Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 30

Nov, 23, 2016

Slide source: from Pedro Domingos UW & Markov Logic: An Interface Layer for Artificial Intelligence Pedro Domingos and Daniel Lowd University of Washington, Seattle

Lecture Overview

- Recap Markov Logic (Networks)
- Relation to First-Order Logics
- Inference in MLN
  - MAP Inference (most likely *pw*)
  - Probability of a formula, Conditional Probability

Prob. Rel. Models vs. Markov Logic

PRM

- Relational Skeleton
- Dependency Graph
- Parameters (CPT)

$\Rightarrow$ BNET

ML

- weighted logical formulas
- set of constants

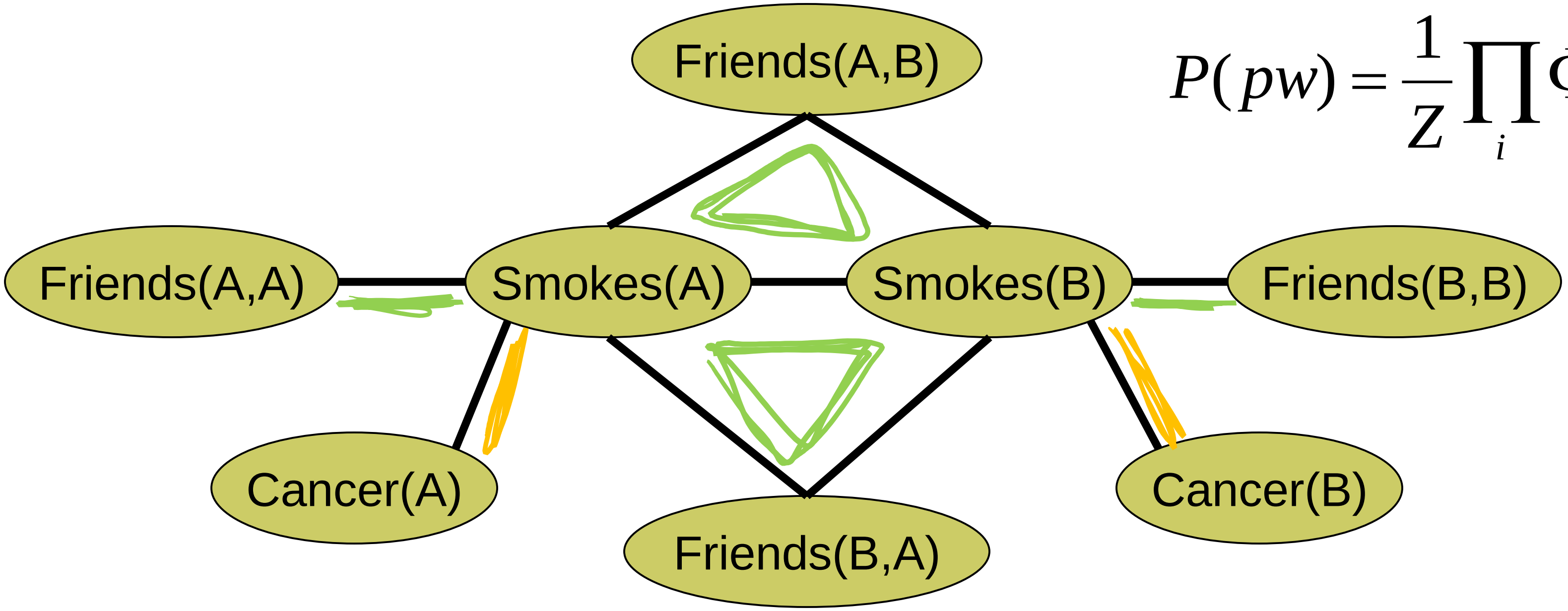
$\Rightarrow$ MARKOV LOGIC NETWORK

MLN features

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

Two constants: **Anna** (A) and **Bob** (B)

$$P(pw) = \frac{1}{Z}\prod_i \Phi_i(pw)$$



MLN: parameters

- For each grounded formula i we have a factor

$\Phi_i(pw) = e^{w_i f_i(pw)}$ ← possible world

- Same for all the groundings of the same formula

w_i weight of formula

$$f_i(pw) = \begin{cases} 1 & \text{when formula is true in } pw \\ 0 & \text{otherwise} \end{cases}$$

$1.5 \quad \forall x\ Smokes(x) \Rightarrow Cancer(x)$

$$f(\text{Smokes(x)}, \text{Cancer(x)}) = \begin{cases} 1 & \text{if Smokes(x)} \Rightarrow \text{Cancer(x)} \\ 0 & \text{otherwise} \end{cases}$$

pw_1 ....

Smokes(A) T

Cancer(A) F $\quad e^0 = 1$

pw_2 ....

Smokes(A) T

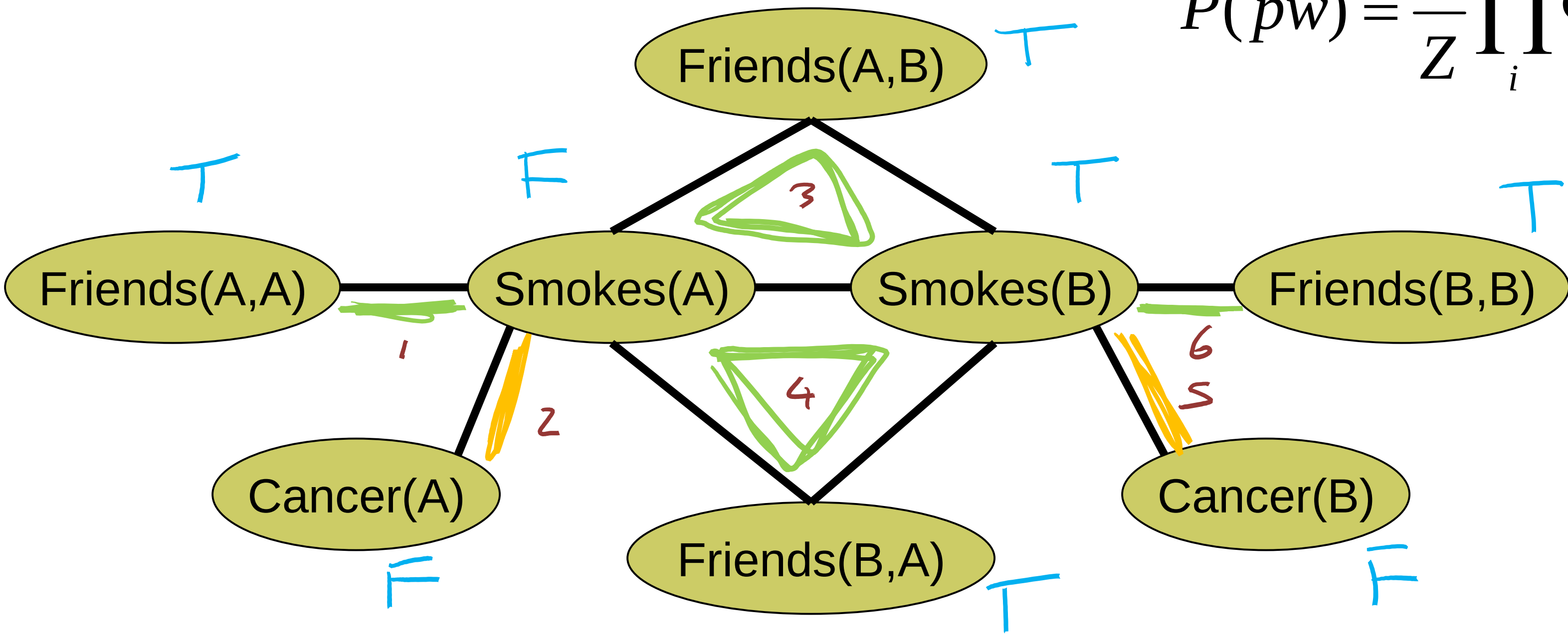
Cancer(A) T $\quad e^{1.5}$

MLN: prob. of possible world

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

Two constants: **Anna** (A) and **Bob** (B)

$$P(pw) = \frac{1}{Z} \prod_i \Phi_i(pw)$$



$$P(pw) = \Big(\underset{1}{e^{1.1}} * \underset{6}{e^{1.1}} * \underset{3}{e^{0}} * \underset{4}{e^{0}} * \underset{2}{e^{1.5}} * \underset{5}{e^{0}}\Big) / Z$$

MLN: prob. Of possible world

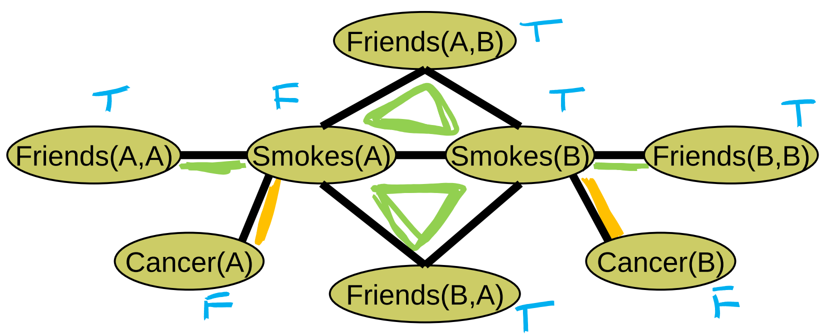
1	1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
2	1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

- Probability of a world pw:

$$P(pw) = \frac{1}{Z} \exp\left(\sum_i \boxed{w_i} \boxed{n_i(pw)} \right)$$

Weight of formula i

No. of true groundings of formula i in pw



$$P(pw) = \left(\underbrace{e^{1.1} * e^{1.1}}_{n_2(pw) = 2} * e^{0} * e^{0} * \underbrace{e^{1.5}}_{n_1(pw) = 1} * e^{0} \right) / Z$$

Lecture Overview

- Recap Markov Logic (Networks)
- **Relation to First-Order Logics**
- **Inference in MLN**
  - **MAP Inference (most likely *pw*)**
  - **Probability of a formula, Conditional Probability**

How MLN s generalize FOL

- Consider MLN containing only one formula

$$w \quad \forall x\, R(x) \Rightarrow S(x) \qquad C = \{A\}$$

$\Phi_1(pw) = e^{w f_1(pw)}$

Φ_1

$Z = 1 + 3e^w$

4 pws

R(A)	S(A)	$f_1(pw)$	$\Phi_1(pw)$	P(pw)
T	T	1	e^w	$e^w / 1+3e^w$
F	T	1	e^w	$e^w / 1+3e^w$
T	F	0	1	$1 / 1+3e^w$
F	F	1	e^w	$e^w / 1+3e^w$

$$P(\underset{=T}{S(A)} \mid \underset{=T}{R(A)}) = \frac{P(S(A) \wedge R(A))}{P(R(A))} = \frac{e^w/Z}{1/Z + e^w/Z} = \frac{e^w}{1+e^w} = \boxed{\frac{1}{e^{-w}+1}}$$

$w \to \infty$ $\quad\to 1$

$$w \to \infty, P(S(A) \mid R(A)) \to 1$$

“recovering logical entailment”

How MLN s generalize FOL

First order logic (with some mild assumptions) is a special Markov Logics obtained when

- all the weight are equal
- and tend to infinity

Lecture Overview

- Recap Markov Logic (Networks)
- Relation to First-Order Logics
- **Inference in MLN**
  - **MAP Inference (most likely *pw*)**
  - **Probability of a formula, Conditional Probability**

Inference in MLN

- MLN acts as a template for a Markov Network
- We can always answer prob. queries using standard Markov network inference methods on the instantiated network
- **However**, due to the size and complexity of the resulting network, this is often infeasible.
- Instead, we combine **probabilistic methods** with ideas from **logical inference**, including **satisfiability** and **resolution**.
- This leads to efficient methods that take full advantage of the logical structure.

MAP Inference

- **Problem:** Find most likely state of world

$$\arg\max_{pw} \; P(pw)$$

- Probability of a world *pw*:

$$P(pw) = \frac{1}{Z}\exp\left(\sum_i \boxed{w_i}\,\boxed{n_i(pw)}\right)$$

Weight of formula *i*

No. of true groundings of formula *i* in *pw*

$$\arg\max_{pw} \; \frac{1}{Z}\exp\left(\sum_i w_i n_i(pw)\right)$$

MAP Inference

$$\arg\max_{pw} \frac{1}{Z} \exp\left(\sum_i w_i n_i(pw) \right)$$

$$\arg\max_{pw} \sum_i w_i n_i(pw)$$

- Are these two equivalent?

A. Yes B. No C. It depends . . . .

MAP Inference

- Therefore, the MAP problem in Markov logic reduces to finding the truth assignment that maximizes the sum of weights of satisfied formulas (let's assume clauses)

$$\arg\max_{pw} \sum_i w_i n_i(pw)$$

- This is just the weighted MaxSAT problem
- Use weighted SAT solver
(e.g., MaxWalkSAT [Kautz et al., 1997])

WalkSAT algorithm (in essence) (from lecture 21 – one change)

(Stochastic) Local Search Algorithms can be used for this task!

Evaluation Function: number of satisfied clauses

WalkSat: One of the simplest and most effective algorithms:

Start from a randomly generated interpretation (*pw*)

- Pick randomly an unsatisfied clause
- Pick a proposition/atom to flip (randomly ~~1 or 2~~)
  1. Randomly
  2. To maximize # of satisfied clauses

if all clauses satisfied DONE ☺
else

MaxWalkSAT algorithm (in essence)

Evaluation Function *f(pw)* : Σ weights(sat. clauses in pw)

current pw <- randomly generated interpretation

Generate *new pw* by doing the following

- Pick randomly an unsatisfied clause
- Pick a proposition/atom to flip (randomly 1 or 2)
  1. Randomly
  2. To maximize Σ weights(sat. clauses in resulting *pw*)

if f(new pw) > f(current pw)
current pw ← new pw

Computing Probabilities

P(Formula|$M_{L,C}$) = ?

- **Brute force:** Sum probs. of possible worlds where formula holds

$M_{L,C}$ Markov Logic Network

PW_F possible worlds in which F is true

$$P(F, M_{L,C}) = \sum_{pw \in PW_F} P(pw, M_{L,C})$$

- **MCMC:** Sample worlds, check formula holds

S all samples

S_F samples (i.e. possible worlds) in which F is true

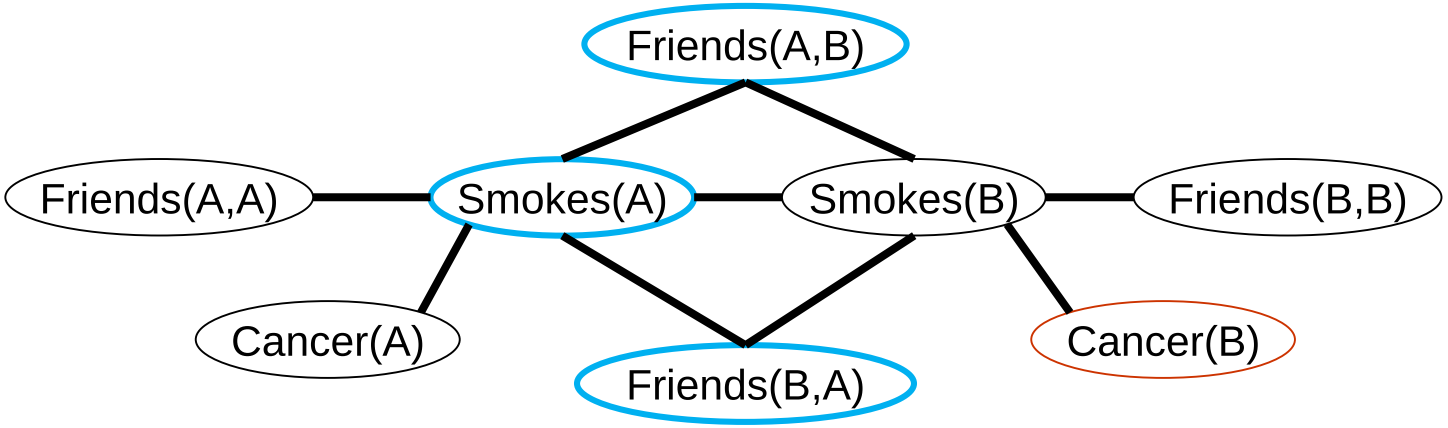
$$P(F, M_{L,C}) = \frac{|S_F|}{|S|}$$

Computing Cond. Probabilities

Let's look at the simplest case

P(ground literal | conjuction of ground literals, $M_{L,C}$)

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



To answer this query do you need to create (ground) the whole network?

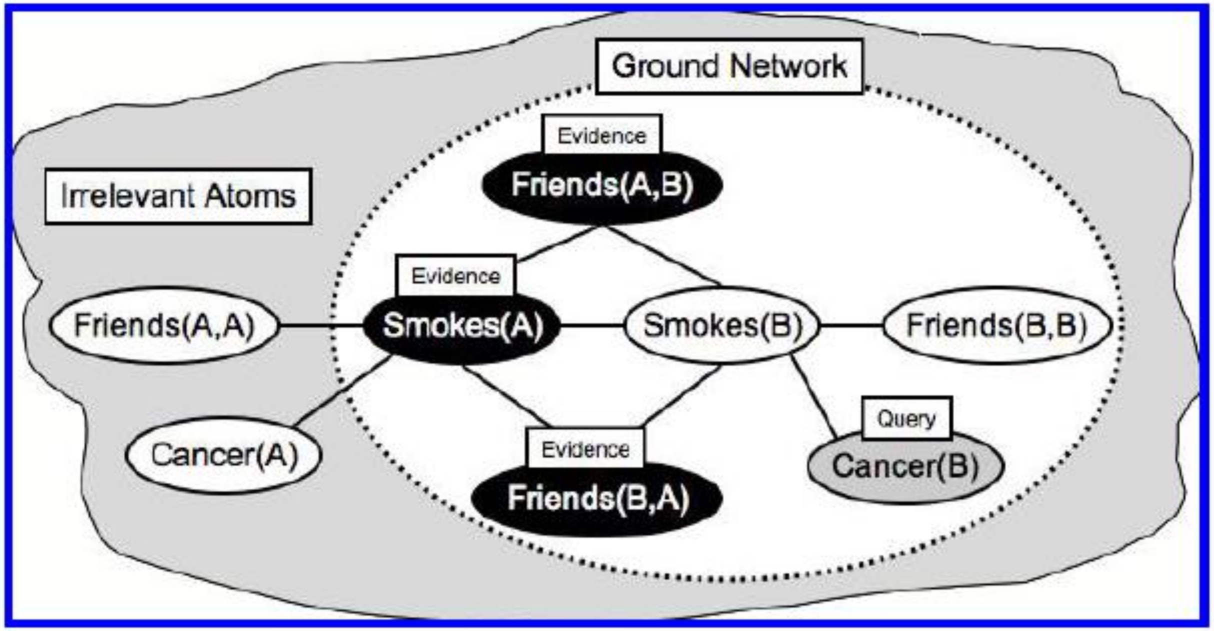
A. Yes B. No C. It depends . . .

Computing Cond. Probabilities

Let's look at the simplest case

P(ground literal | conjuction of ground literals, $M_{L,C}$)

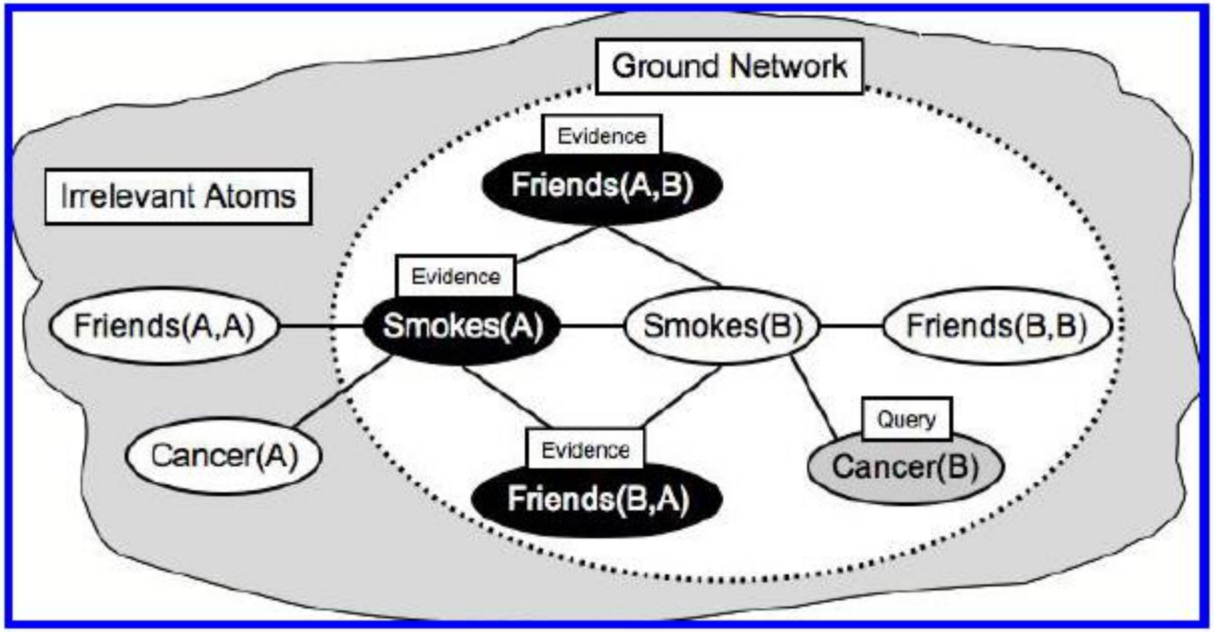
P(Cancer(B)| Smokes(A), Friends(A, B), Friends(B, A))



You do not need to create (ground) the part of the Markov Network from which the query is independent given the evidence

Computing Cond. Probabilities

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



Then you can perform Gibbs Sampling in this Sub Network

Learning Goals for today's class

You can:

- Show on an example how MLNs generalize FOL
- Compute the most likely *pw* (given some evidence)
- Probability of a formula, Conditional Probability

Next class on Fri

- Markov Logic: applications
- Start. Prob Relational Models

Start working on **Assignment-4**

Due **Dec 2**

Inference in MLN

- MLN acts as a template for a Markov Network
- We can always answer prob. queries using standard Markov network inference methods on the instantiated network
- **However**, due to the size and complexity of the resulting network, this is often infeasible.
- Instead, we combine **probabilistic methods** with ideas from **logical inference**, including **satisfiability** and **resolution**.
- This leads to efficient methods that take full advantage of the logical structure.

MAP Inference

- Find most likely state of world

$$\arg\max_{pw} \; P(pw)$$

- Reduces to finding the *pw* that maximizes the sum of weights of satisfied clauses

$$\arg\max_{pw} \; \sum_i w_i n_i(pw)$$

- Use weighted SAT solver (e.g., MaxWalkSAT [Kautz et al., 1997])

Probabilistic problem solved by logical inference method

Computing Probabilities

P(Formula,$M_{L,C}$) = ?

- **Brute force:** Sum probs. of possible worlds where formula holds

$M_{L,C}$ Markov Logic Network

PW_F possible worlds in which F is true

$$P(F, M_{L,C}) = \sum_{pw \in PW_F} P(pw, M_{L,C})$$

- **MCMC:** Sample worlds, check formula holds

S all samples

S_F samples (i.e. possible worlds) in which F is true

$$P(F, M_{L,C}) = \frac{|S_F|}{|S|}$$

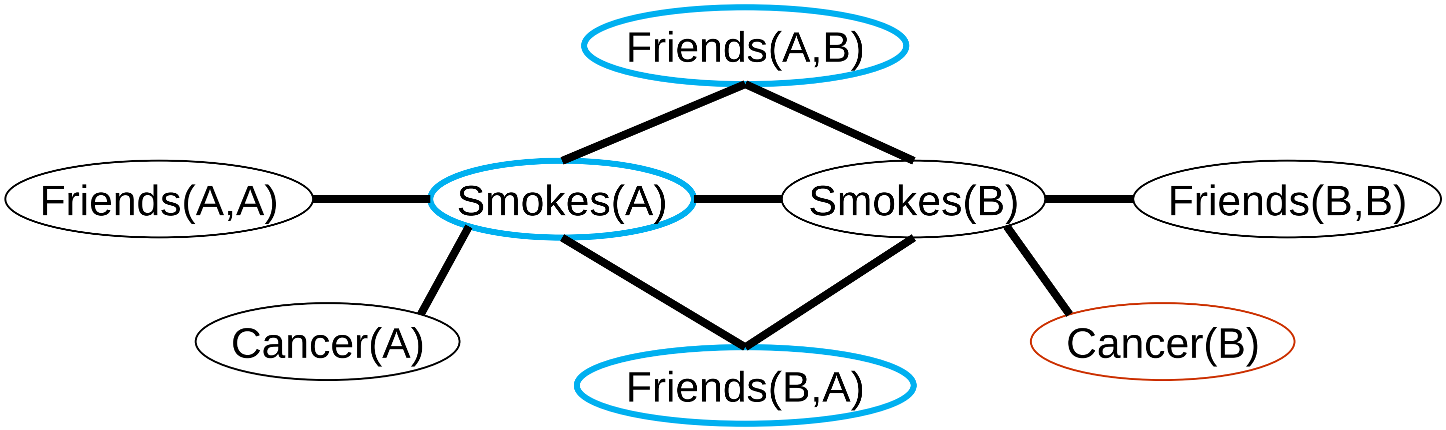
Computing Cond. Probabilities

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

Let's look at the simplest case

P(ground literal | conjuction of ground literals, $M_{L,C}$)

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



iclicker.

To answer this query do you need to create (ground) the whole network?

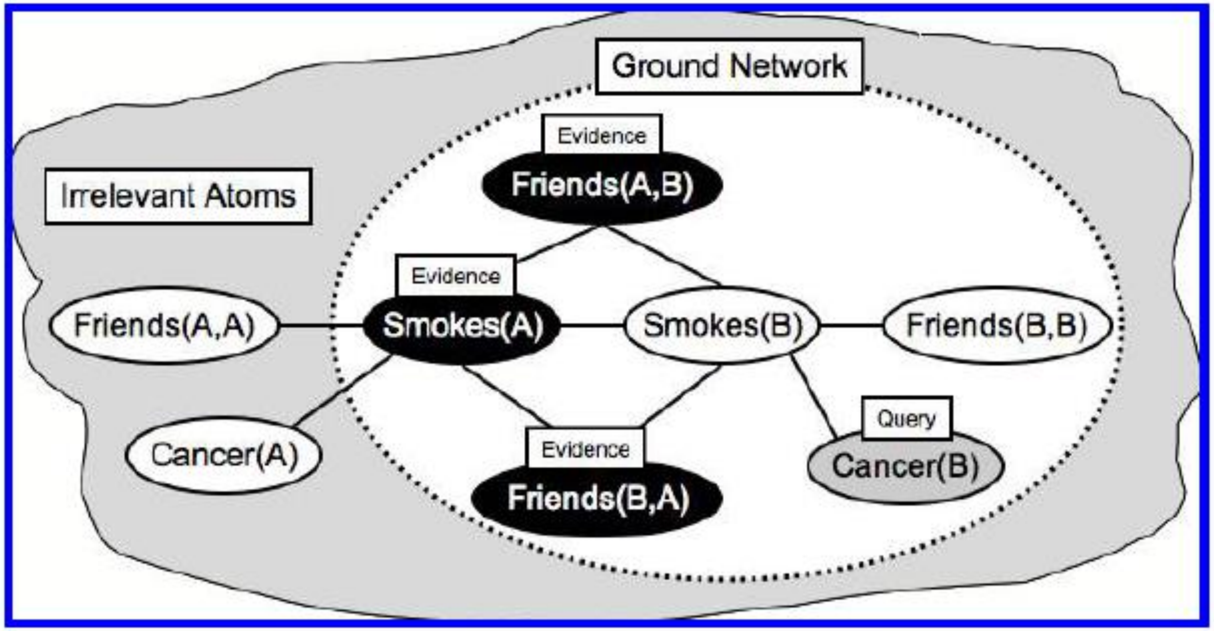
A. Yes B. No C. It depends . . .

Computing Cond. Probabilities

Let's look at the simplest case

P(ground literal | conjuction of ground literals, $M_{L,C}$)

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



You do not need to create (ground) the part of the Markov Network from which the query is independent given the evidence

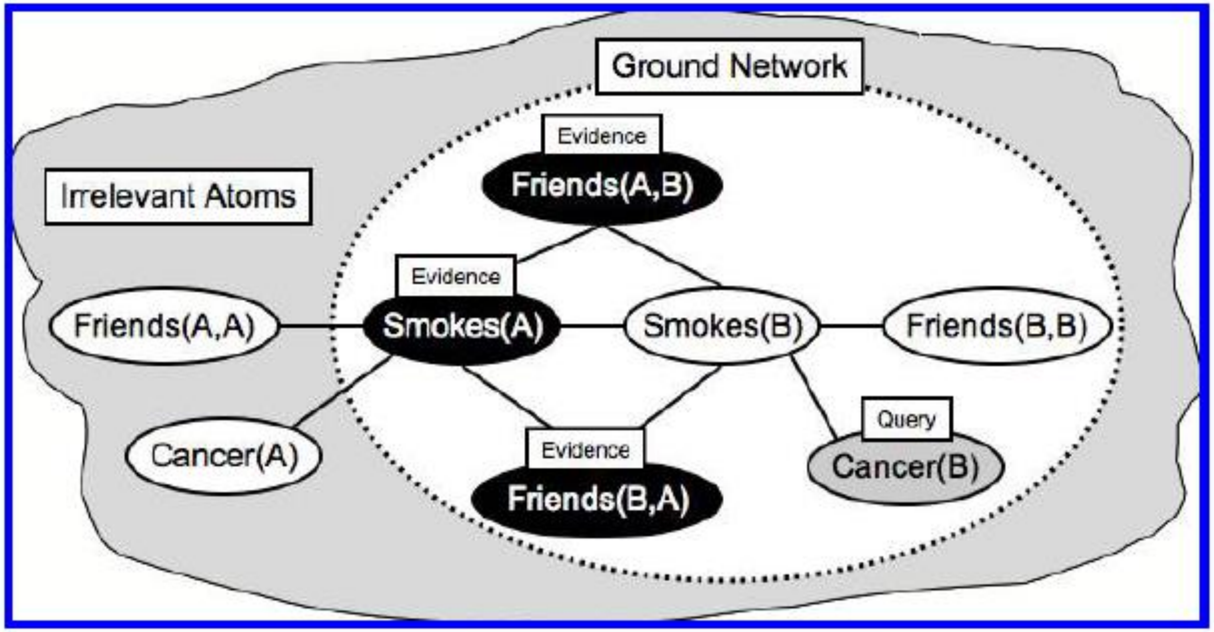
Computing Cond. Probabilities

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

The sub network is determined by the formulas

(the logical structure of the problem)

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



You can then perform Gibbs Sampling in this Sub Network

Lecture Overview

- Finish Inference in MLN
  - Probability of a formula, Conditional Probability
- Markov Logic: applications
- Beyond 322/422 (ML + grad courses)
- AI conf. and journals
- Watson….
- Final Exam (office hours, samples)
- TA evaluation

Entity Resolution

- Determining which observations correspond to the same real-world objects

- (e.g., database records, noun phrases, video regions, etc)

- Crucial importance in many areas

(e.g., data cleaning, NLP, Vision)

Entity Resolution: Example

AUTHOR: *H. POON & P. DOMINGOS*
TITLE: *UNSUPERVISED SEMANTIC PARSING*
VENUE: *EMNLP-09*

SAME?

AUTHOR: *Hoifung Poon and Pedro Domings*
TITLE: *Unsupervised semantic parsing*
VENUE: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*

AUTHOR: *Poon, Hoifung and Domings, Pedro*
TITLE: *Unsupervised ontology induction from text*
VENUE: *Proceedings of the Forty-Eighth Annual Meeting of the Association for Computational Linguistics*

SAME?

AUTHOR: *H. Poon, P. Domings*
TITLE: *Unsupervised ontology induction*
VENUE: *ACL-10*

Entity Resolution (relations)

Problem: **Given citation database, find duplicate records**
Each citation has `author`, `title`, and `venue` fields
We have 10 relations

provided as evidence

`Author(bib,author)`
`Title(bib,title)`
`Venue(bib,venue)`

`relate citations to their fields`

`HasWord(author, word)`
`HasWord(title, word)`
`HasWord(venue, word)`

`indicate which words are present in each field;`

`SameAuthor (author, author)` `represent field equality;`
`SameTitle(title, title)`
`SameVenue(venue, venue)`

`SameBib(bib, bib)` `represents citation equality;`

To be inferred

Entity Resolution (formulas)

Predict citation equality based on words in the fields

```
Title(b1, t1) ∧ Title(b2, t2) ∧
HasWord(t1,+word) ∧ HasWord(t2,+word) ⇒
SameBib(b1, b2)
```

1000s of rules one for each word

```
(NOTE: +word is a shortcut notation, you
actually have a rule for each word e.g.,
Title(b1, t1) ∧ Title(b2, t2) ∧
HasWord(t1,"bayesian") ∧
HasWord(t2,"bayesian" ) ⇒ SameBib(b1, b2) )
```

Same 1000s of rules for author

Same 1000s of rules for venue

Entity Resolution (formulas)

Transitive closure

SameBib(b1,b2) ∧ SameBib(b2,b3) ⇒ SameBib(b1,b3)

SameAuthor(a1,a2) ∧ SameAuthor(a2,a3) ⇒ SameAuthor(a1,a3)

Same rule for **title**

Same rule for **venue**

Link fields equivalence to citation equivalence – *e.g., if two citations are the same, their authors should be the same*

Author(b1, a1) ∧ Author(b2, a2) ∧ SameBib(b1, b2) ⇒ SameAuthor(a1, a2)

…and that citations with the same author are more likely to be the same

Author(b1, a1) ∧ Author(b2, a2) ∧ SameAuthor(a1, a2) ⇒ SameBib(b1, b2)

Same rules for **title**

Same rules for **venue**

Benefits of MLN model

Standard approach: build a classifier that given two citations tells you if they are the same or not, and then apply transitive closure

New MLN approach:

- performs *collective* entity resolution, where resolving one pair of entities helps to resolve pairs of related entities

e.g., inferring that a pair of citations are equivalent can provide evidence that the names *AAAI-06* and *21st Natl. Conf. on AI* refer to the same venue, even though they are superficially very different. This equivalence can then aid in resolving other entities.

Other MLN applications

- **Information Extraction**
- **Co-reference Resolution (see lecture 1!)**
- **Robot Mapping** (infer the map of an indoor environment from laser range data)
- **Link-based Clustering** (uses relationships among the objects in determining similarity)
- **Ontologies extraction from Text**
- **…..**

Summary of tutorial on MLN for NLP at NA-ACL (2010)

- We need to unify logical and statistical NLP
- **Markov logic** provides a language for this
  - **Syntax:** Weighted first-order formulas
  - **Semantics:** Feature templates of Markov nets
  - **Inference:** Satisfiability, MCMC, lifted BP, etc.
  - **Learning:** Pseudo-likelihood, VP, PSCG, ILP, etc.
- Growing set of NLP applications
- Open-source software: Alchemy

alchemy.cs.washington.edu

- Book: Domingos & Lowd, ***Markov Logic***, Morgan & Claypool, 2009.

Learning Goals for today's class

You can:

- Show on an example how MLNs generalize FOL
- Compute the most likely *pw* given some evidence
- Probability of a formula, Conditional Probability

Next class on Mon (last class)

- Markov Logic: applications
- Watson….
- Beyond 322/422 (ML + grad courses)
- AI conf. and journals
- Final Exam (office hours, samples)

Assignment-4 due on Mon (last class)

Marked Summaries for last paper discussion

The MaxWalkSAT Algorithm

for *i* ← 1 to *max-tries* **do**
    solution = random truth assignment
    for *j* ← 1 **to** *max-flips* **do**
        if ∑ weights(sat. clauses) > threshold **then**
            return *solution*
        c ← random unsatisfied clause
        with probability *p*
            flip a random variable in *c*
        else
            flip variable in *c* that maximizes
                ∑ weights(sat. clauses)
return failure, best *solution* found

Markov Logic Network

What is the probability that a formula F_1 holds given that formula F_2 does?

$$P(F_1 \mid F_2, L, C)$$

$$= P(F_1 \mid F_2, M_{L,C})$$

$$= \frac{P(F_1 \wedge F_2, M_{L,C})}{P(F_2, M_{L,C})}$$

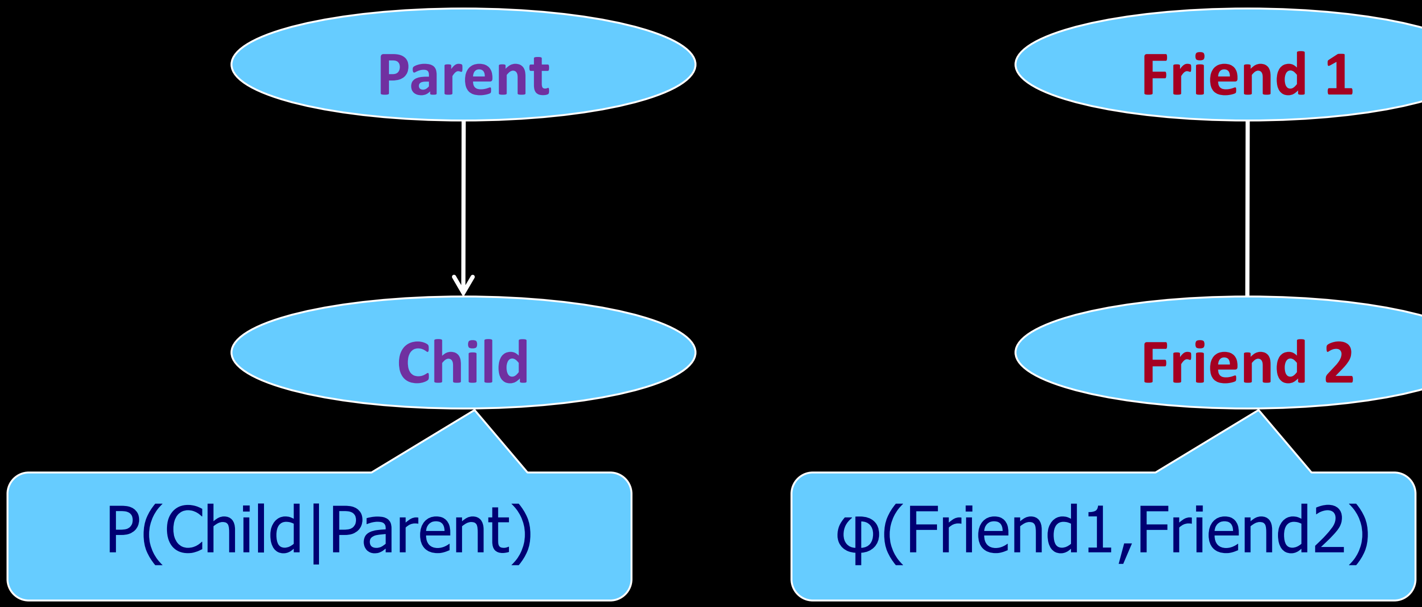
$$= \frac{\sum_{x \in \chi_{F_1} \cap x \in \chi_{F_2}} P(X = x, M_{L,C})}{\sum_{x \in \chi_{F_2}} P(X = x, M_{L,C})}$$

Computing Probabilities

P(Formula|Formula2,MLN,C) = ?

- Discard worlds where Formula 2 does not hold
- In practice: More efficient alternatives

Directed Models vs. Undirected Models



Undirected Probabilistic Logic Models

- Upgrade undirected propositional models to relational setting
  - Markov Nets → Markov Logic Networks
  - Markov Random Fields → Relational Markov Nets
  - Conditional Random Fields → Relational CRFs

Markov Logic Networks (Richardson & Domingos)

- Soften logical clauses
  - A first-order clause is a **hard** constraint on the world

$$\forall x, \mathit{person}(x) \rightarrow \exists y, \mathit{person}(y), \mathit{father}(x, y)$$

  - Soften the constraints so that when a constraint is violated, the world is less probably, not impossible

$$w: \mathit{friends}(x, y) \wedge \mathit{smokes}(x) \rightarrow \mathit{smokes}(y)$$

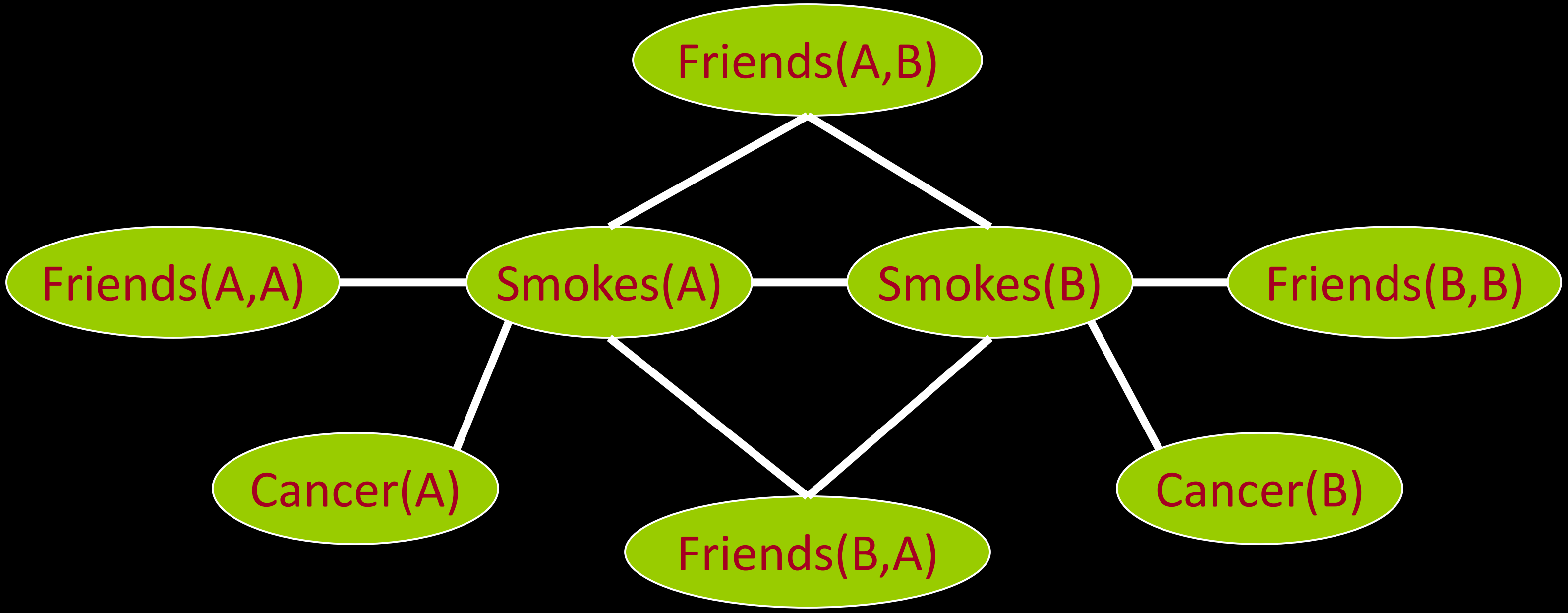
  - Higher weight $\Rightarrow$ Stronger constraint
  - Weight of ∞ $\Rightarrow$ first-order logic

Probability(World S) = (1 / Z) $\times$ exp { Σ weight $_i$ x **numberTimesTrue(f $_i$, S)** }

Example: Friends & Smokers

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

Two constants: Anna (A) and Bob (B)



Alphabetic Soup => Endless Possibilities

- Probabilistic Relational Models (PRM)
- Bayesian Logic Programs (BLP)
- PRISM
- Stochastic Logic Programs (SLP)
- Independent Choice Logic (ICL)
- Markov Logic Networks (MLN)
- Relational Markov Nets (RMN)
- CLP-BN
- Relational Bayes Nets (RBN)
- Probabilistic Logic Progam (PLP)
- ProbLog

….

- Web data (**web**)
- Biological data (**bio**)
- Social Network Analysis (**soc**)
- Bibliographic data (**cite**)
- Epidimiological data (**epi**)
- Communication data (**comm**)
- Customer networks (**cust**)
- Collaborative filtering problems (**cf**)
- Trust networks (**trust**)

…

Fall 2003– Dietterich @ OSU, Spring 2004 –Page @ UW, Spring 2007-Neville @ Purdue, Fall 2008 – Pedro @ CMU

Recent Advances in SRL Inference

- Preprocessing for Inference
  - FROG – Shavlik & Natarajan (2009)
- Lifted Exact Inference
  - Lifted Variable Elimination – Poole (2003), Braz et al(2005) Milch et al (2008)
  - Lifted VE + Aggregation – Kisynski & Poole (2009)
- Sampling Methods
  - MCMC techniques – Milch & Russell (2006)
  - Logical Particle Filter – Natarajan et al (2008), ZettleMoyer et al (2007)
  - Lazy Inference – Poon et al (2008)
- Approximate Methods
  - Lifted First-Order Belief Propagation – Singla & Domingos (2008)
  - Counting Belief Propagation – Kersting et al (2009)
  - MAP Inference – Riedel (2008)
- Bounds Propagation
  - Anytime Belief Propagation – Braz et al (2009)

Conclusion

- Inference is the key issue in several SRL formalisms
- FROG - Keeps the count of unsatisfied groundings
  - Order of Magnitude reduction in number of groundings
  - Compares favorably to Alchemy in different domains
- Counting BP - BP + grouping nodes sending and receiving identical messages
  - Conceptually easy, scaleable BP algorithm
  - Applications to challenging AI tasks
- Anytime BP – Incremental Shattering + Box Propagation
  - Only the most necessary fraction of model considered and shattered
  - Status – Implementation and evaluation

Relation to Statistical Models

- Special cases:
  - Markov networks
  - Markov random fields
  - Bayesian networks
  - Log-linear models
  - Exponential models
  - Max. entropy models
  - Gibbs distributions
  - Boltzmann machines
  - Logistic regression
  - Hidden Markov models
  - Conditional random fields

- Obtained by making all predicates zero-arity

- Markov logic allows objects to be interdependent (non-i.i.d.)