Learning, Logic, and Probability: A Unified View

Pedro Domingos
Dept. Computer Science & Eng.
University of Washington

(Joint work with Stanley Kok, Matt Richardson and Parag Singla)

Overview

- Motivation
- Background
- Markov logic networks
- Inference in MLNs
- Learning MLNs
- Experiments
- Discussion

The Way Things Were

- First-order logic is the foundation of computer science
  - **Problem:** Logic is too brittle
- Programs are written by hand
  - **Problem:** Too expensive, not scalable

The Way Things Are

- **Probability** overcomes the brittleness
- **Machine learning** automates programming
- Their use is spreading rapidly
- **Problem:** For the most part, they apply only to vectors
- What about structured objects, class hierarchies, relational databases, etc.?

The Way Things Will Be

- Learning and probability applied to the full expressiveness of first-order logic
- **This talk:** First approach that does this
- Benefits: Robustness, reusability, scalability, reduced cost, human-friendliness, etc.
- Learning and probability will become everyday tools of computer scientists
- Many things will be practical that weren’t before

State of the Art

- **Learning:** Decision trees, SVMs, etc.
- **Logic:** Resolution, WalkSat, Prolog, description logics, etc.
- **Probability:** Bayes nets, Markov nets, etc.
- **Learning + Logic:** Inductive logic prog. (ILP)
- **Learning + Probability:** EM, K2, etc.
- **Logic + Probability:** Halpern, Bacchus, KBMC, PRISM, etc.

Learning + Logic + Probability

- Recent (last five years)
- Workshops: SRL [‘00, ‘03, ‘04], MRDM [‘02, ‘03, ‘04]
- Special issues: SIGKDD, Machine Learning
- All approaches so far use only **subsets** of first-order logic
  - Horn clauses (e.g., SLPs [Cussens, 2001; Muggleton, 2002])
  - Description logics (e.g., PRMs [Friedman et al., 1999])
  - Database queries (e.g., RMNs [Taskar et al., 2002])

Questions

- Is it possible to combine the full power of first-order logic and probabilistic graphical models in a single representation?
- Is it possible to reason and learn efficiently in such a representation?

Markov Logic Networks

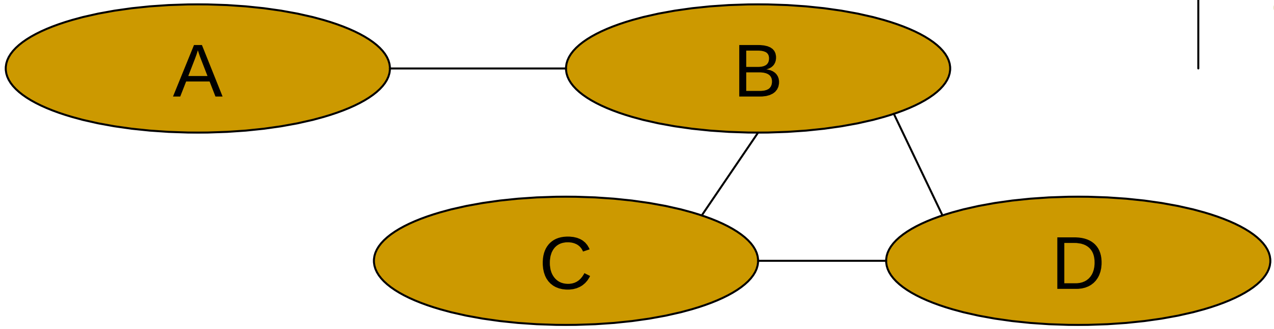
- **Syntax:** First-order logic + Weights
- **Semantics:** Templates for Markov nets
- **Inference:** KBMC + MCMC
- **Learning:** ILP + Pseudo-likelihood
- **Special cases:** Collective classification, link prediction, link-based clustering, social networks, object identification, etc.

Overview

- Motivation
- Background
- Markov logic networks
- Inference in MLNs
- Learning MLNs
- Experiments
- Discussion

Markov Networks

- **Undirected** graphical models



- Potential functions defined over cliques

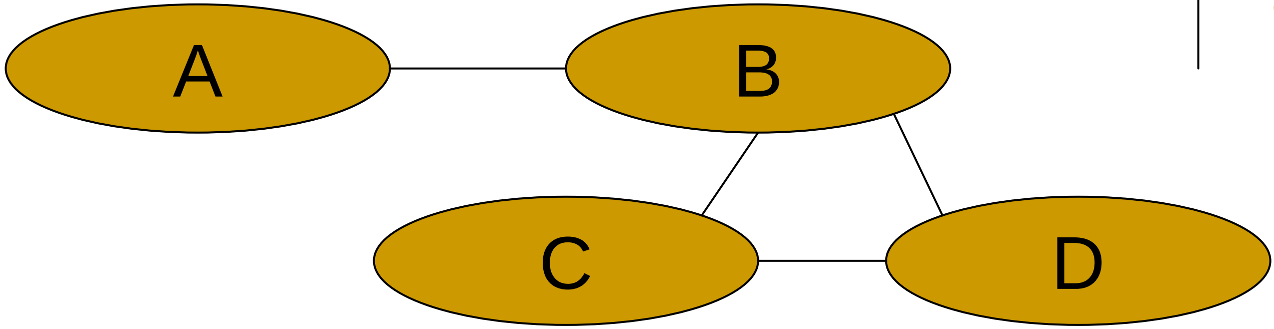
$$P(X) = \frac{1}{Z} \prod_c \Phi_c(X) \qquad\qquad Z = \sum_X \prod_c \Phi_c(X)$$

$$\Phi(A,B) = \begin{cases} 3.7 & \text{if A and B} \\ 2.1 & \text{if A and } \overline{\text{B}} \\ 0.7 & \text{otherwise} \end{cases}$$

$$\Phi(B,C,D) = \begin{cases} 2.3 & \text{if B and } \overline{\text{C}} \text{ and D} \\ 5.1 & \text{otherwise} \end{cases}$$

Markov Networks

- **Undirected** graphical models



- Potential functions defined over cliques

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

$$f(A,B)=\begin{cases}1 & \text{if A and B}\\ 0 & \text{otherwise}\end{cases}$$

$$f(B,C,D)=\begin{cases}1 & \text{if B and }\overline{\text{C}}\text{ and D}\\ 0 & \end{cases}$$

First-Order Logic

- Constants, variables, functions, predicates
  E.g.: Anna, X, mother_of(X), friends(X, Y)
- Grounding: Replace all variables by constants
  E.g.: friends (Anna, Bob)
- **World** (model, interpretation):
  Assignment of truth values to all ground predicates

Example of First-Order KB

Smoking causes cancer

Friends either both smoke or both don’t smoke

Example of First-Order KB

$$\forall x\ Smokes(x) \Rightarrow Cancer(x)$$

$$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$$

Overview

- Motivation
- Background
- Markov logic networks
- Inference in MLNs
- Learning MLNs
- Experiments
- Discussion

Markov Logic Networks

- A logical KB is a set of **hard constraints** on the set of possible worlds
- Let's make them **soft constraints**:
  When a world violates a formula,
  It becomes less probable, not impossible
- Give each formula a **weight**
  (Higher weight ⇒ Stronger constraint)

$$P(world) \propto \exp\left(\sum \text{weights of formulas it satisfies}\right)$$

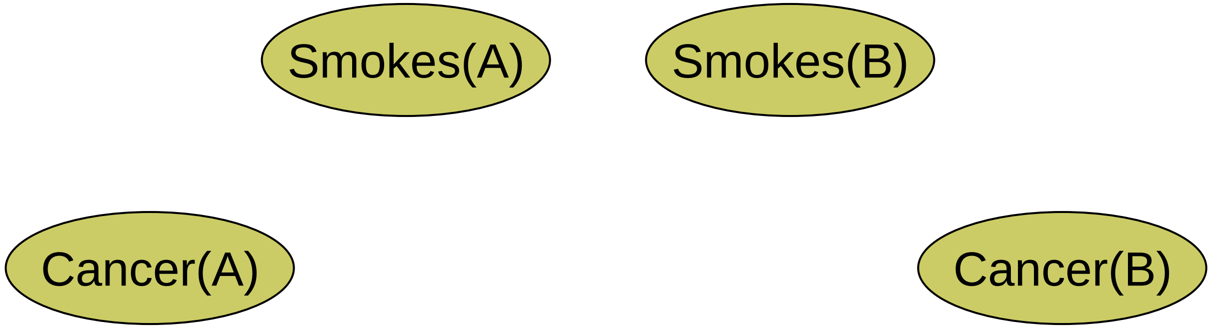
Definition

- 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
- Together with a set of constants, it 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 of an MLN

1.5	$\forall x \; Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y \; Friends(x, y) \Rightarrow \big(Smokes(x) \Leftrightarrow Smokes(y)\big)$

Suppose we have two constants: **Anna** (A) and **Bob** (B)



Smokes(A)

Smokes(B)

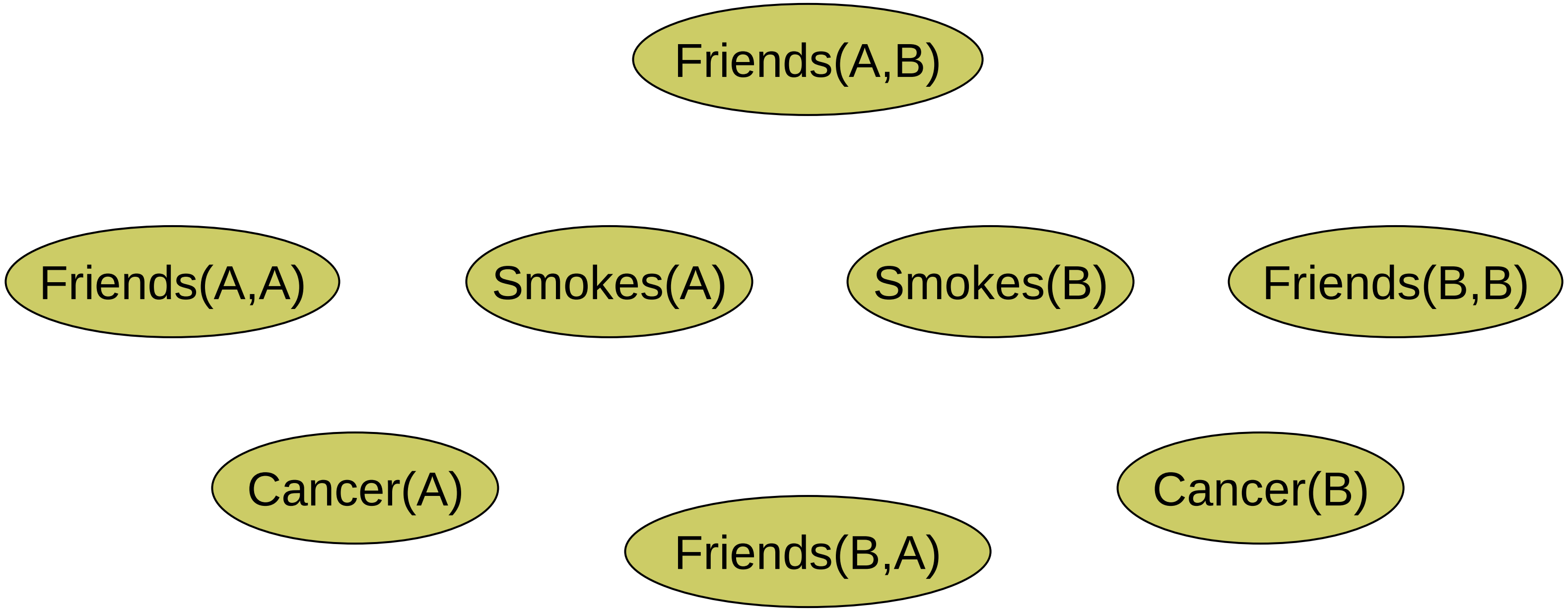
Cancer(A)

Cancer(B)

Example of an MLN

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Suppose we have two constants: **Anna** (A) and **Bob** (B)



Friends(A,B)

Friends(A,A)

Smokes(A)

Smokes(B)

Friends(B,B)

Cancer(A)

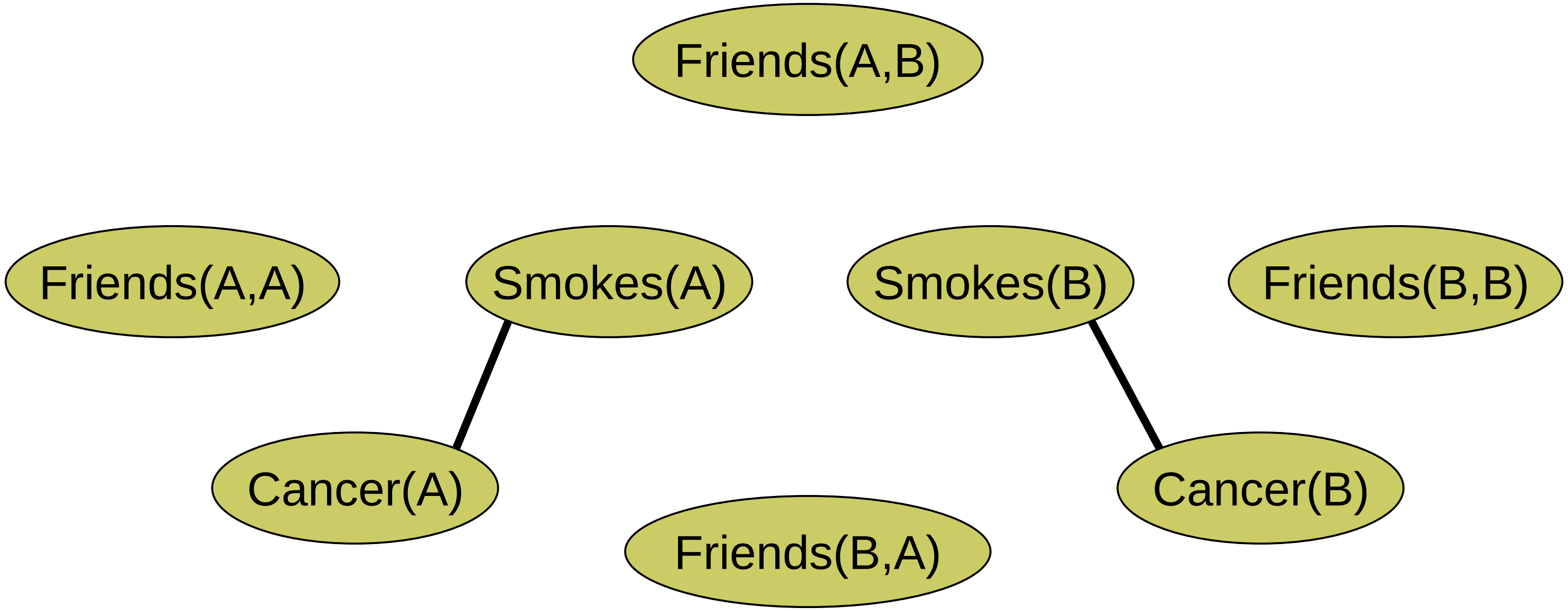
Cancer(B)

Friends(B,A)

Example of an MLN

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Suppose we have two constants: **Anna** (A) and **Bob** (B)



Friends(A,B)

Friends(A,A)

Smokes(A)

Smokes(B)

Friends(B,B)

Cancer(A)

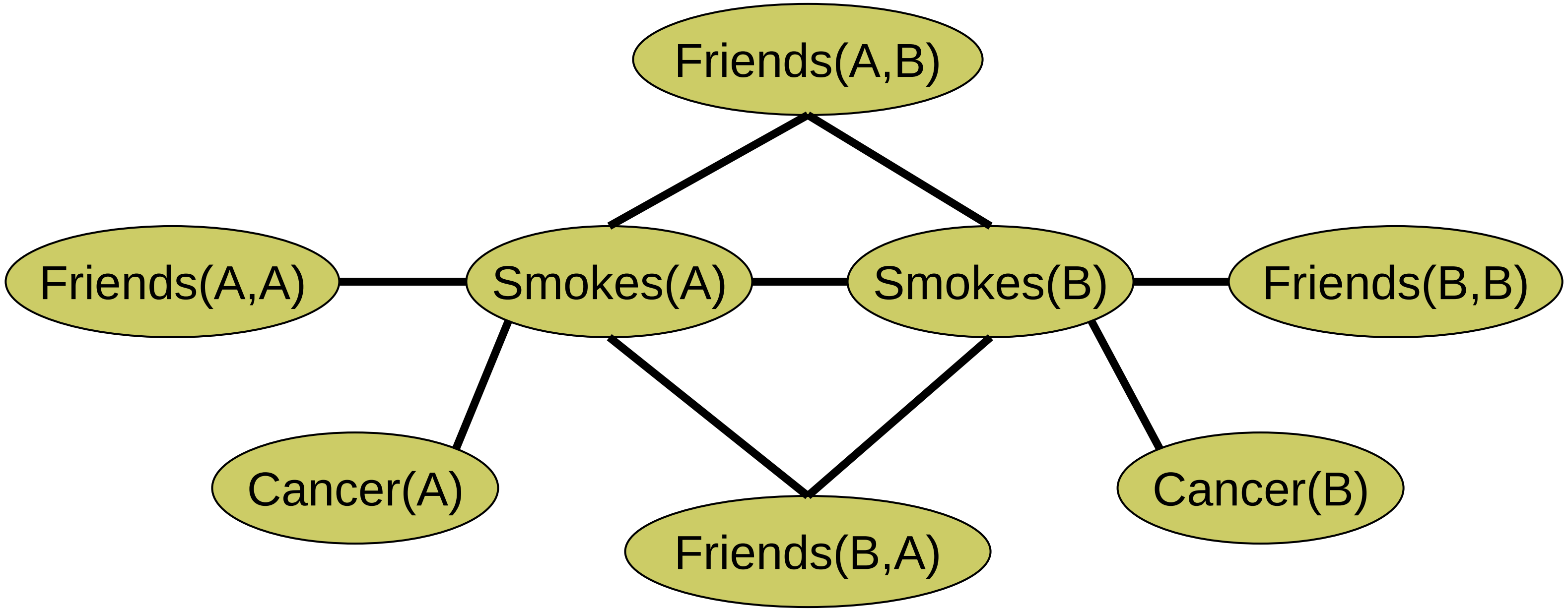
Cancer(B)

Friends(B,A)

Example of an MLN

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Suppose we have two constants: **Anna** (A) and **Bob** (B)



More on MLNs

- **Graph structure:** Arc between two nodes iff predicates appear together in some formula
- MLN is **template** for ground Markov nets
- **Typed** variables and constants greatly reduce size of ground Markov net
- Functions, existential quantifiers, etc.
- MLN without variables = Markov network (subsumes graphical models)

MLNs Subsume FOL

- Infinite weights ⇒ First-order logic
- Satisfiable KB, positive weights ⇒ Satisfying assignments = Modes of distribution
- MLNs allow contradictions between formulas
- How to break KB into formulas?
  - Adding probability increases degrees of freedom
  - Knowledge engineering decision
  - Default: Convert to clausal form

Overview

- Motivation
- Background
- Markov logic networks
- Inference in MLNs
- Learning MLNs
- Experiments
- Discussion

Inference

- Given query predicate(s) and evidence
  1. Extract minimal subset of ground Markov network required to answer query
  2. Apply probabilistic inference to this network

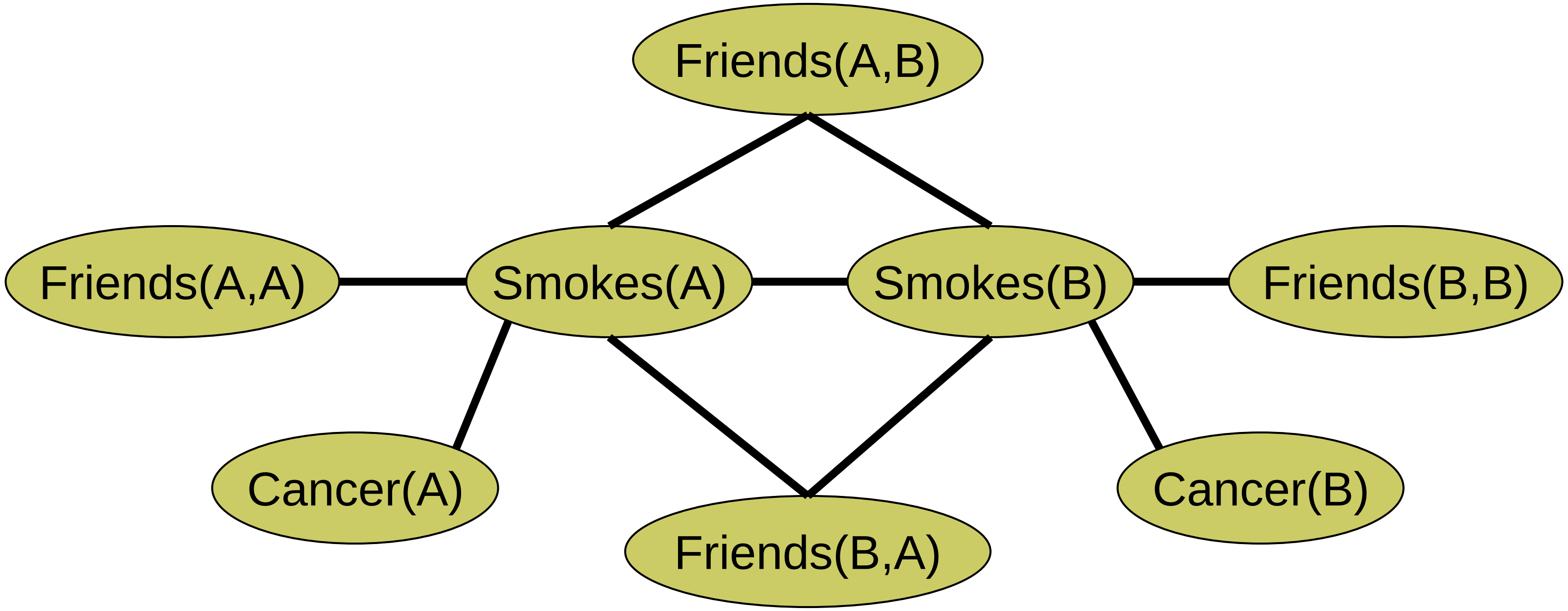
  (Generalization of KBMC [Wellman et al., 1992])

Grounding the Template

- Initialize Markov net to contain all query preds
- For each node in network
  - Add node’s Markov blanket to network
  - Remove any evidence nodes
- Repeat until done

Example Grounding

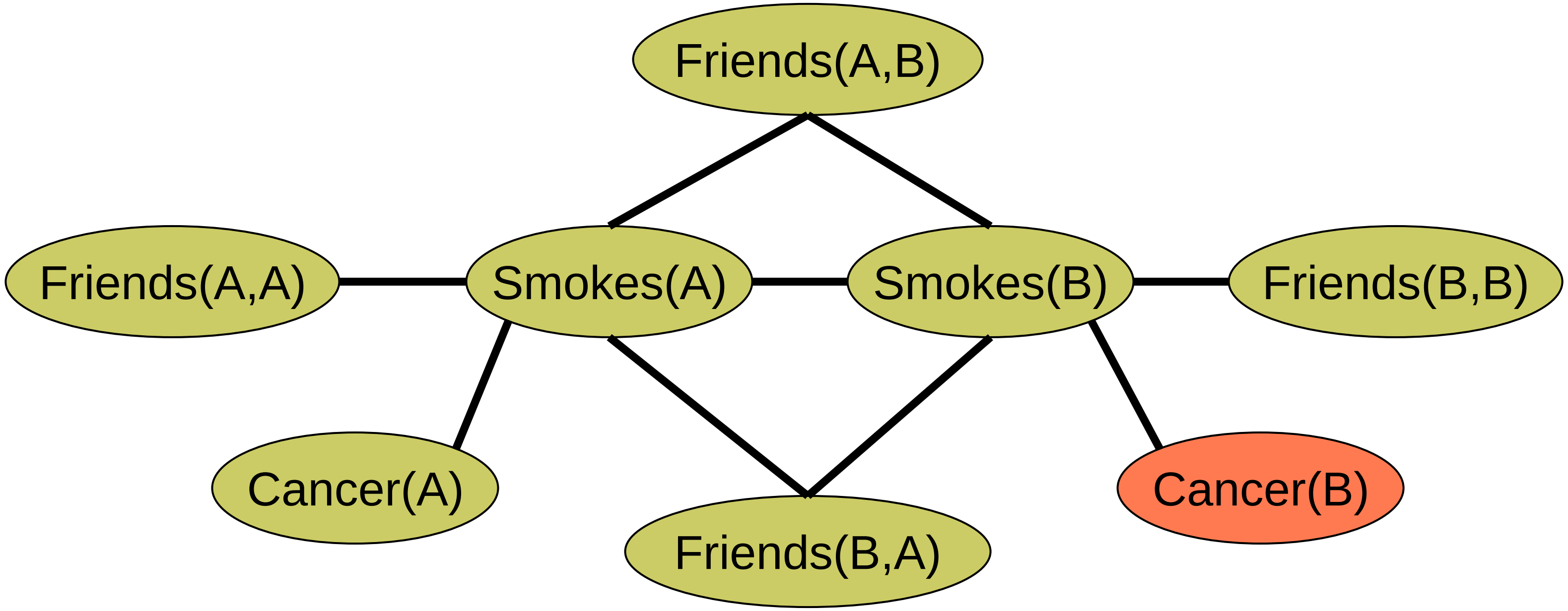
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$



P(Cancer(B) | Smokes(A), Friends(A,B), Friends(B,A))

Example Grounding

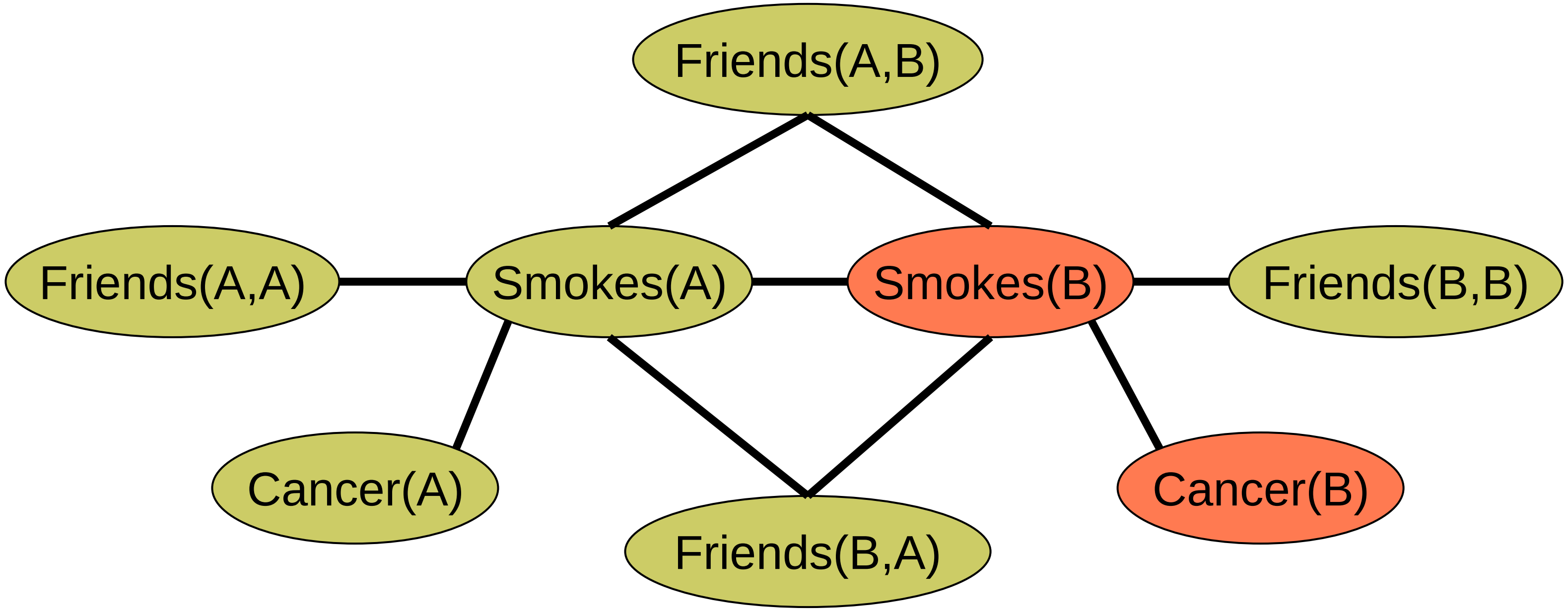
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$



P(Cancer(B) | Smokes(A), Friends(A,B), Friends(B,A))

Example Grounding

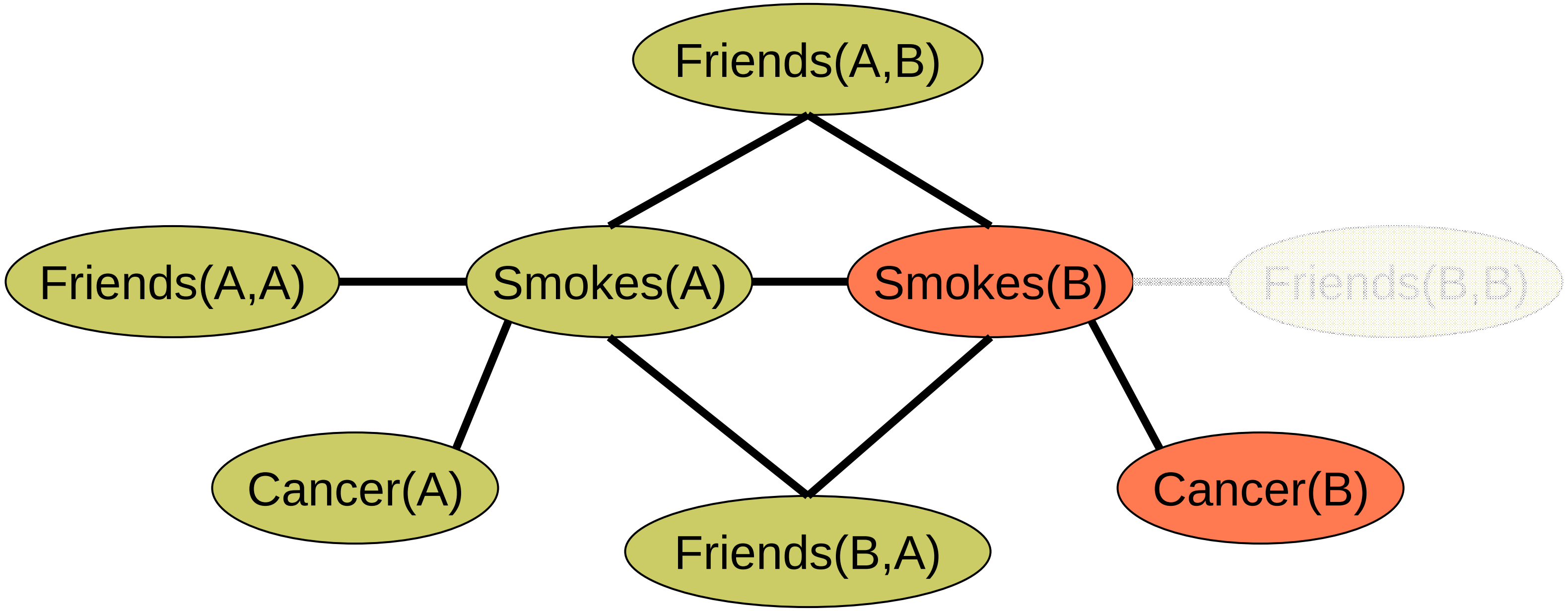
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$



P(Cancer(B) | Smokes(A), Friends(A,B), Friends(B,A))

Example Grounding

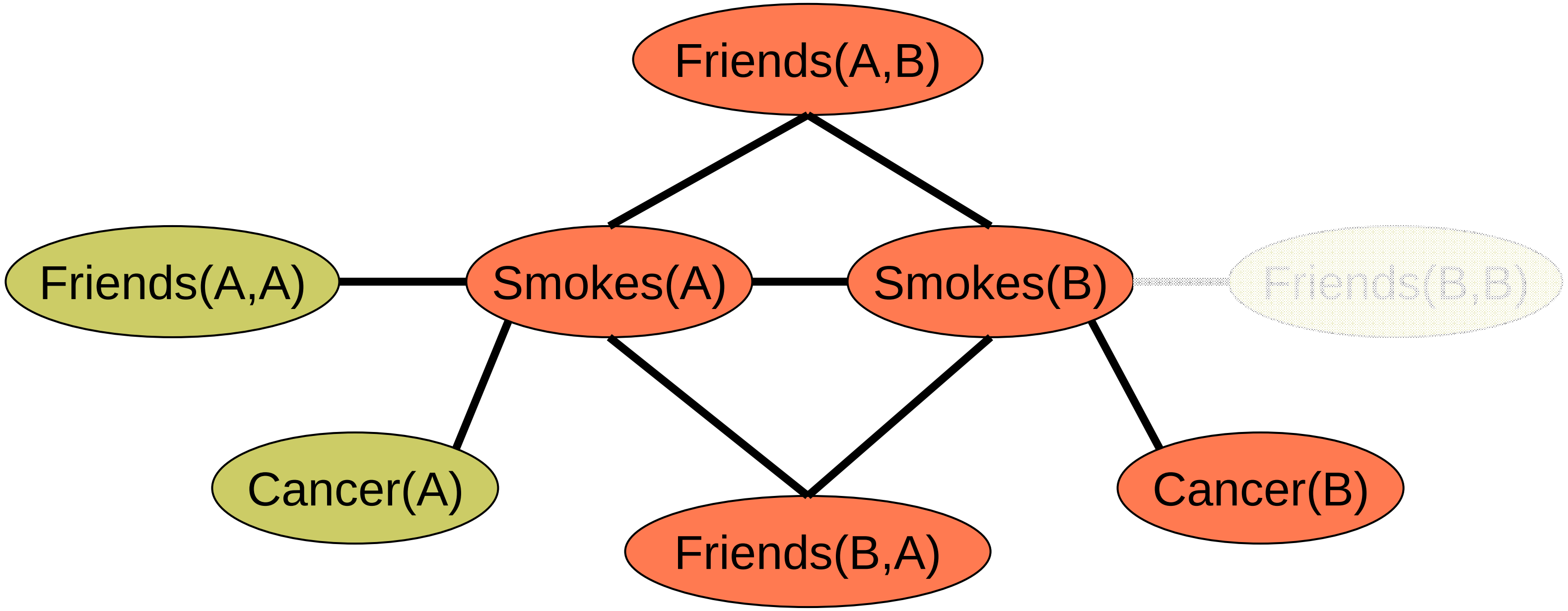
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$



P(Cancer(B) | Smokes(A), Friends(A,B), Friends(B,A))

Example Grounding

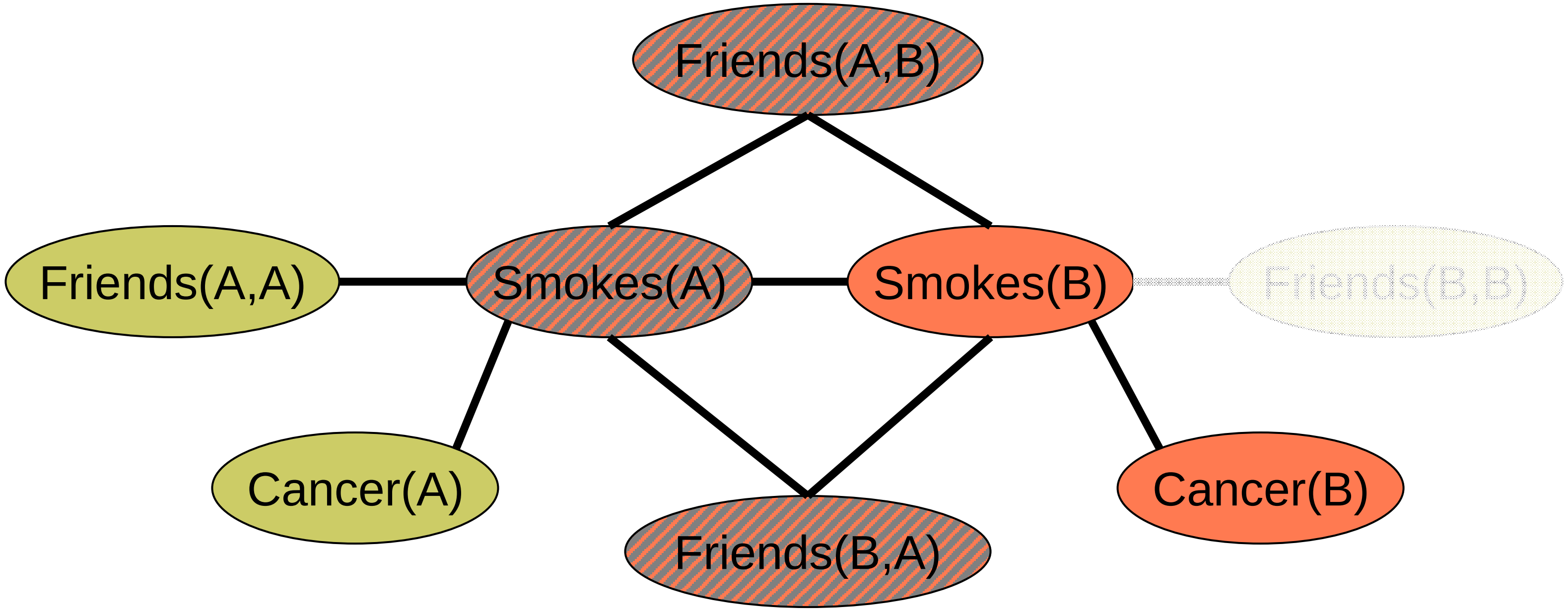
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$



P(Cancer(B) | Smokes(A), Friends(A,B), Friends(B,A))

Example Grounding

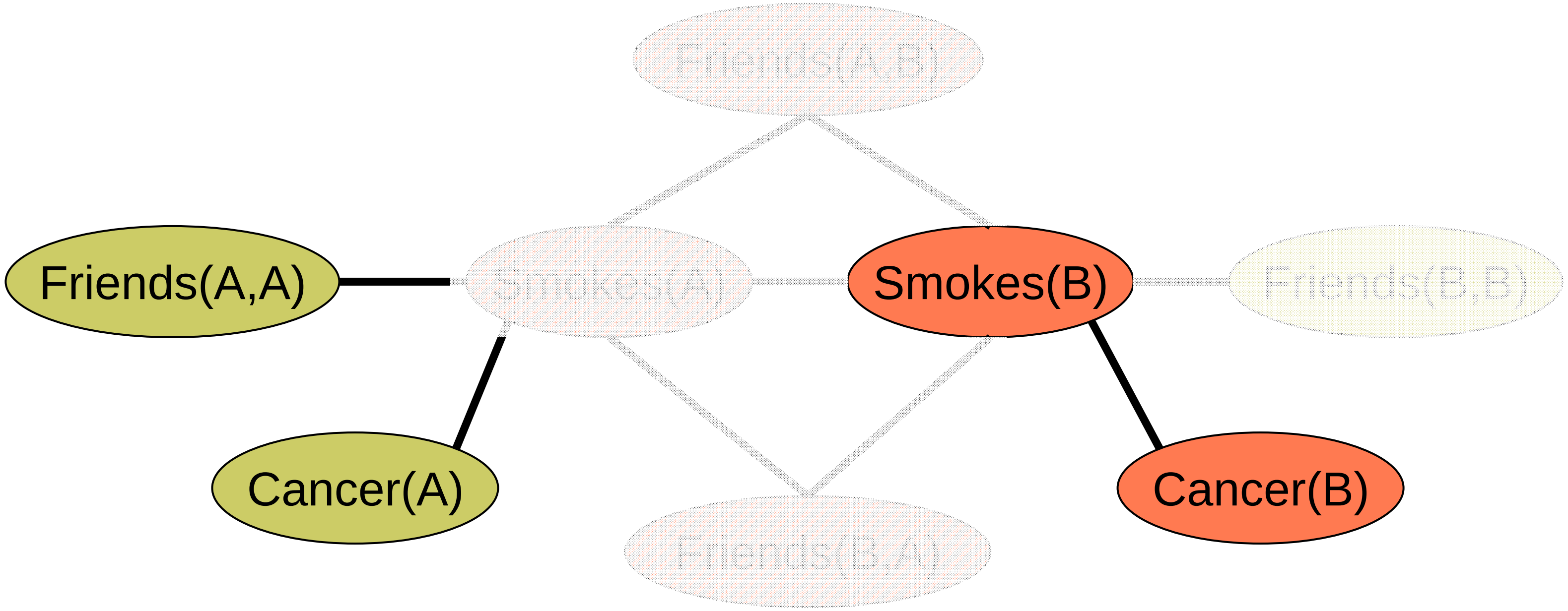
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$



P(Cancer(B) | Smokes(A), Friends(A,B), Friends(B,A))

Example Grounding

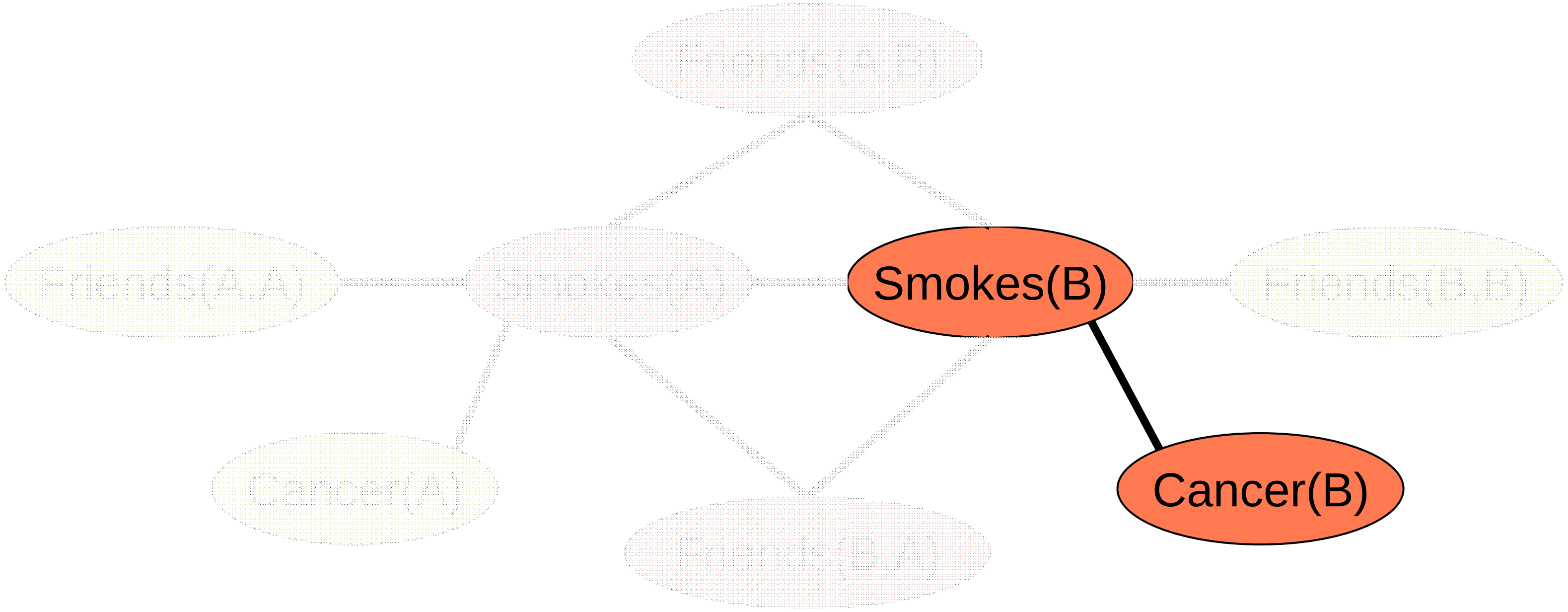
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$



P(Cancer(B) | Smokes(A), Friends(A,B), Friends(B,A))

Example Grounding

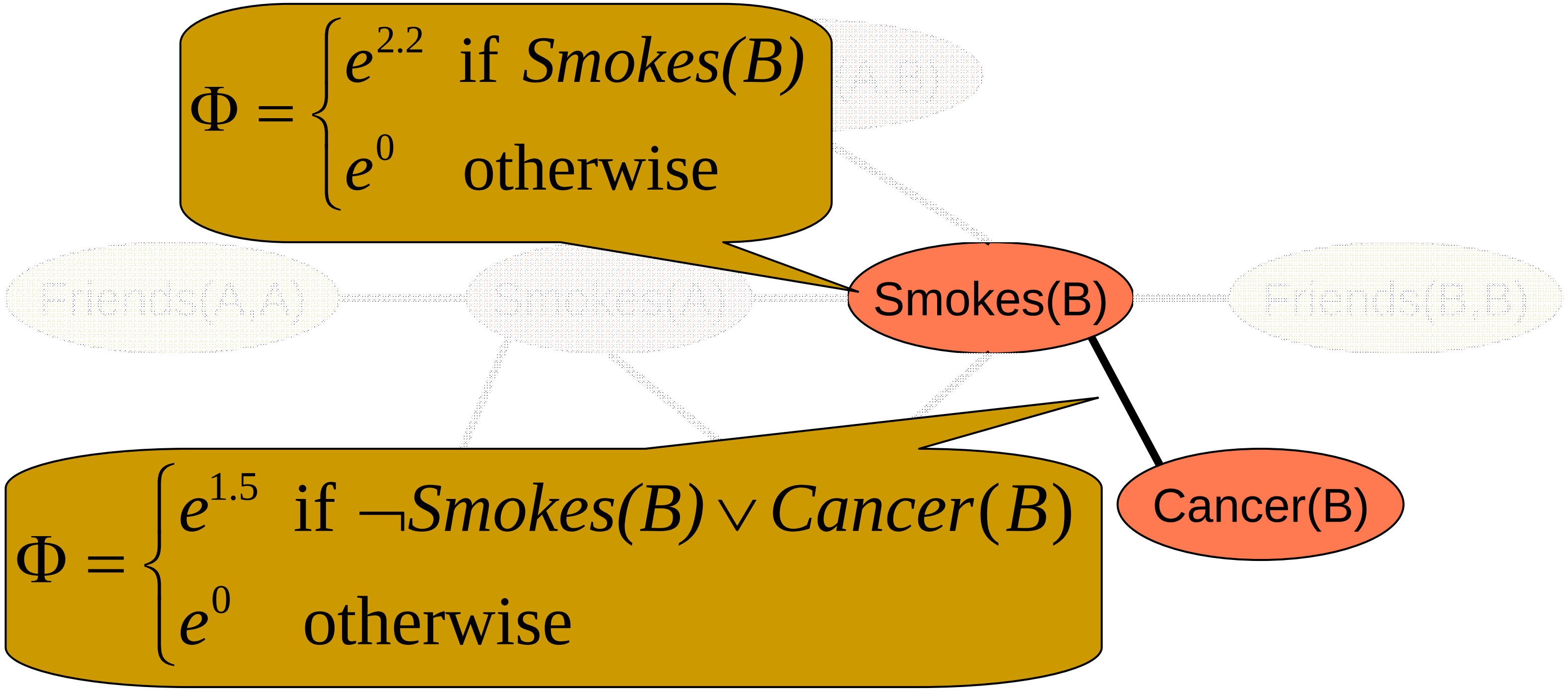
1.5	$\forall x \; Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y \; Friends(x, y) \Rightarrow \big(Smokes(x) \Leftrightarrow Smokes(y)\big)$



P(Cancer(B) | Smokes(A), Friends(A,B), Friends(B,A))

Example Grounding

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$



$$\Phi = \begin{cases} e^{2.2} & \text{if } Smokes(B) \\ e^{0} & \text{otherwise} \end{cases}$$

$$\Phi = \begin{cases} e^{1.5} & \text{if } \neg Smokes(B) \vee Cancer(B) \\ e^{0} & \text{otherwise} \end{cases}$$

P(Cancer(B) | Smokes(A), Friends(A,B), Friends(B,A))

Probabilistic Inference

- Recall

$$P(X) = \frac{1}{Z}\exp\left(\sum_i w_i f_i(X)\right) \qquad Z = \sum_X \exp\left(\sum_i w_i f_i(X)\right)$$

- Exact inference is #P-complete
- Conditioning on Markov blanket is easy:

$$P(x \mid MB(x)) = \frac{\exp\left(\sum_i w_i f_i(x)\right)}{\exp\left(\sum_i w_i f_i(x=0)\right) + \exp\left(\sum_i w_i f_i(x=1)\right)}$$

- Gibbs sampling exploits this

Markov Chain Monte Carlo

- **Gibbs Sampler**
  1. Start with an initial assignment to nodes
  2. One node at a time, sample node given others
  3. Repeat
  4. Use samples to compute P(X)
- Apply to ground network
- Many modes $\Rightarrow$ Multiple chains
- Initialization: MaxWalkSat [Selman et al., 1996]

Overview

- Motivation
- Background
- Markov logic networks
- Inference in MLNs
- Learning MLNs
- Experiments
- Discussion

Learning

- Data is a relational database
- Closed world assumption
- Learning structure
  - Corresponds to feature induction in Markov nets
  - Learn / modify clauses
  - Inductive logic programming (e.g., CLAUDIEN [De Raedt & Dehaspe, 1997])
- Learning parameters (weights)

Learning Weights

- Maximize likelihood (or posterior)
- Use gradient ascent

$$\frac{d}{dw_i}\log P(X) = \underbrace{\boxed{f_i(X)}}_{\text{Feature count according to data}} - \underbrace{\boxed{E_Y[f_i(Y)]}}_{\text{Feature count according to model}}$$

- Requires inference at each step (slow!)

Pseudo-Likelihood [Besag, 1975]

$$PL(X) \equiv \prod_{x} P(x \mid MB(x))$$

- Likelihood of each variable given its Markov blanket in the data
- Does not require inference at each step
- Very fast gradient ascent
- Widely used in spatial statistics, social networks, natural language processing

MLN Weight Learning

- Parameter tying over groundings of same clause
- Maximize pseudo-likelihood using conjugate gradient with line minimization

$$\boxed{\nabla_i = \sum_x nsat_i(x) - \left[p(x{=}0)\,nsat_i(x{=}0) + p(x{=}1)\,nsat_i(x{=}1) \right]}$$

where *$nsat_i(x{=}v)$* is the number of satisfied groundings of clause i **in the training data** when x takes value v

- Most terms not affected by changes in weights
- After initial setup, each iteration takes O(# ground predicates x # first-order clauses)

Overview

- Motivation
- Background
- Markov logic networks
- Inference in MLNs
- Learning MLNs
- Experiments
- Discussion

Domain

- University of Washington CSE Dept.
- 24 first-order predicates:
  Professor, Student, TaughtBy, AuthorOf, AdvisedBy, etc.
- 2707 constants divided into 11 types:
  Person (400), Course (157), Paper (76), Quarter (14), etc.
- 8.2 million ground predicates
- 9834 ground predicates (tuples in database)

Systems Compared

- Hand-built knowledge base (KB)
- ILP: CLAUDIEN [De Raedt & Dehaspe, 1997]
- Markov logic networks (MLNs)
  - Using KB
  - Using CLAUDIEN
  - Using KB + CLAUDIEN
- Bayesian network learner [Heckerman et al., 1995]
- Naïve Bayes [Domingos & Pazzani, 1997]

Sample Clauses in KB

- Students are not professors
- Each student has only one advisor
- If a student is an author of a paper, so is her advisor
- Advanced students only TA courses taught by their advisors
- At most one author of a given paper is a professor

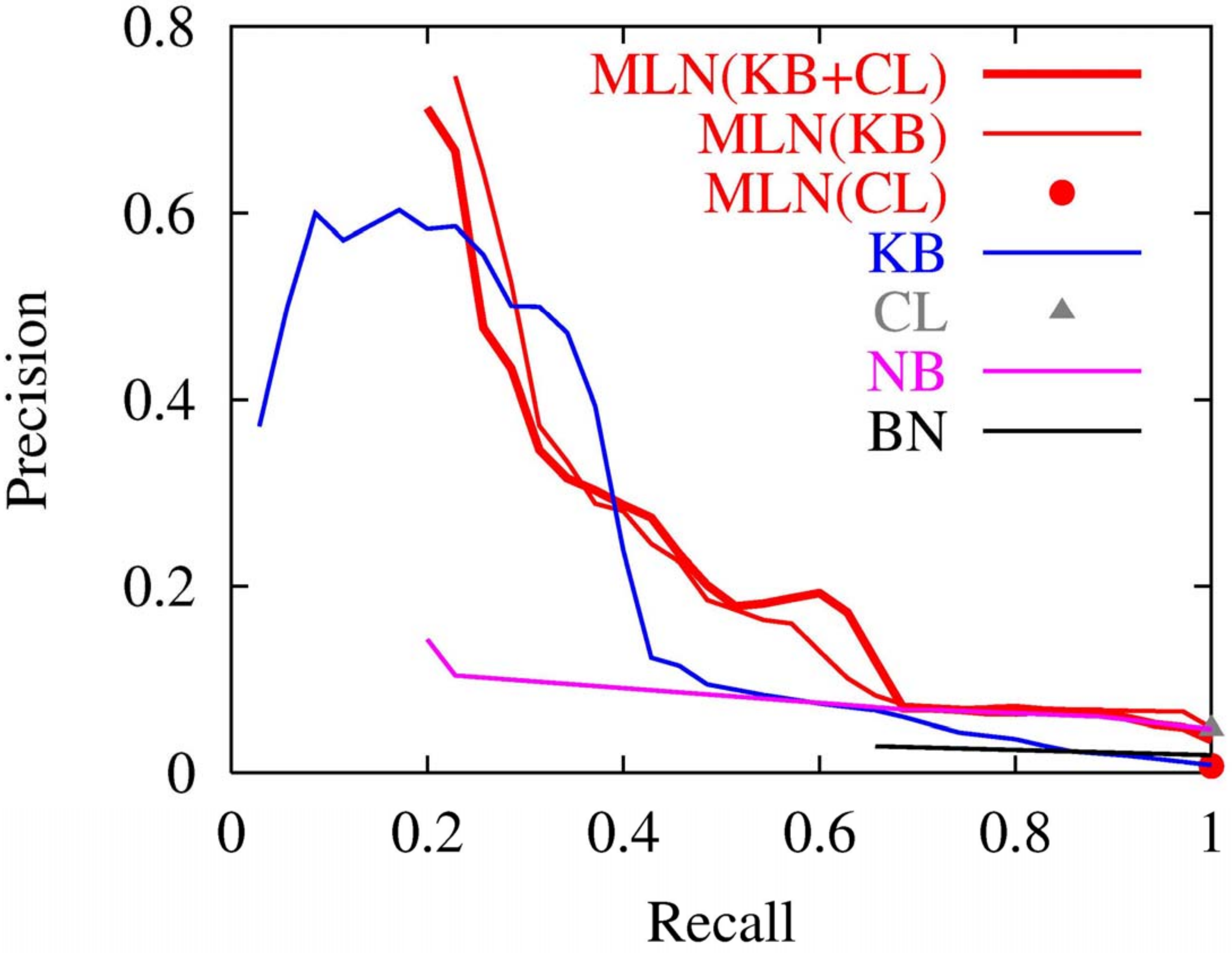
Methodology

- Data split into five areas:
  AI, graphics, languages, systems, theory
- Leave-one-area-out testing
- Task: Predict AdvisedBy(x, y)
  - All Info: Given all other predicates
  - Partial Info: With Student(x) and Professor(x) missing
- Evaluation measures:
  - Conditional log-likelihood
    (KB, CLAUDIEN: Run WalkSat 100x to get probabilities)
  - Area under precision-recall curve

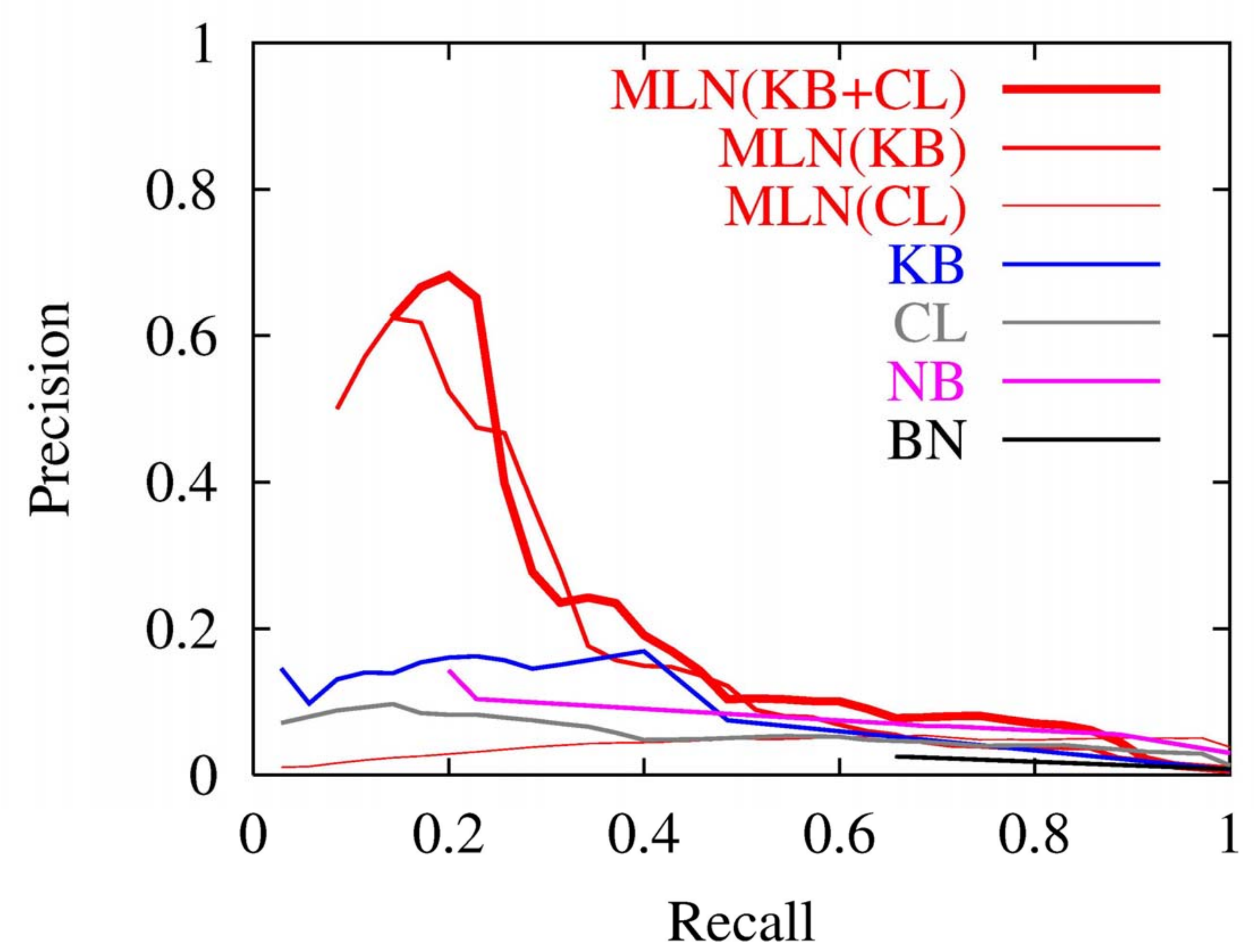
Results

System	All Info		Partial Info	
	CLL	Area	CLL	Area
MLN(KB+CL)	-0.044	0.27	-0.040	0.25
MLN(KB)	-0.047	0.25	-0.043	0.20
MLN(CL)	-0.341	0.02	-0.852	0.03
KB	-0.124	0.16	-0.067	0.08
CL	-0.255	0.03	-0.371	0.04
NB	-0.218	0.17	-0.059	0.19
BN	-0.095	0.02	-0.166	0.04

Results: All Info



Results: Partial Info



Efficiency

- Learning time: 88 mins
- Time to infer all 4900 AdvisedBy predicates:
  - With complete info: 23 mins
  - With partial info: 24 mins

(10,000 samples)

Overview

- Motivation
- Background
- Markov logic networks
- Inference in MLNs
- Learning MLNs
- Experiments
- Discussion

Related Work

- Knowledge-based model construction
  [Wellman et al., 1992; etc.]
- Stochastic logic programs
  [Muggleton, 1996; Cussens, 1999; etc.]
- Probabilistic relational models
  [Friedman et al., 1999; etc.]
- Relational Markov networks
  [Taskar et al., 2002]
- Etc.

Special Cases of Markov Logic

- Collective classification
- Link prediction
- Link-based clustering
- Social network models
- Object identification
- Etc.

Future Work: Inference

- Lifted inference
- Better MCMC (e.g., Swendsen-Wang)
- Belief propagation
- Selective grounding
- Abstraction, summarization, multi-scale
- Special cases
- Etc.

Future Work: Learning

- Faster optimization
- Beyond pseudo-likelihood
- Discriminative training
- Learning and refining structure
- Learning with missing info
- Learning by reformulation
- Etc.

Future Work: Applications

- Object identification
- Information extraction & integration
- Natural language processing
- Scene analysis
- Systems biology
- Social networks
- Assisted cognition
- Semantic Web
- Etc.

Conclusion

- Computer systems must learn, reason logically, and handle uncertainty
- **Markov logic networks** combine full power of first-order logic and prob. graphical models
  - **Syntax:** First-order logic + Weights
  - **Semantics:** Templates for Markov networks
- **Inference:** MCMC over minimal grounding
- **Learning:** Pseudo-likelihood and ILP
- Experiments on UW DB show promise