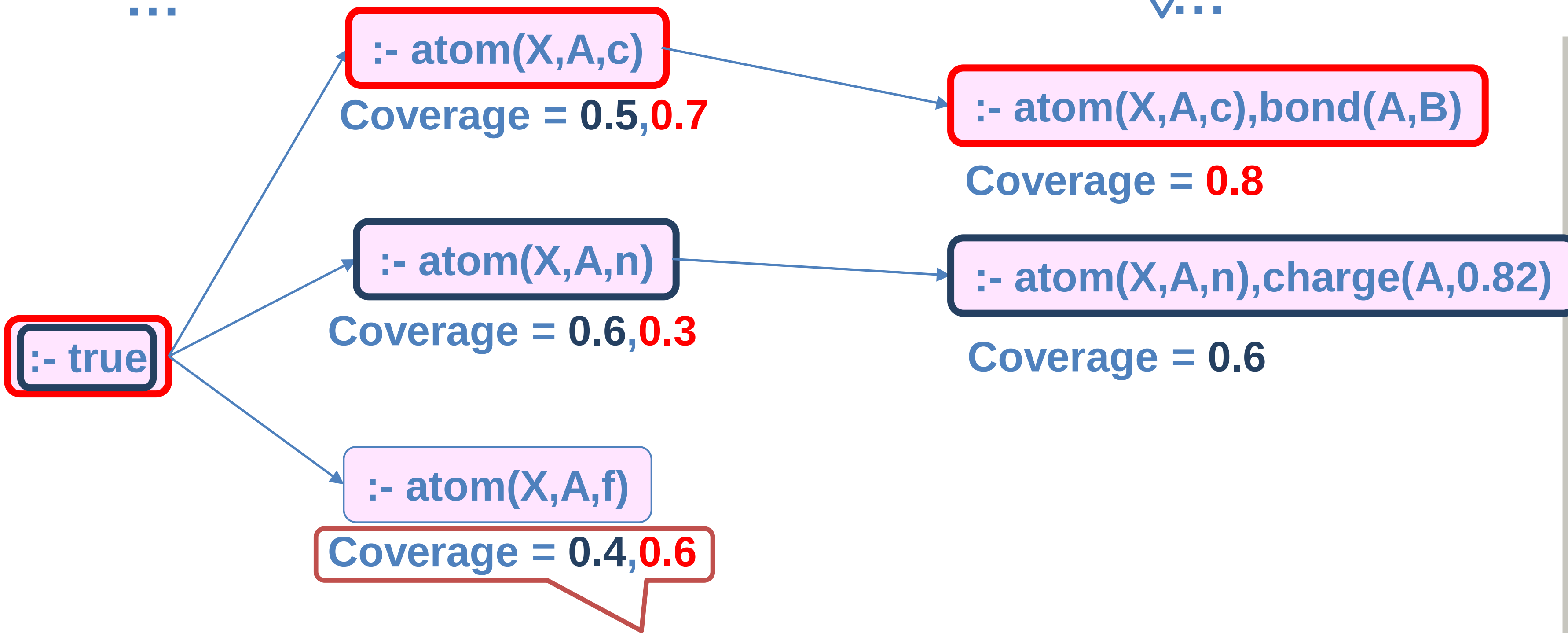
Example ILP Algorithm: FOIL

[Quinlan MLJ 5:239-266, 1990]

mutagenic(X) :- atom(X,A,n),charge(A,0.82) 0

mutagenic(X) :- atom(X,A,c),bond(A,B) $\vee$ 1 $\equiv$ 1

... $\vee$...



Some objective function, e.g.
percentage of covered positive examples

Vanilla SRL Approach [De Raedt, K ALT04]

mutagenic(X) :- atom(X,A,n),charge(A,0.82)

mutagenic(X) :- atom(X,A,c),bond(A,B)

...

=0.882

- Traverses the hypotheses space a la ILP
- Replaces ILP's 0-1 covers relation by a "smooth", probabilistic one [0,1]

$$\begin{aligned} \mathrm{cover}(e,H,B) &= P(e|H,B) \\ \mathrm{cover}(E,H,B) &= \prod_{e\in E} \mathrm{cover}(e,H,B) \end{aligned}$$

MARKOV LOGIC

MARKOV LOGIC

uai2011
Conference on Uncertainty in Artificial Intelligence
Barcelona, Spain
July 14-17, 2011

Overview

- **Representation**
- Inference
- Learning
- Applications
- Discussion

Propositional Logic

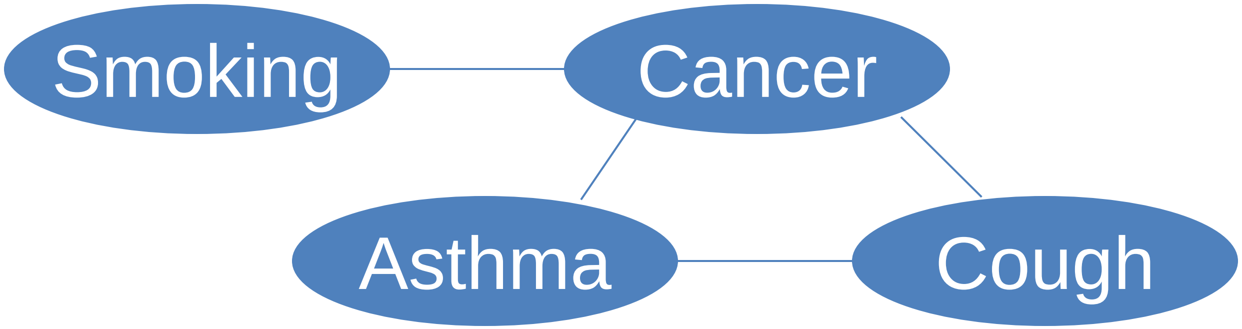
- **Atoms:** Symbols representing propositions
- **Logical connectives:** ¬, ∧, ∨, etc.
- **Knowledge base:** Set of formulas
- **World:** Truth assignment to all atoms
- Every KB can be converted to **CNF**
  - *CNF:* Conjunction of clauses
  - *Clause:* Disjunction of literals
  - *Literal:* Atom or its negation
- **Entailment:** Does KB entail query?

First-Order Logic

- **Atom:** Predicate(Variables,Constants)
  E.g.: $Friends(Anna, x)$
- **Ground atom:** All arguments are constants
- **Quantifiers:** $\forall$, $\exists$
- **This talk:** Finite, Herbrand interpretations

Markov Networks

- **Undirected** graphical models



- Potential functions defined over cliques

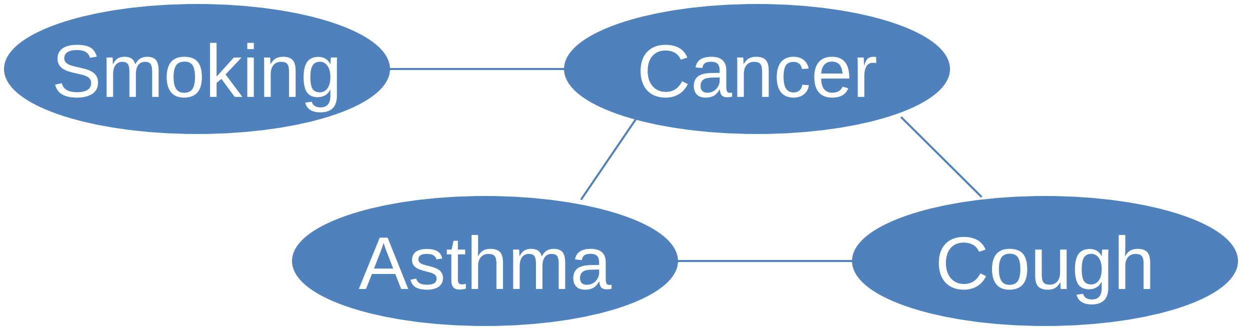
$$P(x) = \frac{1}{Z} \prod_c \Phi_c(x_c)$$

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

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

Markov Networks

- **Undirected** graphical models



- Log-linear model:

$$P(x) = \frac{1}{Z} \exp\left(\sum_i \underbrace{w_i}_{\text{Weight of Feature } i} \underbrace{f_i(x)}_{\text{Feature } i} \right)$$

$$f_1(\text{Smoking}, \text{Cancer}) = \begin{cases} 1 & \text{if } \neg \text{Smoking} \vee \text{Cancer} \\ 0 & \text{otherwise} \end{cases}$$

$$w_1 = 0.51$$

Probabilistic Knowledge Bases

PKB = Set of formulas and their probabilities
+ Consistency + Maximum entropy
= Set of formulas and their weights
= Set of formulas and their potentials
(1 if formula true, ϕ_i if formula false)

$$\boxed{P(world) = \frac{1}{Z}\prod_i \phi_i^{n_i(world)}}$$

Markov Logic

- A Markov Logic Network (MLN) is a set of pairs (F, w) where
  - F is a formula in first-order logic
  - w is a real number
- An MLN defines a Markov network with
  - One node for each grounding of each predicate in the MLN
  - One feature for each grounding of each formula F in the MLN, with the corresponding weight w

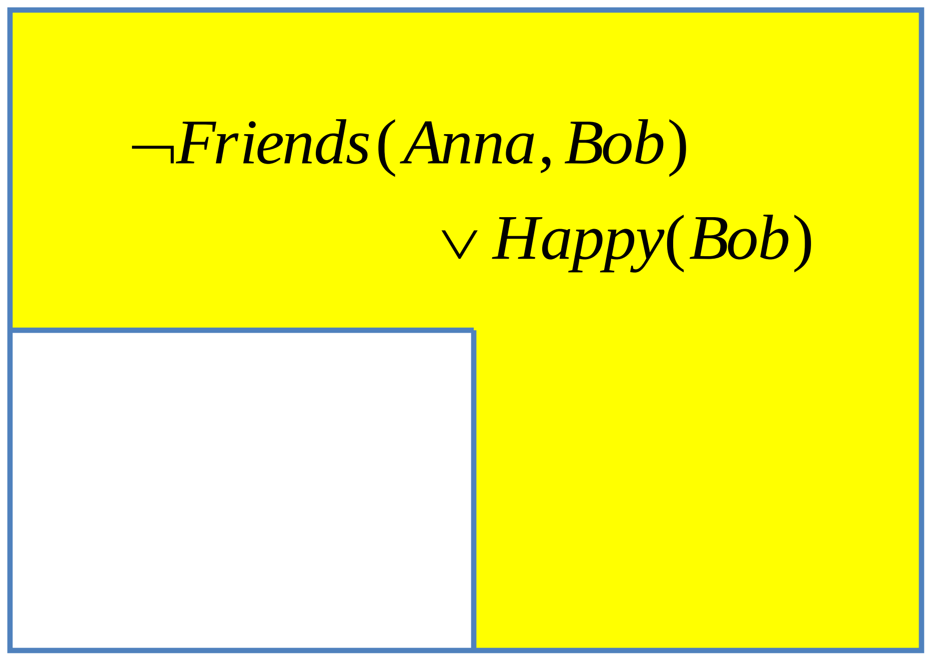
Example

	$\neg Happy(Bob)$	$Happy(Bob)$
$\neg Friends(Anna, Bob)$		
$Friends(Anna, Bob)$		

Example

$\neg Friends(Anna, Bob)$

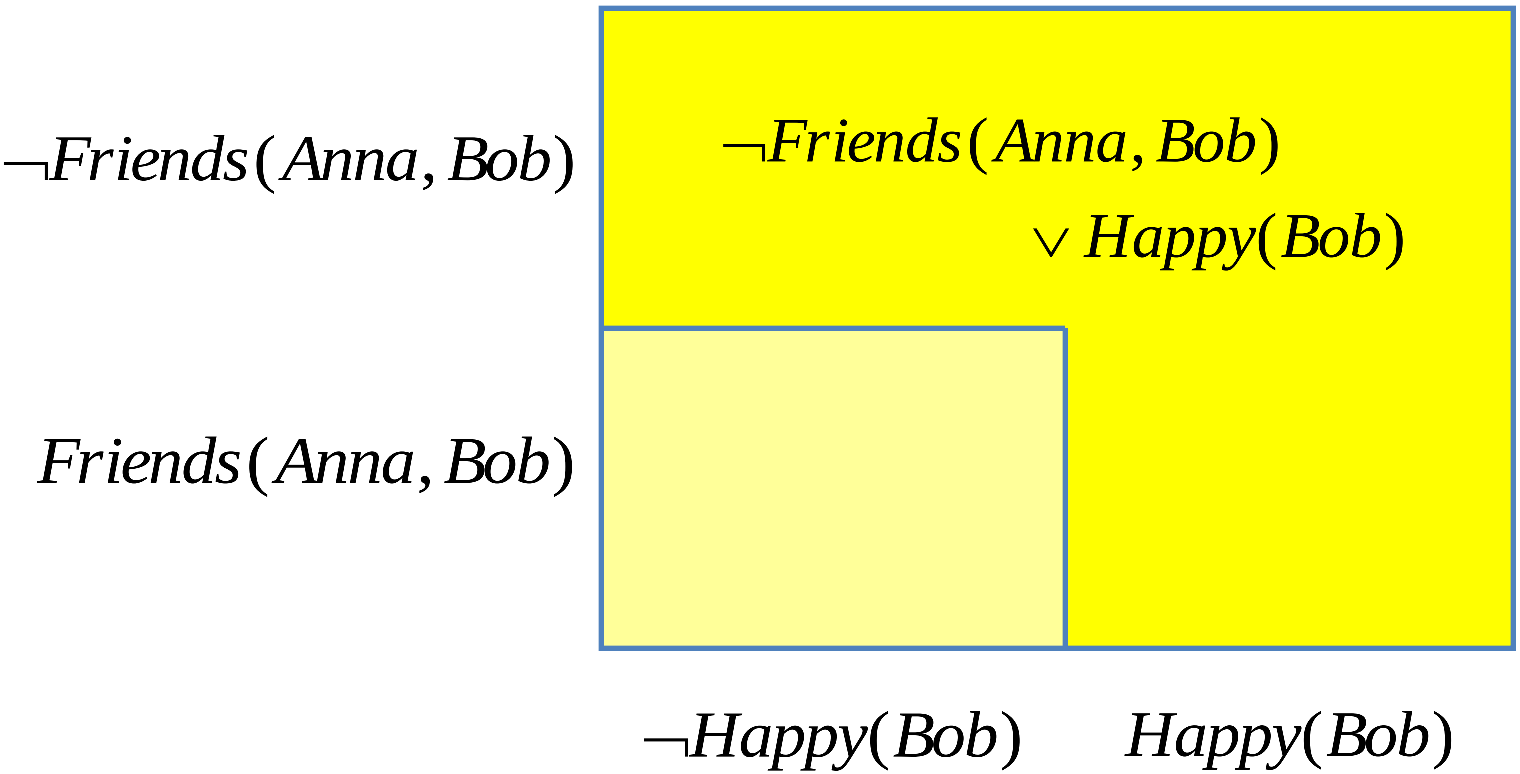
$Friends(Anna, Bob)$



$\neg Happy(Bob)$ $Happy(Bob)$

Example

$$P(\neg Friends(Anna, Bob) \vee Happy(Bob)) = 0.8$$



Example

$$\Phi(\neg Friends(Anna, Bob) \vee Happy(Bob)) = 1$$

$$\Phi(Friends(Anna, Bob) \wedge \neg Happy(Bob)) = 0.75$$

	$\neg Happy(Bob)$	$Happy(Bob)$
$\neg Friends(Anna, Bob)$	1	1
$Friends(Anna, Bob)$	0.75	1

Example

$$w(\Phi(\neg Friends(Anna, Bob) \vee Happy(Bob)))$$
$$= \log(1/0.75) = 0.29$$

	$\neg Happy(Bob)$	$Happy(Bob)$
$\neg Friends(Anna, Bob)$	1	1
$Friends(Anna, Bob)$	0.75	1

Overview

- Representation
- **Inference**
- Learning
- Applications
- Discussion

Theorem Proving

TP(KB, $Query$)

$KB_Q \leftarrow KB \cup \{\neg\, Query\}$

return $\neg$SAT(CNF(KB_Q))

Satisfiability (DPLL)

SAT(*CNF*)

if *CNF* is empty **return** *True*

if *CNF* contains empty clause **return** *False*

choose an atom *A*

return SAT(CNF(*A*)) V SAT(CNF(¬*A*))

First-Order Theorem Proving

- **Propositionalization**
  1. Form all possible ground atoms
  2. Apply propositional theorem prover
- **Lifted Inference: Resolution**
  - Resolve pairs of clauses until empty clause derived
  - Unify literals by substitution, e.g.: $x = Bob$ unifies $Friends(Anna, x)$ and $Friends(Anna, Bob)$

$$\frac{\begin{array}{c}\neg Friends(Anna, x) \vee Happy(x) \\ Friends(Anna, Bob)\end{array}}{Happy(Bob)}$$

Probabilistic Theorem Proving

Given Probabilistic knowledge base K
Query formula Q

Output $P(Q|K)$

Weighted Model Counting

- ModelCount(CNF) = # worlds that satisfy CNF
- Assign a weight to each literal
- Weight(world) = $\prod$ weights(true literals)
- Weighted model counting:

  Given CNF *C* and literal weights *W*

  Output $\sum$ weights(worlds that satisfy *C*)

PTP is reducible to lifted WMC

Example

$$Friends(Anna, Bob)$$

	$\neg Happy(Bob)$	$Happy(Bob)$
$\neg Friends(Anna, Bob)$		
$Friends(Anna, Bob)$	0.75	1

Example

$$P(Happy(Bob)\,|\,Friends(Anna,Bob)) = \frac{1}{1+0.75} \approx 0.57$$

	$\neg Happy(Bob)$	$Happy(Bob)$
$\neg Friends(Anna,Bob)$		
$Friends(Anna,Bob)$	0.75	1

Example

If $P(\neg Friends(Anna, Bob) \vee Happy(Bob)) = 0.8$

Then $P(Happy(Bob) \mid Friends(Anna, Bob)) = \dfrac{1}{1+0.75} \approx 0.57$

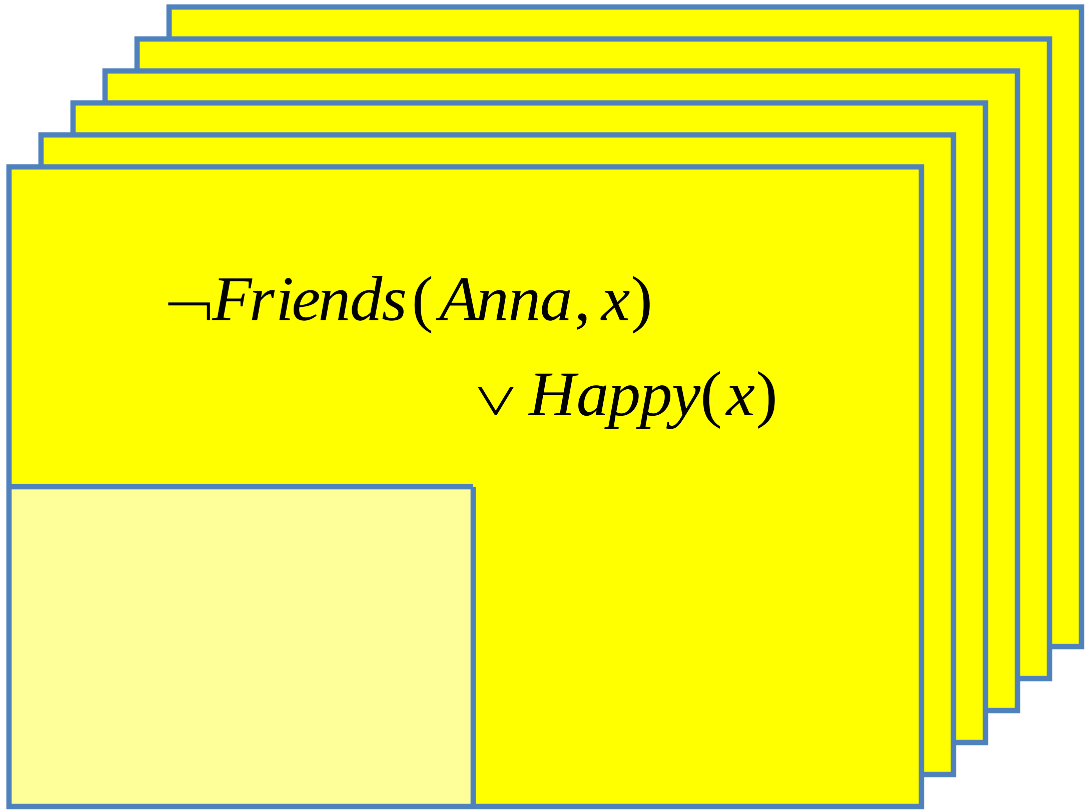
	$\neg Happy(Bob)$	$Happy(Bob)$
$\neg Friends(Anna, Bob)$		
$Friends(Anna, Bob)$	0.75	1

Example

$$P(\neg Friends(Anna,x) \vee Happy(x)) = 0.8$$

$\neg Friends(Anna,x)$

$Friends(Anna,x)$

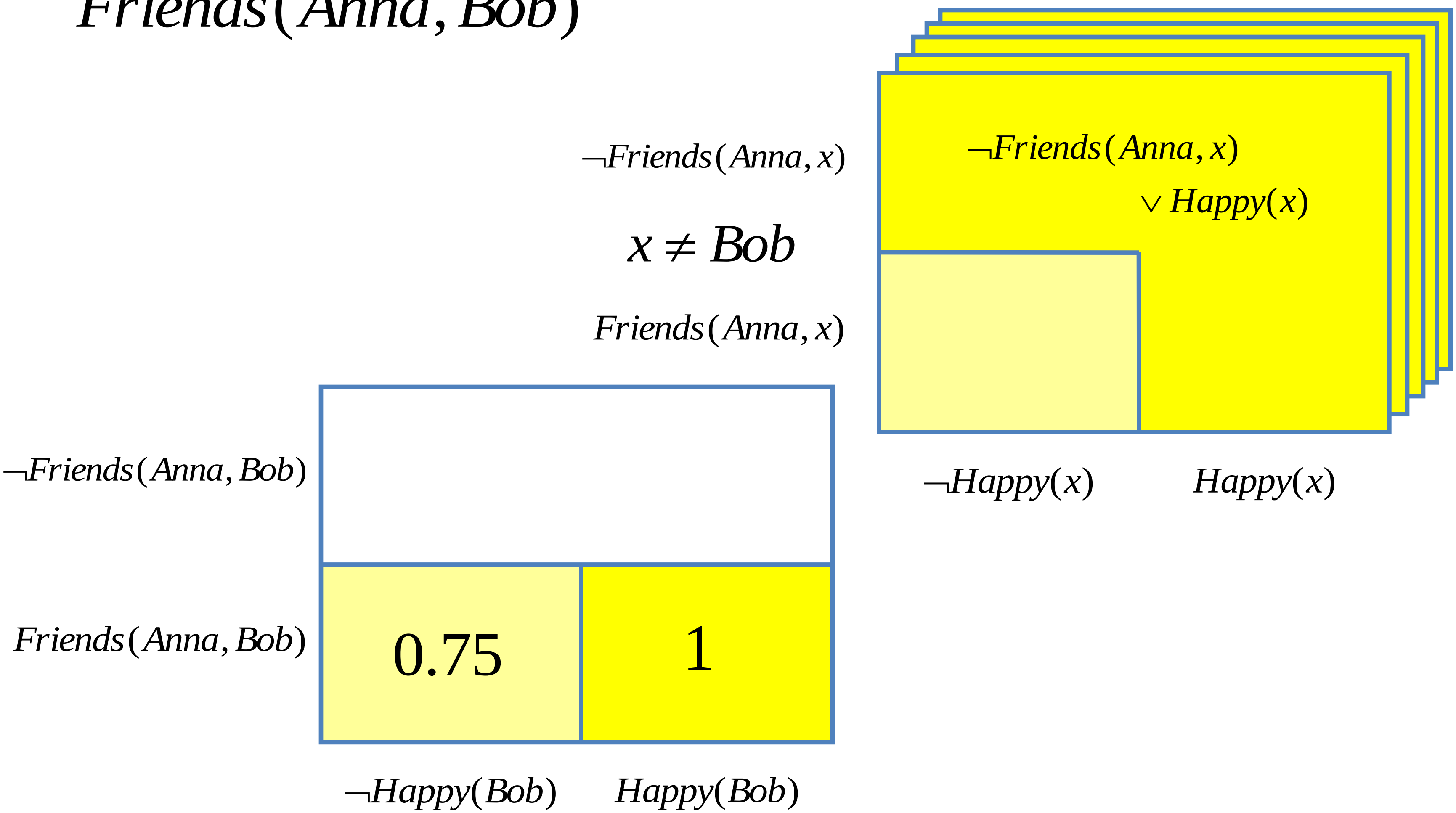


$\neg Happy(x)$ $Happy(x)$

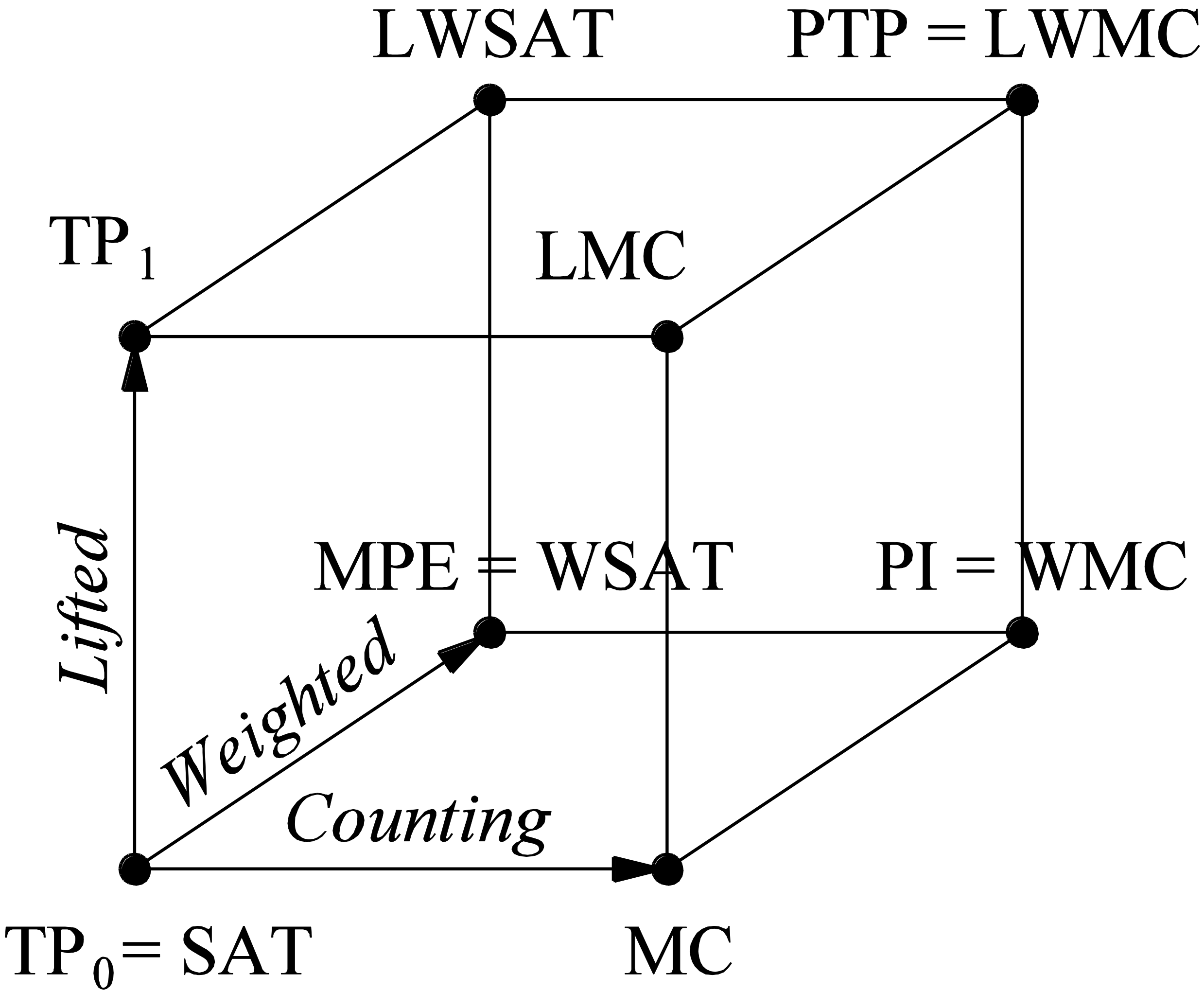
Example

$$P(\neg Friends(Anna, x) \vee Happy(x)) = 0.8$$

$$Friends(Anna, Bob)$$



Inference Problems



Propositional Case

- All conditional probabilities are ratios of partition functions:

$$P(Query \mid PKB) = \frac{\sum_{worlds} 1_{Query}(world) \prod_i \Phi_i(world)}{Z(PKB)}$$

$$= \frac{Z(PKB \cup \{(Query, 0)\})}{Z(PKB)}$$

- All partition functions can be computed by weighted model counting

Conversion to CNF + Weights

WCNF(*PKB*)

for all $(F_i, \Phi_i) \in PKB$ s.t. $\Phi_i > 0$ **do**

$PKB \leftarrow PKB \cup \{(F_i \Leftrightarrow A_i, 0)\} \setminus \{(F_i, \Phi_i)\}$

$CNF \leftarrow \mathrm{CNF}(PKB)$

for all $\neg A_i$ literals do $W_{\neg Ai} \leftarrow \Phi_i$

for all other literals L do $w_L \leftarrow 1$

return (*CNF*, *weights*)

Probabilistic Theorem Proving

PTP(*PKB*, *Query*)

$PKB_Q \leftarrow PKB \cup \{(Query, 0)\}$

return WCNF(WCNF(PKB_Q))

/ WMC(WCNF(*PKB*))

Probabilistic Theorem Proving

PTP(*PKB*, *Query*)

$PKB_Q \leftarrow PKB \cup \{(Query, 0)\}$

return WMC(WCNF(PKB_Q)) / WMC(WCNF(*PKB*))

Compare:

TP(*KB*, *Query*)

$KB_Q \leftarrow KB \cup \{\neg\, Query\}$

return ¬SAT(CNF(KB_Q))

Weighted Model Counting

WMC(*CNF*, *weights*)

if all clauses in *CNF* are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})$

Base Case

if *CNF* has empty unsatisfied clause **return** 0

Weighted Model Counting

WMC(*CNF*, *weights*)

if all clauses in CNF are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})$

if *CNF* has empty unsatisfied clause **return** 0

if *CNF* can be partitioned into CNFs $C_1, \ldots, C_k$ sharing no atoms

return $\prod_{i=1}^{k} WMC(C_i, weights)$

Decomp. Step

Weighted Model Counting

WMC(*CNF*, *weights*)

if all clauses in *CNF* are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})$

if *CNF* has empty unsatisfied clause **return** 0

if *CNF* can be partitioned into CNFs $C_1, \ldots, C_k$ sharing no atoms

return $\prod_{i=1}^{k} WMC(C_i, weights)$

choose an atom *A*

return $w_A WMC(CNF \mid A, weights)$
$+ w_{\neg A} WMC(CNF \mid \neg A, weights)$

Splitting Step

First-Order Case

- PTP schema remains the same
- Conversion of PKB to hard CNF and weights:
  New atom in $F_i \Leftrightarrow A_i$ is now
  $Predicate_i(variables\ in\ F_i,\ constants\ in\ F_i)$
- New argument in WMC:
  Set of substitution constraints of the form
  $x = A,\ x \neq A,\ x = y,\ x \neq y$
- Lift each step of WMC

Lifted Weighted Model Counting

LWMC(*CNF, substs, weights*)

if all clauses in *CNF* are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})^{n_A(substs)}$

if *CNF* has empty unsatisfied clause **return** 0

Base Case

Lifted Weighted Model Counting

LWMC(*CNF, substs, weights*)

if all clauses in *CNF* are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})^{n_A(substs)}$

if *CNF* has empty unsatisfied clause **return** 0

if there exists a lifted decomposition of *CNF*

return $\prod_{i=1}^{k} [LWMC(CNF_{i,1}, substs, weights)]^{m_i}$

Decomp. Step

Lifted Weighted Model Counting

LWMC(*CNF, substs, weights*)

if all clauses in *CNF* are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})^{n_A(substs)}$

if *CNF* has empty unsatisfied clause **return** 0

if there exists a lifted decomposition of *CNF*

return $\prod_{i=1}^{k} [LWMC(CNF_{i,1}, substs, weights)]^{m_i}$

choose an atom *A*

return

$$\sum_{i=1}^{l} n_i w_A^{t_i} w_{\neg A}^{f_i} LWMC(CNF \mid \sigma_j, substs_j, weights)$$

Splitting Step

Extensions

- Unit propagation, etc.
- Caching / Memoization
- Knowledge-based model construction

Approximate Inference

WMC(*CNF*, *weights*)

  if all clauses in *CNF* are satisfied

    return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})$

  if *CNF* has empty unsatisfied clause **return** 0

  if *CNF* can be partitioned into CNFs $C_1, \ldots, C_k$ sharing no atoms

    return $\prod_{i=1}^{k} WMC(C_i, weights)$

  choose an atom *A*

  return $\frac{w_A}{Q(A|CNF, weights)} WMC(CNF \mid A, weights)$

    with probability $Q(A|CNF, weights)$, etc.

Splitting Step

MPE Inference

- Replace sums by maxes
- Use branch-and-bound for efficiency
- Do traceback

More on Sunday at Noon

Session on First-Order Inference

- *Probabilistic Theorem Proving*
  V. Gogate and P. Domingos

- *Inference in Probabilistic Logic Programs Using Weighted CNF*
  D. Fierens, G. van den Broeck, I. Thon, B. Gutmann and L. de Raedt

Even More on Monday

IJCAI-11 Tutorial on Lifted Inference in Probabilistic Logical Models

- Eyal Amir
- Pedro Domingos
- Lise Getoor
- Kristian Kersting
- Sriraam Natarajan
- David Poole
- Rodrigo de S. Braz
- Prithviraj Sen

Overview

- Representation
- Inference
- **Learning**
- Applications
- Discussion

Learning

- Data is a relational database
- Closed world assumption (if not: EM)
- Learning parameters (weights)
  - Generatively
  - Discriminatively
- Learning structure (formulas)

Generative Weight Learning

- Maximize likelihood
- Use gradient ascent or L-BFGS
- No local maxima

$$\frac{\partial}{\partial w_i} \log P_w(x) = \boxed{n_i(x)} - \boxed{E_w\left[n_i(x)\right]}$$

No. of true groundings of clause *i* in data

Expected no. true groundings according to model

- Requires inference at each step (slow!)

Pseudo-Likelihood

$$PL(x) \equiv \prod_i P(x_i \mid neighbors(x_i))$$

- Likelihood of each variable given its neighbors in the data [Besag, 1975]
- Does not require inference at each step
- Consistent estimator
- Widely used in vision, spatial statistics, etc.
- But PL parameters may not work well for long inference chains

Discriminative Weight Learning

- Maximize conditional likelihood of query (y) given evidence (x)

$$\frac{\partial}{\partial w_i} \log P_w(y \mid x) = \boxed{n_i(x, y)} - \boxed{E_w\left[n_i(x, y)\right]}$$

No. of true groundings of clause *i* in data

Expected no. true groundings according to model

- Expected counts can be approximated by counts in MAP state of y given x

Voted Perceptron

- Originally proposed for training HMMs discriminatively [Collins, 2002]
- Assumes network is linear chain

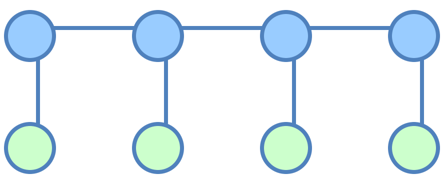
$w_i \leftarrow 0$

for $t \leftarrow 1$ **to** T **do**

$\quad y_{MAP} \leftarrow \text{Viterbi}(x)$

$\quad w_i \leftarrow w_i + \eta\, [\text{count}_\text{i}(y_{Data}) - \text{count}_\text{i}(y_{MAP})]$

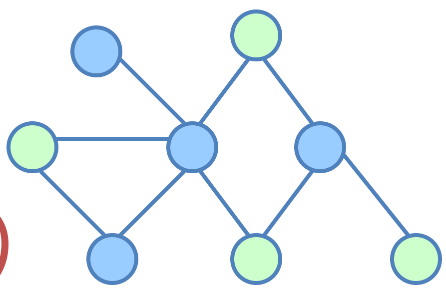
return $\sum_t w_i$ / T



Voted Perceptron for MLNs

- HMMs are special case of MLNs
- Replace Viterbi by prob. theorem proving
- Network can now be arbitrary graph

$w_i \leftarrow 0$
for $t \leftarrow 1$ **to** T **do**
    $y_{MAP} \leftarrow PTP(MLN\ U\ \{x\}, y)$
    $w_i \leftarrow w_i + \eta\ [\text{count}_\text{i}(y_{Data}) - \text{count}_\text{i}(y_{MAP})]$
return $\sum_t w_i\ /\ \text{T}$



Structure Learning

- Generalizes feature induction in Markov nets
- Any inductive logic programming approach can be used, but . . .
- Goal is to induce any clauses, not just Horn
- Evaluation function should be likelihood
- Requires learning weights for each candidate
- Turns out not to be bottleneck
- Bottleneck is counting clause groundings
- Solution: Subsampling

Structure Learning

- **Initial state:** Unit clauses or hand-coded KB
- **Operators:** Add/remove literal, flip sign
- **Evaluation function:**
  Pseudo-likelihood + Structure prior
- **Search:**
  - Beam, shortest-first [Kok & Domingos, 2005]
  - Bottom-up [Mihalkova & Mooney, 2007]
  - Relational pathfinding [Kok & Domingos, 2009, 2010]

Alchemy

Open-source software including:

- Full first-order logic syntax
- MAP and marginal/conditional inference
- Generative & discriminative weight learning
- Structure learning
- Programming language features

alchemy.cs.washington.edu

	Alchemy	Prolog	BUGS
Represent-ation	F.O. Logic + Markov nets	Horn clauses	Bayes nets
Inference	Probabilistic thm. proving	Theorem proving	Gibbs sampling
Learning	Parameters & structure	No	Params.
Uncertainty	Yes	No	Yes
Relational	Yes	Yes	No

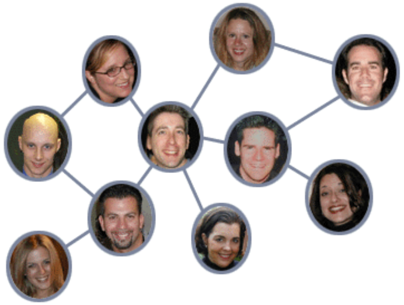
Overview

- Representation
- Inference
- Learning
- **Applications**
- Discussion

Applications to Date

- Natural language processing
- Information extraction
- Entity resolution
- Link prediction
- Collective classification
- Social network analysis
- Robot mapping
- Activity recognition
- Scene analysis
- Computational biology
- Probabilistic Cyc
- Personal assistants
- Etc.




Information Extraction

Parag Singla and Pedro Domingos, “Memory-Efficient Inference in Relational Domains” (AAAI-06).

Singla, P., & Domingos, P. (2006). Memory-efficent inference in relatonal domains. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 500-505). Boston, MA: AAAI Press.

H. Poon & P. Domingos, Sound and Efficient Inference with Probabilistic and Deterministic Dependencies”, in Proc. AAAI-06, Boston, MA, 2006.

P. Hoifung (2006). Efficent inference. In Proceedings of the Twenty-First National Conference on Artificial Intelligence.

Author

Title

Venue

Segmentation

Parag Singla and Pedro Domingos, “Memory-Efficient Inference in Relational Domains” (AAAI-06).

Singla, P., & Domingos, P. (2006). Memory-efficent inference in relatonal domains. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 500-505). Boston, MA: AAAI Press.

H. Poon & P. Domingos, Sound and Efficient Inference with Probabilistic and Deterministic Dependencies”, in Proc. AAAI-06, Boston, MA, 2006.

P. Hoifung (2006). Efficent inference. In Proceedings of the Twenty-First National Conference on Artificial Intelligence.

Entity Resolution

Parag Singla and Pedro Domingos, "Memory-Efficient Inference in Relational Domains" (AAAI-06).

Singla, P., & Domingos, P. (2006). Memory-efficent inference in relational domains. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 500-505). Boston, MA: AAAI Press.

H. Poon & P. Domingos, Sound and Efficient Inference with Probabilistic and Deterministic Dependencies", in Proc. AAAI-06, Boston, MA, 2006.

P. Hoifung (2006). Efficent inference. In Proceedings of the Twenty-First National Conference on Artificial Intelligence.

Entity Resolution

Parag Singla and Pedro Domingos, "Memory-Efficient Inference in Relational Domains" (AAAI-06).

Singla, P., & Domingos, P. (2006). Memory-efficent inference in relational domains. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 500-505). Boston, MA: AAAI Press.

H. Poon & P. Domingos, Sound and Efficient Inference with Probabilistic and Deterministic Dependencies", in Proc. AAAI-06, Boston, MA, 2006.

P. Hoifung (2006). Efficent inference. In Proceedings of the Twenty-First National Conference on Artificial Intelligence.

State of the Art

- Segmentation
  - HMM (or CRF) to assign each token to a field
- Entity resolution
  - Logistic regression to predict same field/citation
  - Transitive closure
- Alchemy implementation: Seven formulas

Types and Predicates

```
token = {Parag, Singla, and, Pedro, ...}
field = {Author, Title, Venue}
citation = {C1, C2, ...}
position = {0, 1, 2, ...}

Token(token, position, citation)
InField(position, field, citation)
SameField(field, citation, citation)
SameCit(citation, citation)
```

Types and Predicates

```
token = {Parag, Singla, and, Pedro, ...}
field = {Author, Title, Venue, ...}
citation = {C1, C2, ...}
position = {0, 1, 2, ...}
```

Optional

```
Token(token, position, citation)
InField(position, field, citation)
SameField(field, citation, citation)
SameCit(citation, citation)
```

Types and Predicates

```
token = {Parag, Singla, and, Pedro, ...}
field = {Author, Title, Venue}
citation = {C1, C2, ...}
position = {0, 1, 2, ...}

Token(token, position, citation)     ← Evidence
InField(position, field, citation)
SameField(field, citation, citation)
SameCit(citation, citation)
```

Types and Predicates

```
token = {Parag, Singla, and, Pedro, ...}
field = {Author, Title, Venue}
citation = {C1, C2, ...}
position = {0, 1, 2, ...}

Token(token, position, citation)
InField(position, field, citation)
SameField(field, citation, citation)
SameCit(citation, citation)
```

Query

Formulas

```
Token(+t,i,c) => InField(i,+f,c)
InField(i,+f,c) <=> InField(i+1,+f,c)
f != f’ => (!InField(i,+f,c) v !InField(i,+f’,c))

Token(+t,i,c) ^ InField(i,+f,c) ^ Token(+t,i’,c’)
   ^ InField(i’,+f,c’) => SameField(+f,c,c’)
SameField(+f,c,c’) <=> SameCit(c,c’)
SameField(f,c,c’) ^ SameField(f,c’,c”)
   => SameField(f,c,c”)
SameCit(c,c’) ^ SameCit(c’,c”) => SameCit(c,c”)
```

Formulas

```
Token(+t,i,c) => InField(i,+f,c)
InField(i,+f,c) <=> InField(i+1,+f,c)
f != f' => (!InField(i,+f,c) v !InField(i,+f',c))

Token(+t,i,c) ^ InField(i,+f,c) ^ Token(+t,i',c')
   ^ InField(i',+f,c') => SameField(+f,c,c')
SameField(+f,c,c') <=> SameCit(c,c')
SameField(f,c,c') ^ SameField(f,c',c")
   => SameField(f,c,c")
SameCit(c,c') ^ SameCit(c',c") => SameCit(c,c")
```

Formulas

```
Token(+t,i,c) => InField(i,+f,c)
InField(i,+f,c) <=> InField(i+1,+f,c)
f != f' => (!InField(i,+f,c) v !InField(i,+f',c))

Token(+t,i,c) ^ InField(i,+f,c) ^ Token(+t,i',c')
   ^ InField(i',+f,c') => SameField(+f,c,c')
SameField(+f,c,c') <=> SameCit(c,c')
SameField(f,c,c') ^ SameField(f,c',c")
   => SameField(f,c,c")
SameCit(c,c') ^ SameCit(c',c") => SameCit(c,c")
```

Formulas

```
Token(+t,i,c) => InField(i,+f,c)
InField(i,+f,c) <=> InField(i+1,+f,c)
f != f' => (!InField(i,+f,c) v !InField(i,+f',c))

Token(+t,i,c) ^ InField(i,+f,c) ^ Token(+t,i',c')
   ^ InField(i',+f,c') => SameField(+f,c,c')
SameField(+f,c,c') <=> SameCit(c,c')
SameField(f,c,c') ^ SameField(f,c',c")
   => SameField(f,c,c")
SameCit(c,c') ^ SameCit(c',c") => SameCit(c,c")
```

Formulas

```
Token(+t,i,c) => InField(i,+f,c)
InField(i,+f,c) <=> InField(i+1,+f,c)
f != f′ => (!InField(i,+f,c) v !InField(i,+f′,c))

Token(+t,i,c) ^ InField(i,+f,c) ^ Token(+t,i′,c′)
   ^ InField(i′,+f,c′) => SameField(+f,c,c′)
SameField(+f,c,c′) <=> SameCit(c,c′)
SameField(f,c,c′) ^ SameField(f,c′,c″)
   => SameField(f,c,c″)
SameCit(c,c′) ^ SameCit(c′,c″) => SameCit(c,c″)
```

Formulas

```
Token(+t,i,c) => InField(i,+f,c)
InField(i,+f,c) <=> InField(i+1,+f,c)
f != f′ => (!InField(i,+f,c) v !InField(i,+f′,c))

Token(+t,i,c) ^ InField(i,+f,c) ^ Token(+t,i′,c′)
   ^ InField(i′,+f,c′) => SameField(+f,c,c′)
SameField(+f,c,c′) <=> SameCit(c,c′)
SameField(f,c,c′) ^ SameField(f,c′,c″)
   => SameField(f,c,c″)
SameCit(c,c′) ^ SameCit(c′,c″) => SameCit(c,c″)
```

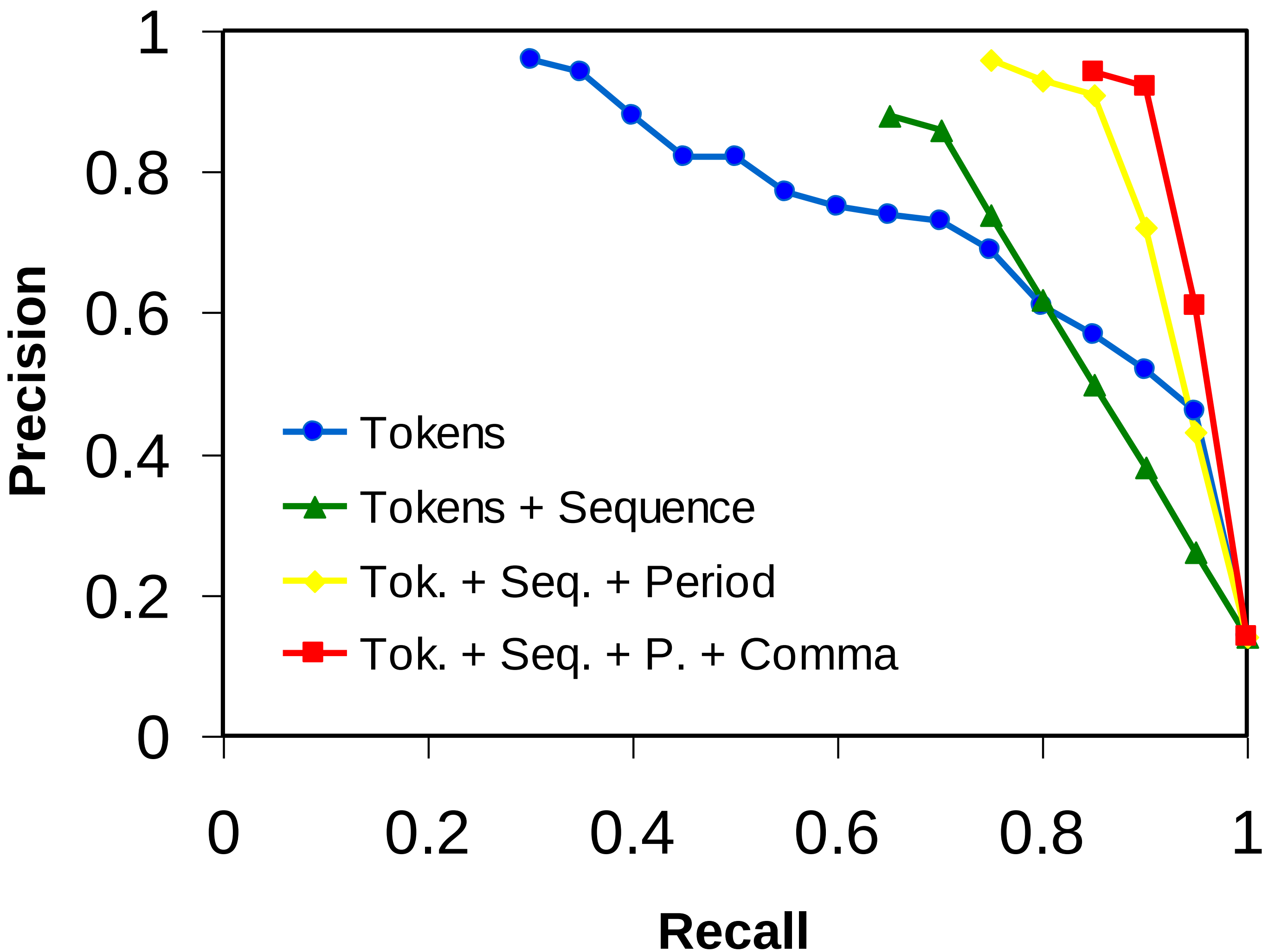
Formulas

```
Token(+t,i,c) => InField(i,+f,c)
InField(i,+f,c) <=> InField(i+1,+f,c)
f != f’ => (!InField(i,+f,c) v !InField(i,+f’,c))

Token(+t,i,c) ^ InField(i,+f,c) ^ Token(+t,i’,c’)
   ^ InField(i’,+f,c’) => SameField(+f,c,c’)
SameField(+f,c,c’) <=> SameCit(c,c’)
SameField(f,c,c’) ^ SameField(f,c’,c”)
   => SameField(f,c,c”)
SameCit(c,c’) ^ SameCit(c’,c”) => SameCit(c,c”)
```

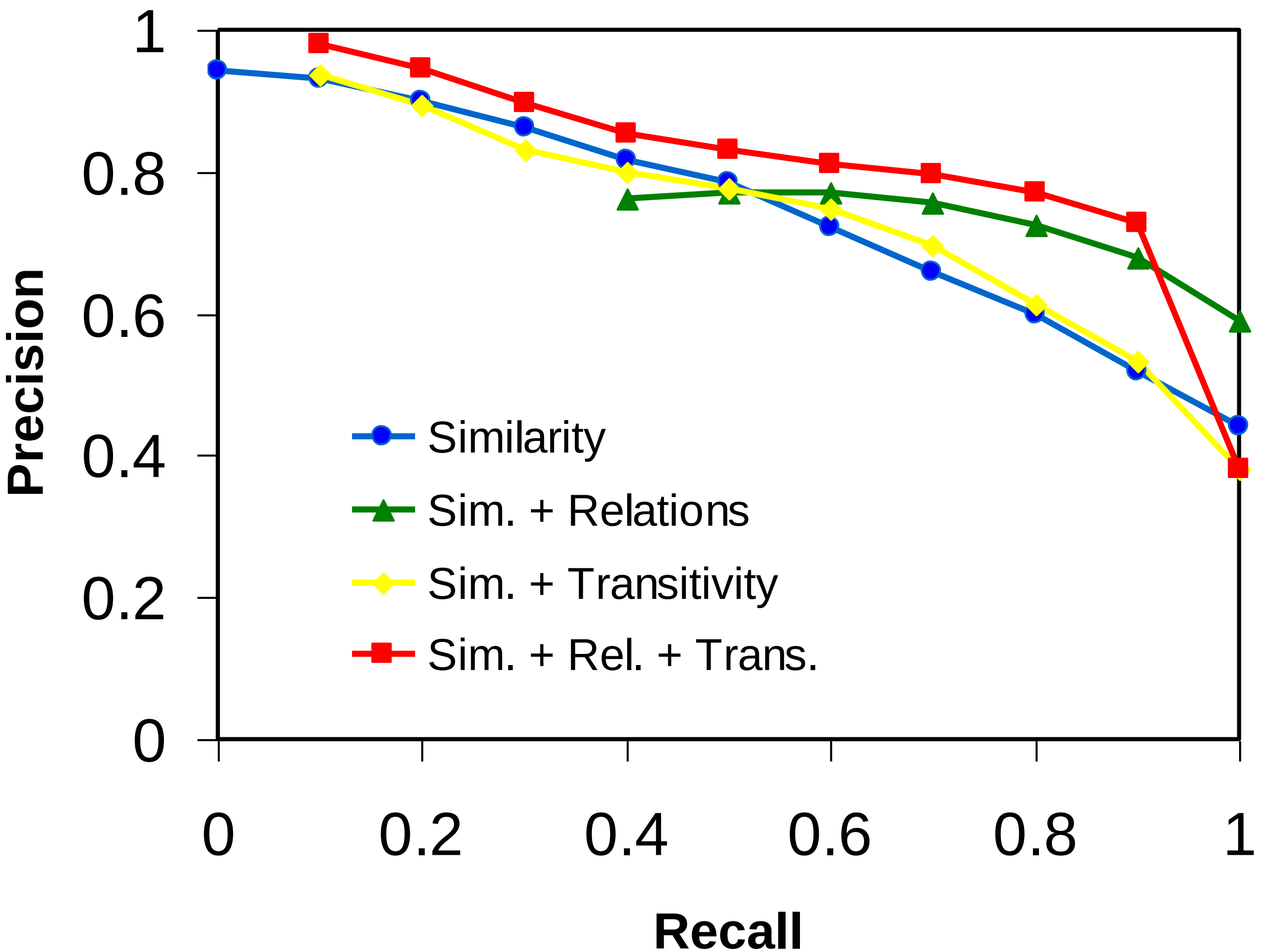
Formulas

```
Token(+t,i,c) => InField(i,+f,c)
InField(i,+f,c) ^ !Token(“.”,i,c) <=> InField(i+1,+f,c)
f != f’ => (!InField(i,+f,c) v !InField(i,+f’,c))

Token(+t,i,c) ^ InField(i,+f,c) ^ Token(+t,i’,c’)
   ^ InField(i’,+f,c’) => SameField(+f,c,c’)
SameField(+f,c,c’) <=> SameCit(c,c’)
SameField(f,c,c’) ^ SameField(f,c’,c”)
   => SameField(f,c,c”)
SameCit(c,c’) ^ SameCit(c’,c”) => SameCit(c,c”)
```

Results: Segmentation on Cora



Results:
Matching Venues on Cora



Overview

- Representation
- Inference
- Learning
- Applications
- **Discussion**

Foundations for Probabilistic Models

- **Graphs are not enough**
- **We need logic**

Logical Models vs. Graphical Models (I)

	Graphical models	Logical models
Required by probability theory	No	Yes
Representable distributions	All (BNs) Positive (MNs)	All
Context-free independences	Some	All
Context-specific independences	None	All
Normalization constraints	Some	All

Logical Models vs. Graphical Models (II)

	Graphical models	Logical models
Inference	Exp(treewidth)	Circuit complexity
Visual aid	Yes	No
Densely connected distrs.	Unreadable	Readable
First-order	Plates	All
Lifted inference	No	Yes
Available technology	Lots, used	Lots, unused