Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 30

Nov, 20, 2019

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

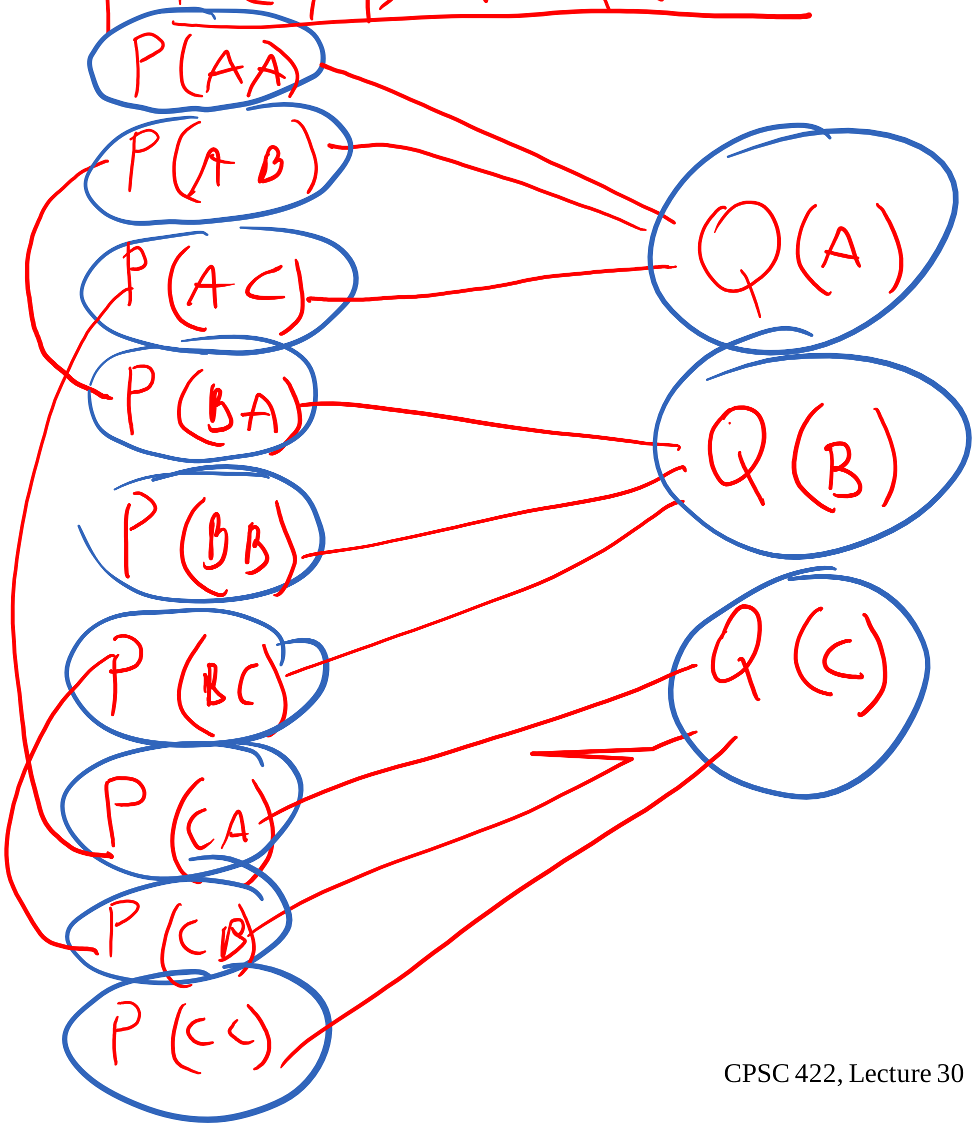
$\Rightarrow$ MARKOV LOGIC NETWORK

$w_1 \; \forall x y \; P(\overset{A}{x}, \overset{B}{y}) \Leftrightarrow P(\overset{B}{y}, \overset{A}{x})$

$w_2 \; \forall x y \; \boxed{P(x, y) \vee Q(x)}$

Constants
A B C

Corresponding Markov Network



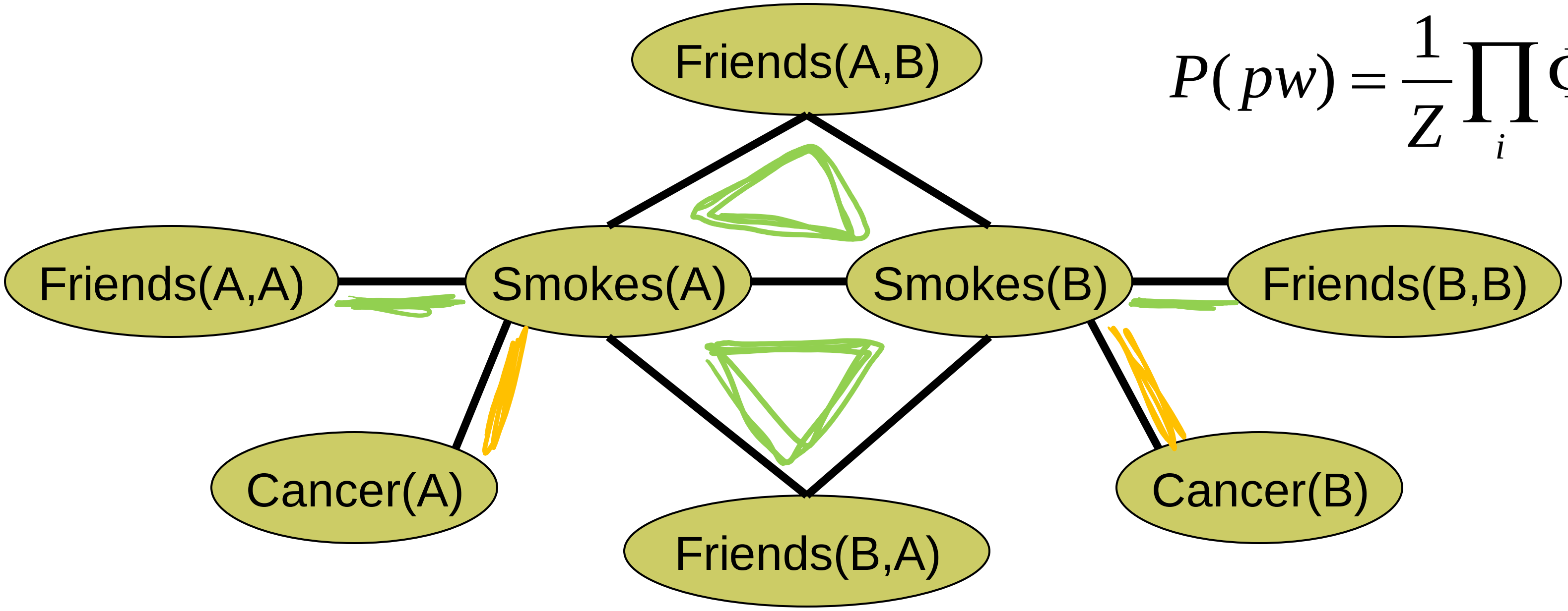
Second example

12 groundings of the predicates
2^12 possible worlds / interpretations

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 $e^0 = 1$

pw_2 ....

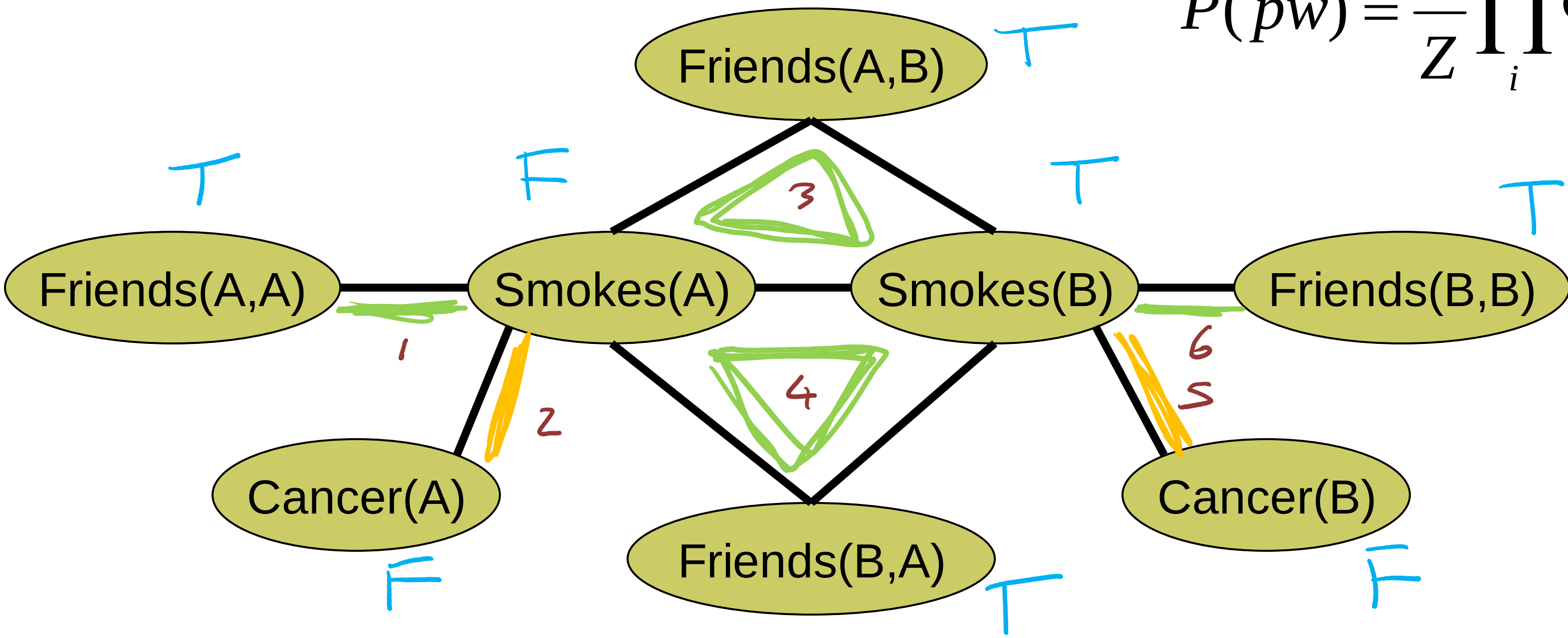
Smokes(A) T
Cancer(A) T $e^{1.5}$

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)

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



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

MLN: prob. Of possible world

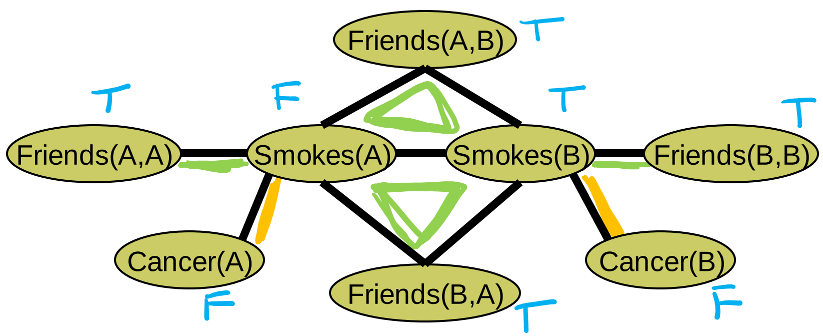
1	1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
2	1.1	$\forall x, y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

- 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

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

$R(A) \overset{\Phi_1}{\text{———}} S(A)$

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

$Z = 1 + 3e^w$

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	$e^w / 1+3e^w$

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

$w \to \infty$ $\quad$ 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*:

$$P(pw) = \frac{1}{Z}\exp\left(\sum_i \underbrace{w_i}_{\text{Weight of formula } i}\ \underbrace{n_i(pw)}_{\text{No. of true groundings of formula } i \text{ in } pw}\right)$$

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

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?

iclicker

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 *f(pw)* **:** 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)* **:** ∑ 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 ∑ 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

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

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

S all samples

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

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

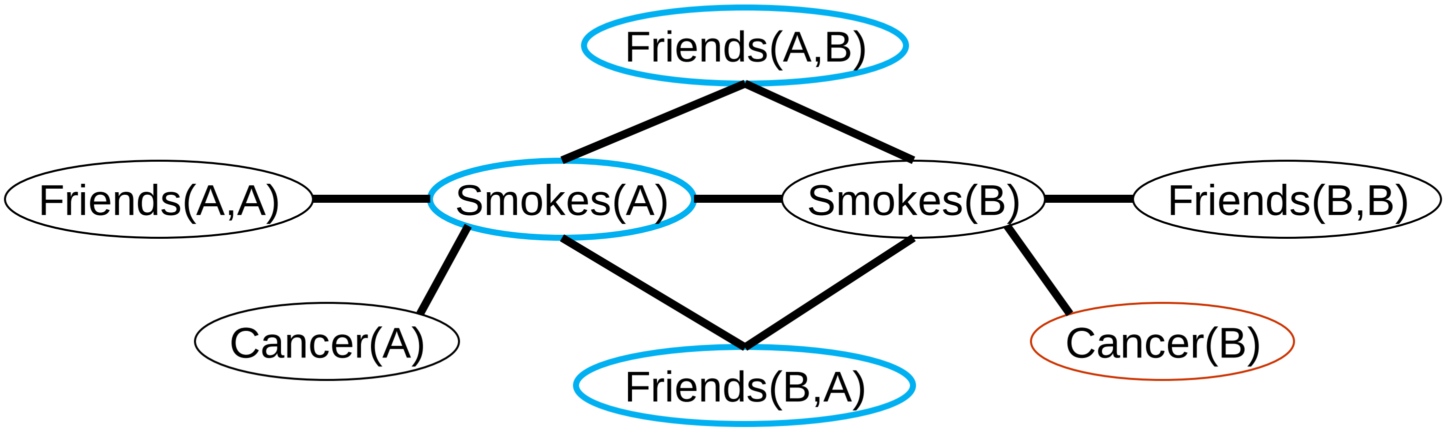
Computing Cond. Probabilities

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))



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

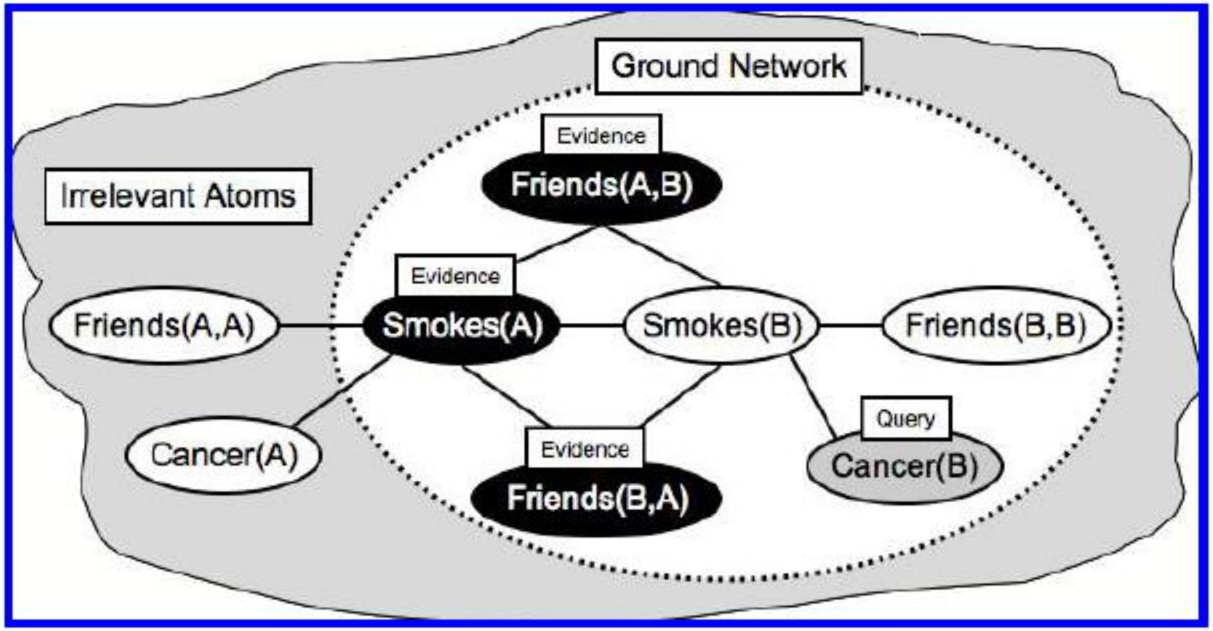
A. Yes B. No C. It depends....

Computing Cond. Probabilities

Let’s look at the simplest case

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

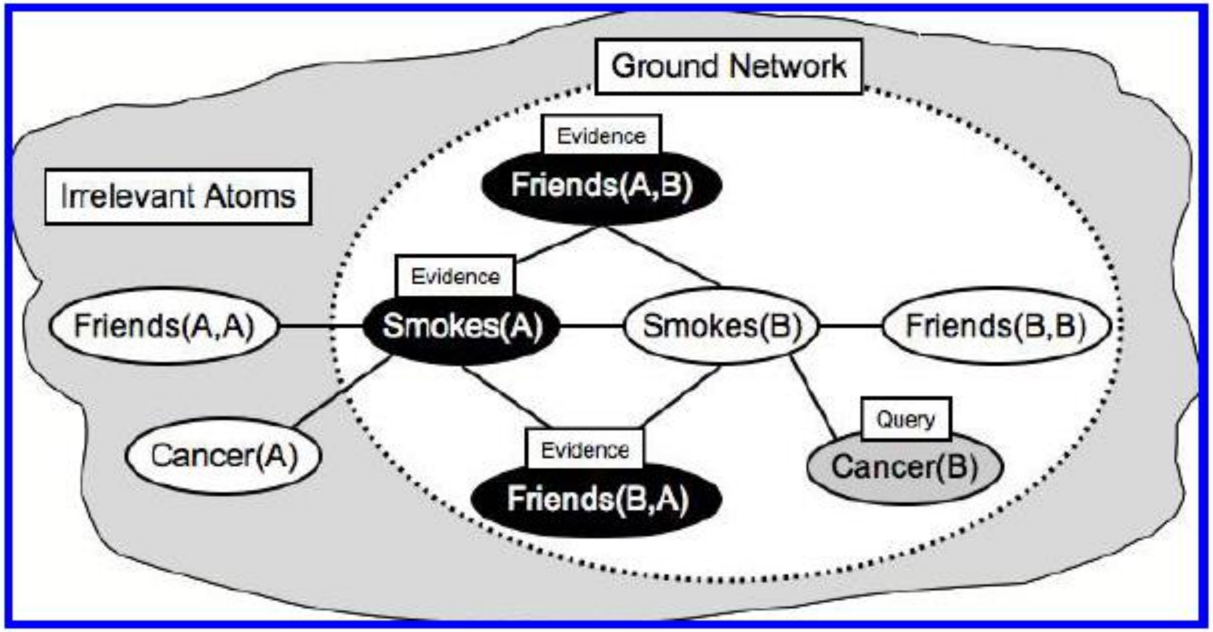
P(Cancer(B)| Smokes(A), Friends(A, B), Friends(B, A))



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 **Nov 29**

In the past, a similar hw took students between 8 - 15 hours to complete. Please start working on it as soon as possible!