

Intelligent Systems (AI-2)

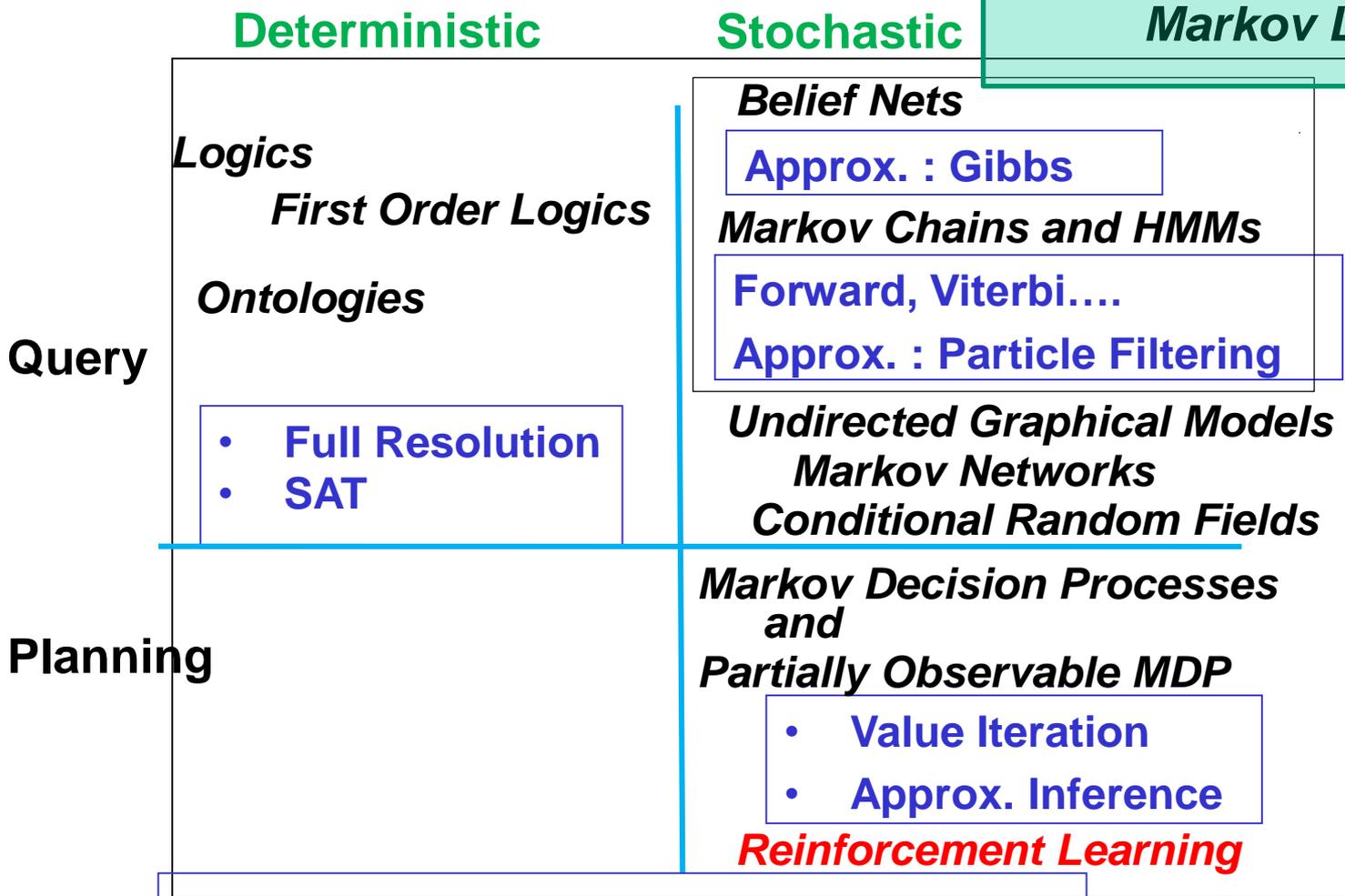
Computer Science cpsc422, Lecture 28

March, 26, 2021

Slide source: from Pedro Domingos UW

422 big picture

StarAI (statistical relational AI)
 Hybrid: Det +Sto
 Prob CFG
 Prob Relational Models
 Markov Logics



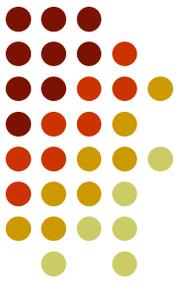
Applications of AI

Representation
 Reasoning
 Technique

Lecture Overview

- **Statistical Relational (**Star-AI**) Models (*for us aka Hybrid*)**
- **Recap Markov Networks and log-linear models**
- **Markov Logic**

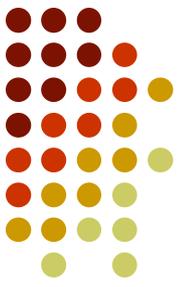
Statistical Relational Models



Goals:

- Combine **(subsets of) logic** and **probability** into a single language (R&R system)
- Develop efficient **inference** algorithms
- Develop efficient **learning** algorithms
- Apply to real-world problems

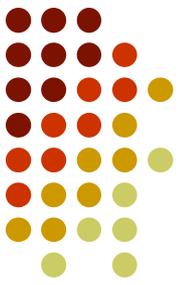
L. Getoor & B. Taskar (eds.), *Introduction to Statistical Relational Learning*, MIT Press, 2007.



Plethora of Approaches

- Knowledge-based model construction [Wellman et al., 1992]
- Stochastic logic programs [Muggleton, 1996]
- Probabilistic relational models [Friedman et al., 1999]
- Relational Markov networks [Taskar et al., 2002]
- Bayesian logic [Milch et al., 2005]
- Markov logic [Richardson & Domingos, 2006]
- *And many others.....!*

Prob. Rel. Models vs. Markov Logic



PRM

- Relational Skeleton
 - Dependency Graph
 - Parameters (CPT)
- } \Rightarrow BNENET

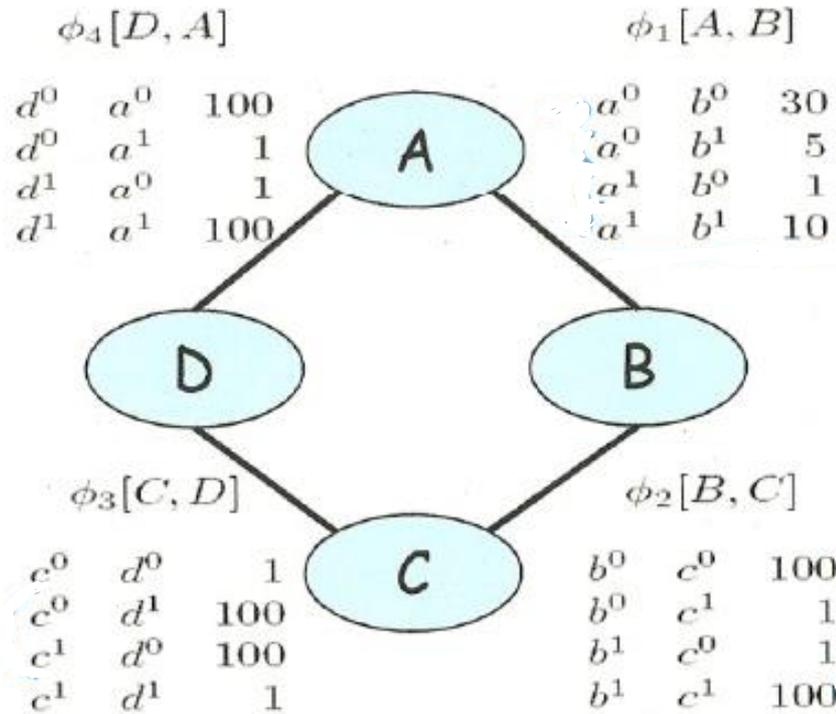
ML

- weighted logical formulas
 - set of constants
- } \Rightarrow MARKOV LOGIC NETWORK

Lecture Overview

- Statistical Relational Models (*for us aka Hybrid*)
- **Recap Markov Networks and log-linear models**
- Markov Logic
 - Markov Logic Network (MLN)

Parameterization of Markov Networks



X set of random
vars: A factor is
 $\underline{\Phi}(\text{val}(X)) \rightarrow \mathbb{R}$

Factors define the local interactions (like CPTs in Bnets)

What about the global model? What do you do with Bnets?

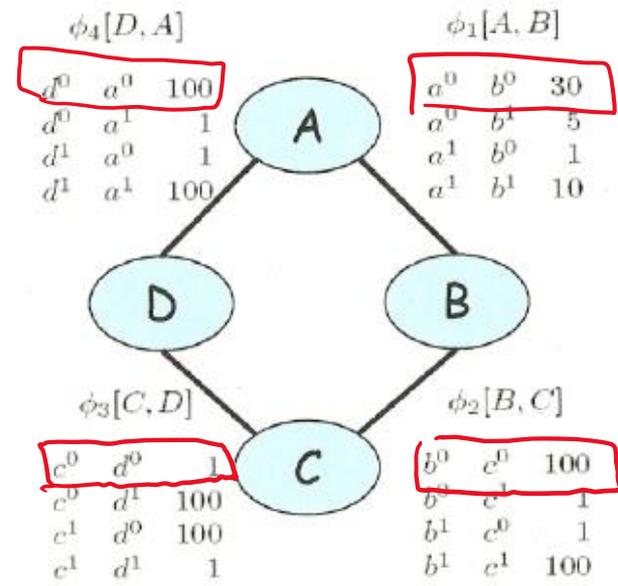
How do we combine local models?

As in BNets by multiplying them!

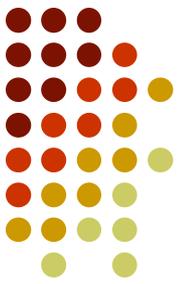
$$\tilde{P}(A, B, C, D) = \overset{30}{\phi_1}(A, B) \times \overset{100}{\phi_2}(B, C) \times \overset{1}{\phi_3}(C, D) \times \overset{100}{\phi_4}(A, D)$$

$$P(A, B, C, D) = \frac{1}{Z} \tilde{P}(A, B, C, D)$$

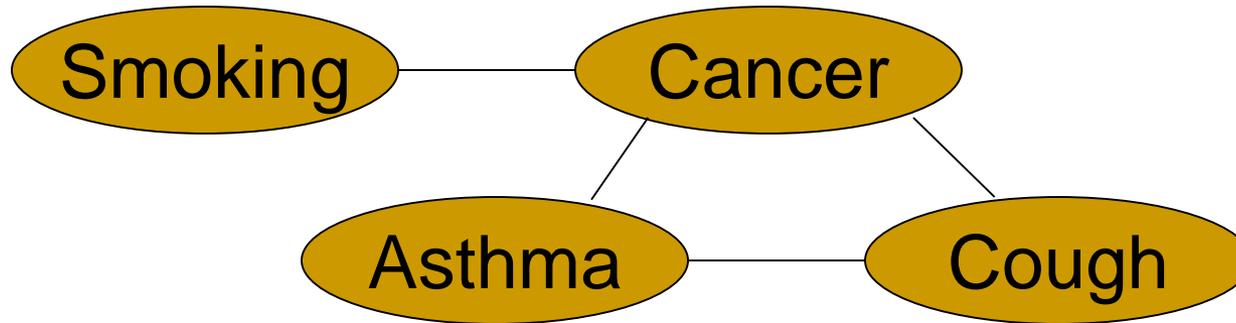
Assignment				Unnormalized	Normalized
a^0	b^0	c^0	d^0	300000	.04
a^0	b^0	c^0	d^1	300000	.04
a^0	b^0	c^1	d^0	300000	.04
a^0	b^0	c^1	d^1	30	4.1×10^{-6}
a^0	b^1	c^0	d^0	500	⋮
a^0	b^1	c^0	d^1	500	⋮
a^0	b^1	c^1	d^0	5000000	.69
a^0	b^1	c^1	d^1	500	⋮
a^1	b^0	c^0	d^0	100	⋮
a^1	b^0	c^0	d^1	1000000	⋮
a^1	b^0	c^1	d^0	100	⋮
a^1	b^0	c^1	d^1	100	⋮
a^1	b^1	c^0	d^0	10	⋮
a^1	b^1	c^0	d^1	100000	⋮
a^1	b^1	c^1	d^0	100000	⋮
a^1	b^1	c^1	d^1	100000	⋮



Markov Networks



- **Undirected** graphical models



- Factors/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	$\Phi(S,C)$
F	F	4.5
F	T	4.5
T	F	2.7
T	T	4.5

Markov Networks :log-linear model



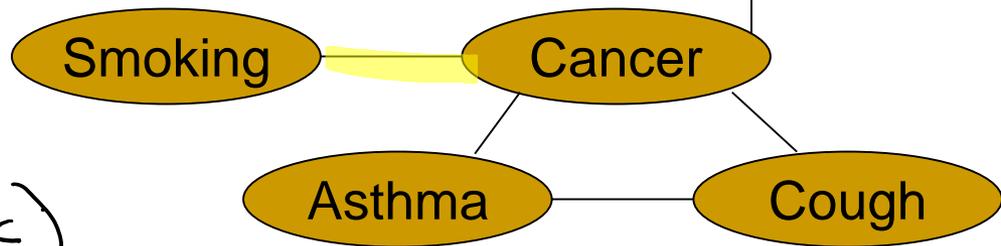
$$P(x) = \frac{1}{Z} \prod_c \Phi_c(x_c)$$

- Log-linear model:

each $\Phi_c(x_c) = e^{w_c t_c(x_c)}$

$w_1 = 0.51$

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



$$P(x) = \frac{1}{Z} \exp \left(\sum_i w_i f_i(x_i) \right)$$

Weight of Feature i Feature i

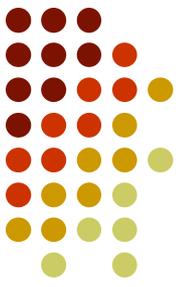
Smoking	Cancer	f_1
1	1	0
1	0	1
0	1	1
0	0	0

$e^{0.51}$
 e^0
 $e^{0.51}$
 $e^{-0.51}$

Lecture Overview

- Statistical Relational Models (for us aka Hybrid)
- Recap Markov Networks
- **Markov Logic**

Markov Logic: Intuition(1)



- A logical KB is a set of **hard constraints** on the set of possible worlds

f

$\forall x \text{Smokes}(x) \Rightarrow \text{Cancer}(x)$

CONSTANT
INDIVIDUALS = {a, b}

Smokes(a) = T
Cancer(a) = F
Smokes(b) = F
Cancer(b) = F

Anna **A** and Bob **B**

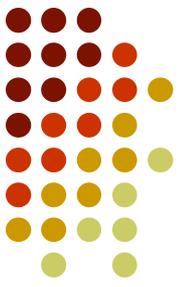
pw
possible world

in FO2 pw is...

A. possible B. impossible
C. cannot tell



Markov Logic: Intuition(1)



- A logical KB is a set of **hard constraints** on the set of possible worlds

f

$\forall x \text{Smokes}(x) \Rightarrow \text{Cancer}(x)$

In FOL \hat{p}_w is impossible

INDIVIDUALS = {a, b}

~~Smokes(a) = T~~

~~Cancer(a) = F~~

~~Smokes(b) = F~~

~~Cancer(b) = F~~

} \hat{p}_w

- Let's make them **soft constraints**:

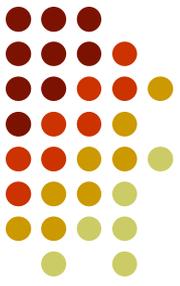
iclicker.

When a world violates a formula,

the world becomes less probable, not impossible

if f is True $\rightarrow P(\hat{w})$ decreases
False in $\hat{p}_w \rightarrow P(\hat{w})$ increases

Markov Logic: Intuition (2)



- The more formulas in the KB a possible world satisfies the more it should be likely
- Give each formula a **weight**
- Adopting a **log-linear model**, by design, if a possible world satisfies a formula its **probability** should go up proportionally to **exp(the formula weight)**.

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

That is, if a possible world satisfies a formula its **log probability** should go up proportionally to the formula weight.

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

Markov Logic: 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 C of **constants**,
- It defines a **Markov network** with
 - One *binary node* for each **grounding** of each **predicate** in the MLN
 - One *feature/factor* for each **grounding** of each **formula F** in the MLN, with the corresponding weight w

Grounding:
substituting vars
with constants

(not required) consider Existential and functions

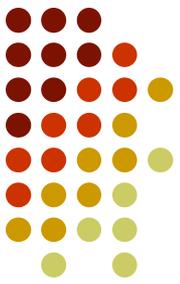


Table 2.2: Construction of all groundings of a first-order formula under Assumptions 2.2–2.4.

function $\text{Ground}(F)$

input: F , a formula in first-order logic

output: G_F , a set of ground formulas

for each existentially quantified subformula $\exists x S(x)$ in F

$F \leftarrow F$ with $\exists x S(x)$ replaced by $S(c_1) \vee S(c_2) \vee \dots \vee S(c_{|C|})$,
where $S(c_i)$ is $S(x)$ with x replaced by c_i

$G_F \leftarrow \{F\}$

for each universally quantified variable x

for each formula $F_j(x)$ in G_F

$G_F \leftarrow (G_F \setminus F_j(x)) \cup \{F_j(c_1), F_j(c_2), \dots, F_j(c_{|C|})\}$,
where $F_j(c_i)$ is $F_j(x)$ with x replaced by c_i

for each formula $F_j \in G_F$

repeat

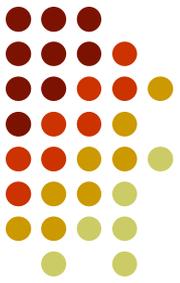
for each function $f(a_1, a_2, \dots)$ all of whose arguments are constants

$F_j \leftarrow F_j$ with $f(a_1, a_2, \dots)$ replaced by c , where $c = f(a_1, a_2, \dots)$

until F_j contains no functions

return G_F

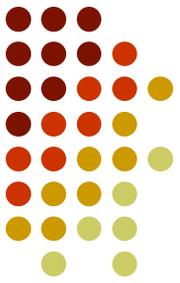
Example: Friends & Smokers



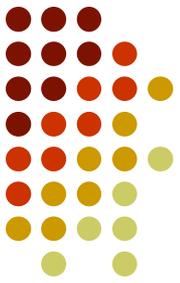
Smoking causes cancer.

Friends have similar smoking habits.

Example: Friends & Smokers


$$\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$$
$$\forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$$

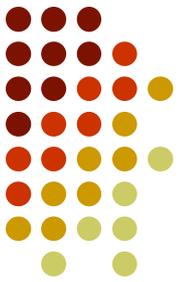
Example: Friends & Smokers



1.5 $\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$

1.1 $\forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$

Example: Friends & Smokers



1.5 $\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$

1.1 $\forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$

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

MLN nodes



1.5 $\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$

1.1 $\forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$

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

- One *binary node* for each grounding of each predicate in the MLN

Grounding:
substituting vars
with constants

Smokes(A)

Smokes(B)

Cancer(A)

Cancer(B)

iclicker.

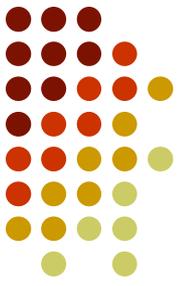
- Any nodes missing?

A. NO

B. Cannot
Tell

C. YES

MLN nodes (complete)

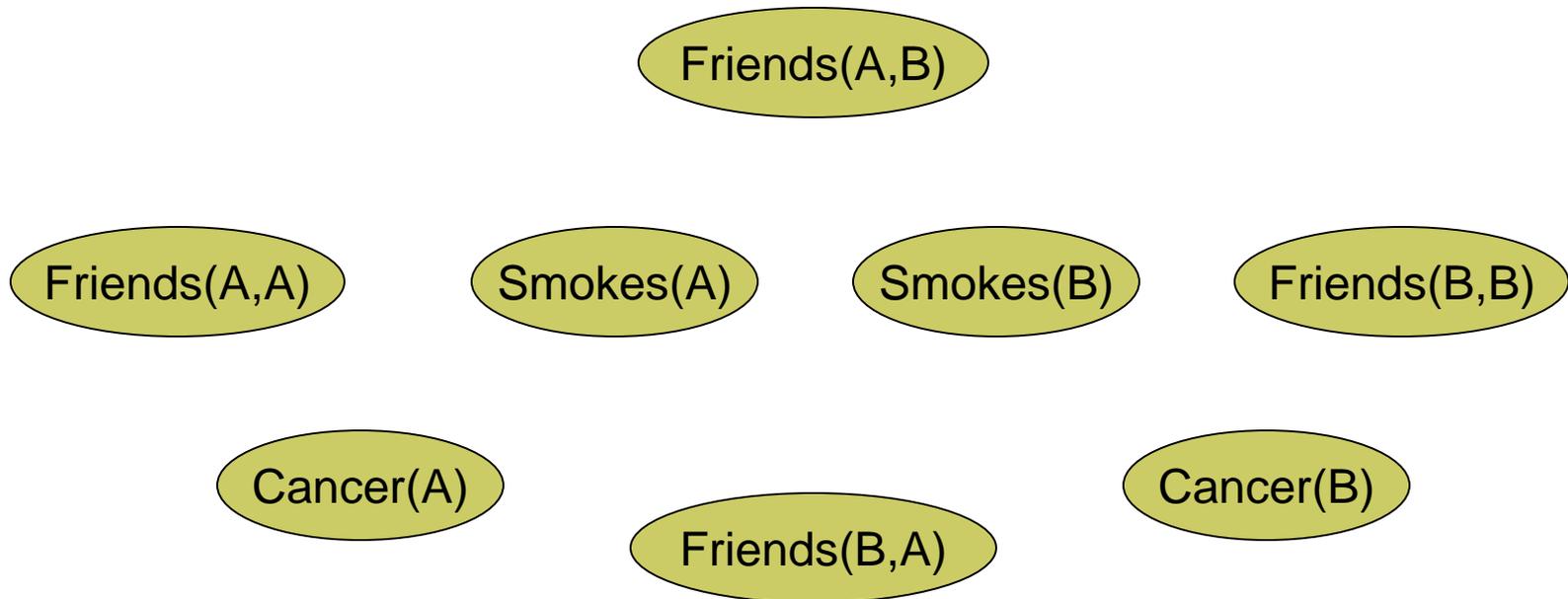


$$1.5 \quad \forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$$

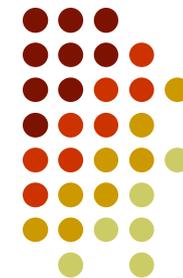
$$1.1 \quad \forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$$

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

- One *binary node* for each grounding of each predicate in the MLN



MLN features

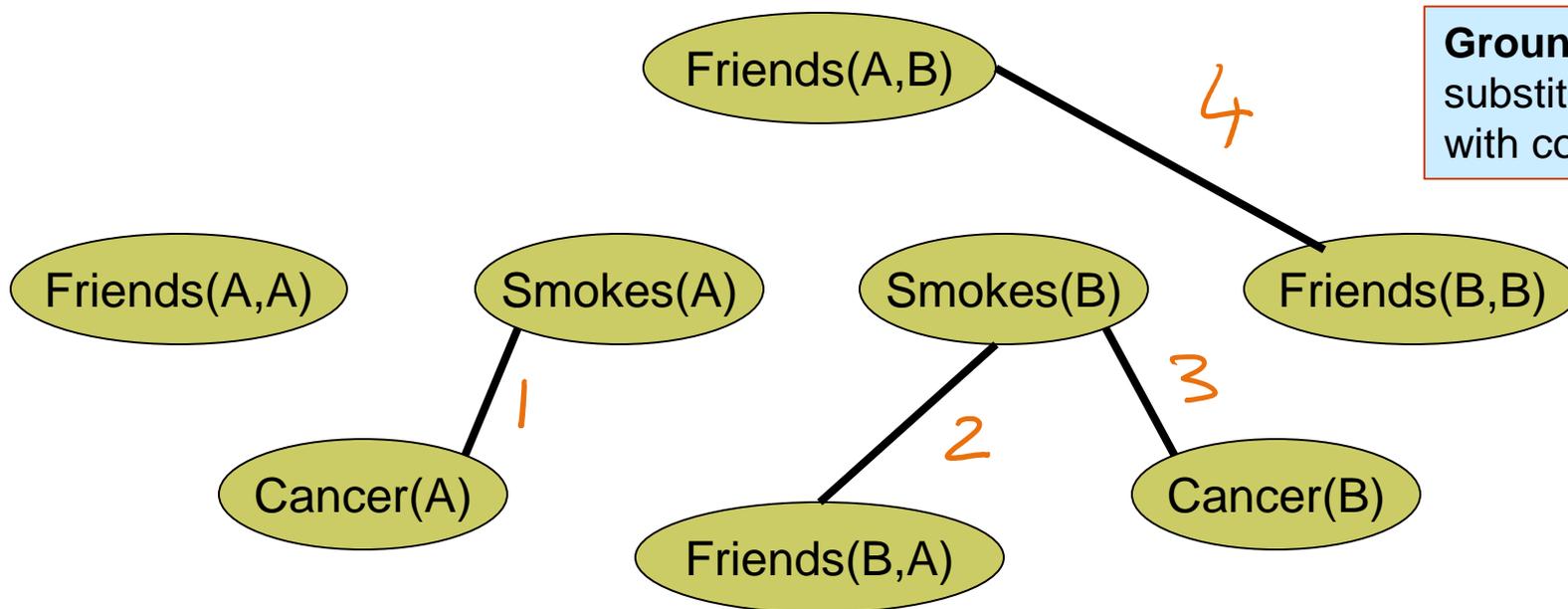


1.5 $\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$

1.1 $\forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$

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

Edge between two nodes iff the corresponding ground predicates appear together in at least one grounding of one formula



iclicker

Which edge should not be there?

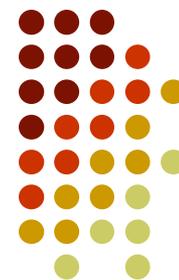
A. 1

B. 2

C. 3

D. 4

MLN features

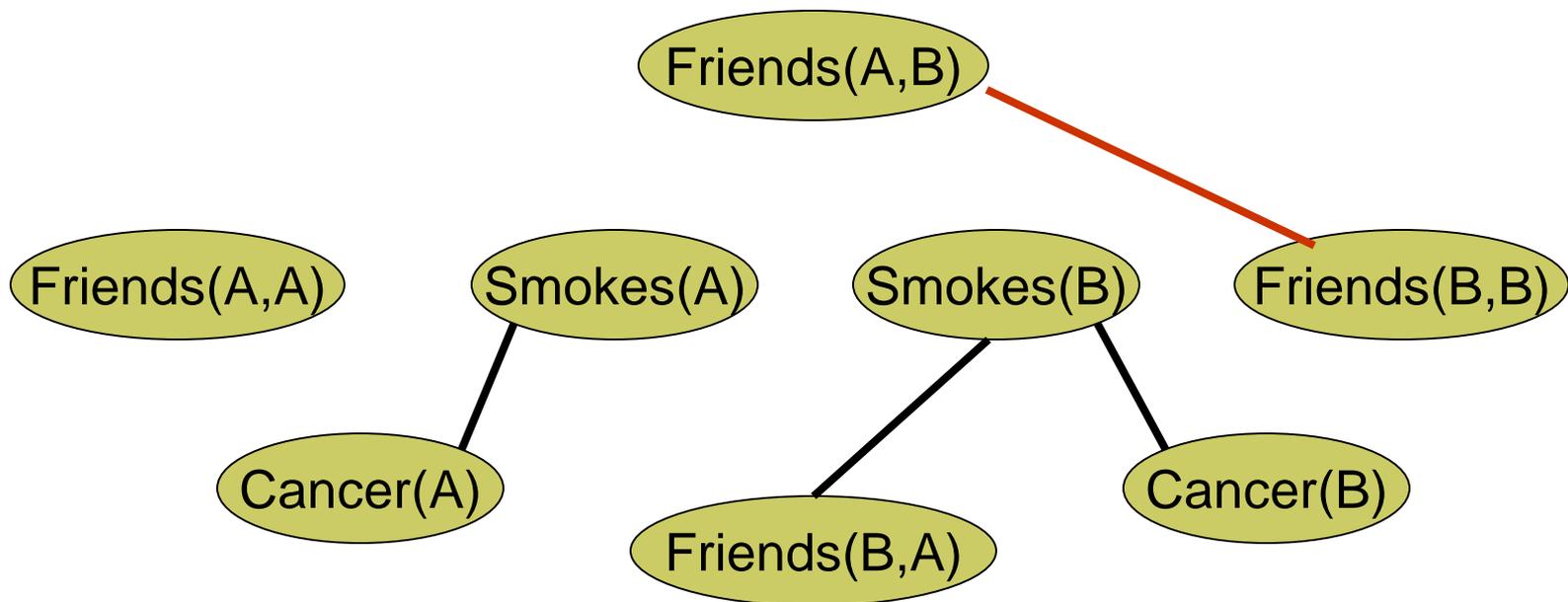


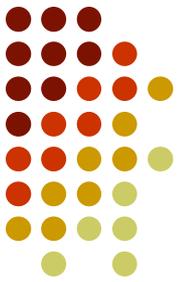
1.5 $\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$

1.1 $\forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$

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

Edge between two nodes iff the corresponding ground predicates appear together in at least one grounding of one formula



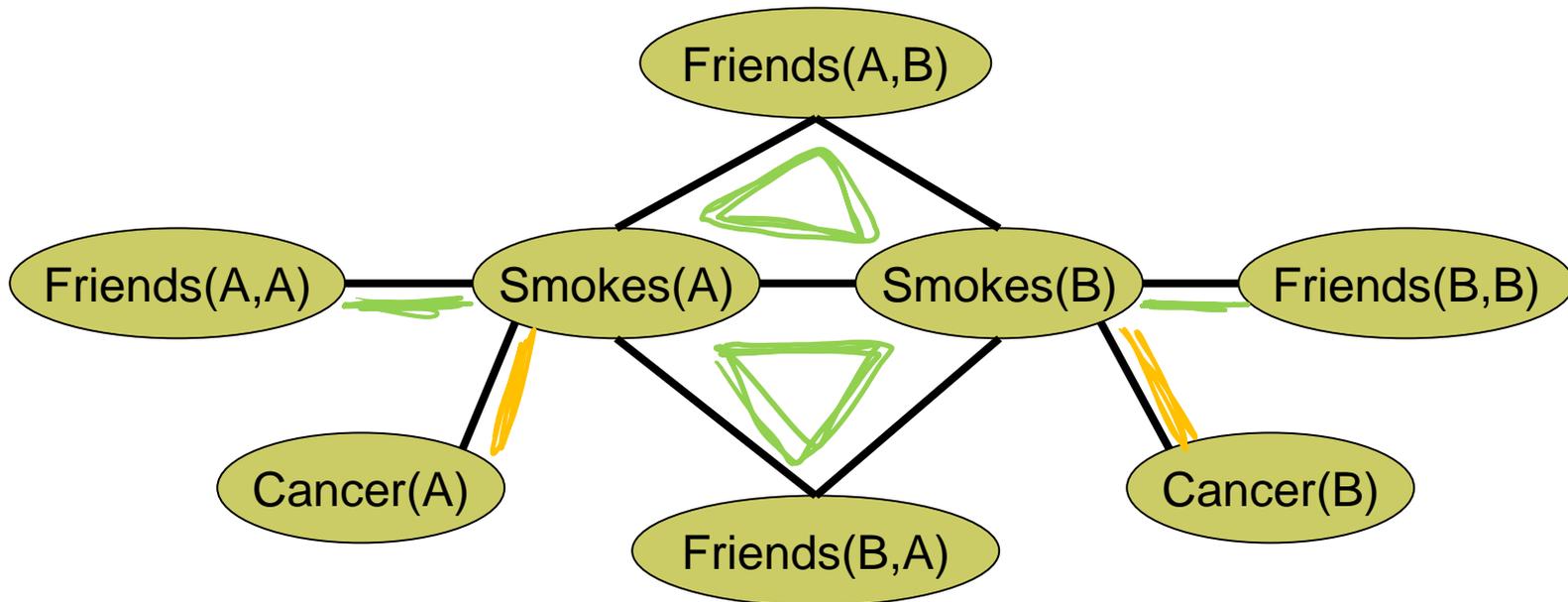


MLN features

1.5 $\forall x \text{Smokes}(x) \Rightarrow \text{Cancer}(x)$

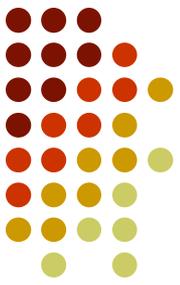
1.1 $\forall x, y \text{Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$

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



One *feature/factor* for each **grounding** of each **formula F** in the MLN

MLN: parameters



- For each formula i we have a **factor**

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

← possible world

w_i weight of formula

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

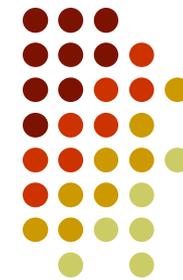
1.5 $\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$

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

pw_1 ...
Smokes(A) T
Cancer(A) F $e^0 = 1$

pw_2 ... $e^{1.5}$
Smokes(A) T
Cancer(A) T

MLN: prob. of possible world

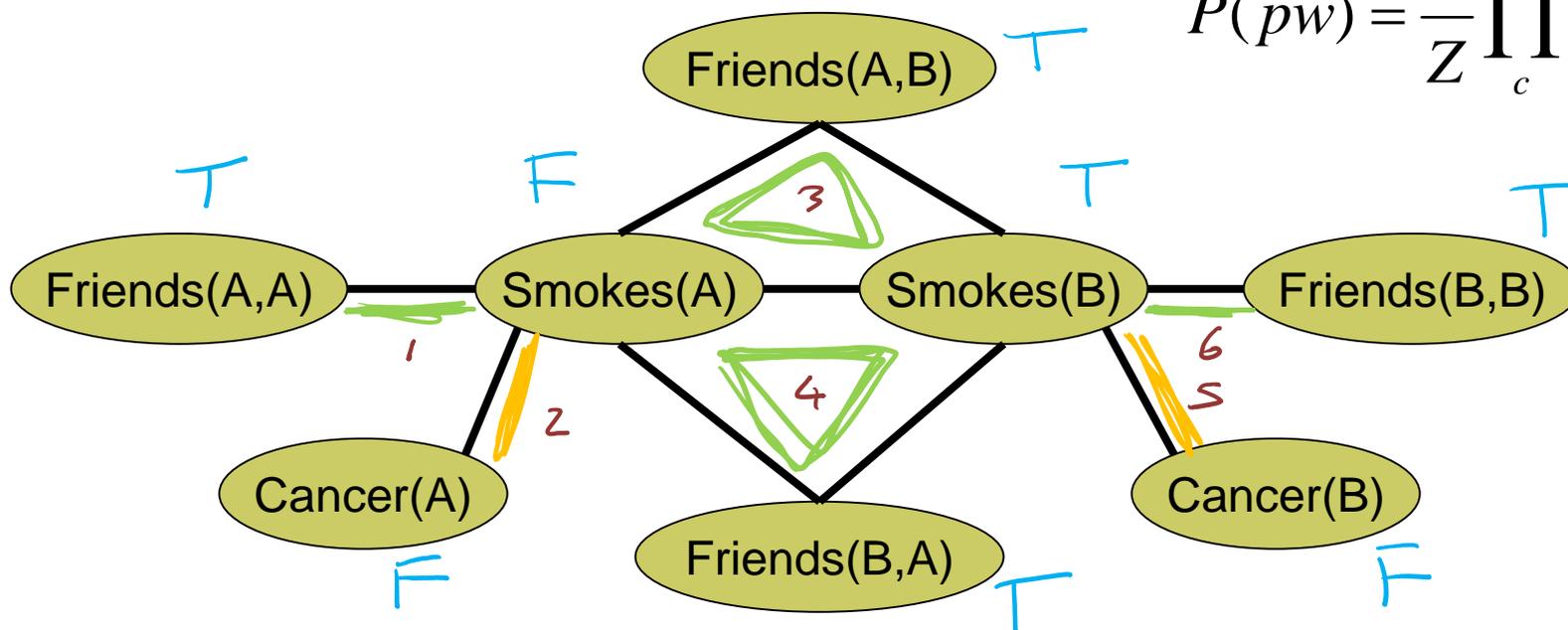


1.5 $\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$

1.1 $\forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$

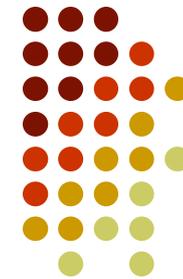
Two constants: **Anna** (A) and **Bob** (B)

$$P(pw) = \frac{1}{Z} \prod_c \Phi_c(pw_c)$$



$$P(pw) = \left(e^{1.1}_1 * e^{1.1}_6 * e^0_3 * e^0_4 * e^{1.5}_2 * e^0_5 \right) / Z$$

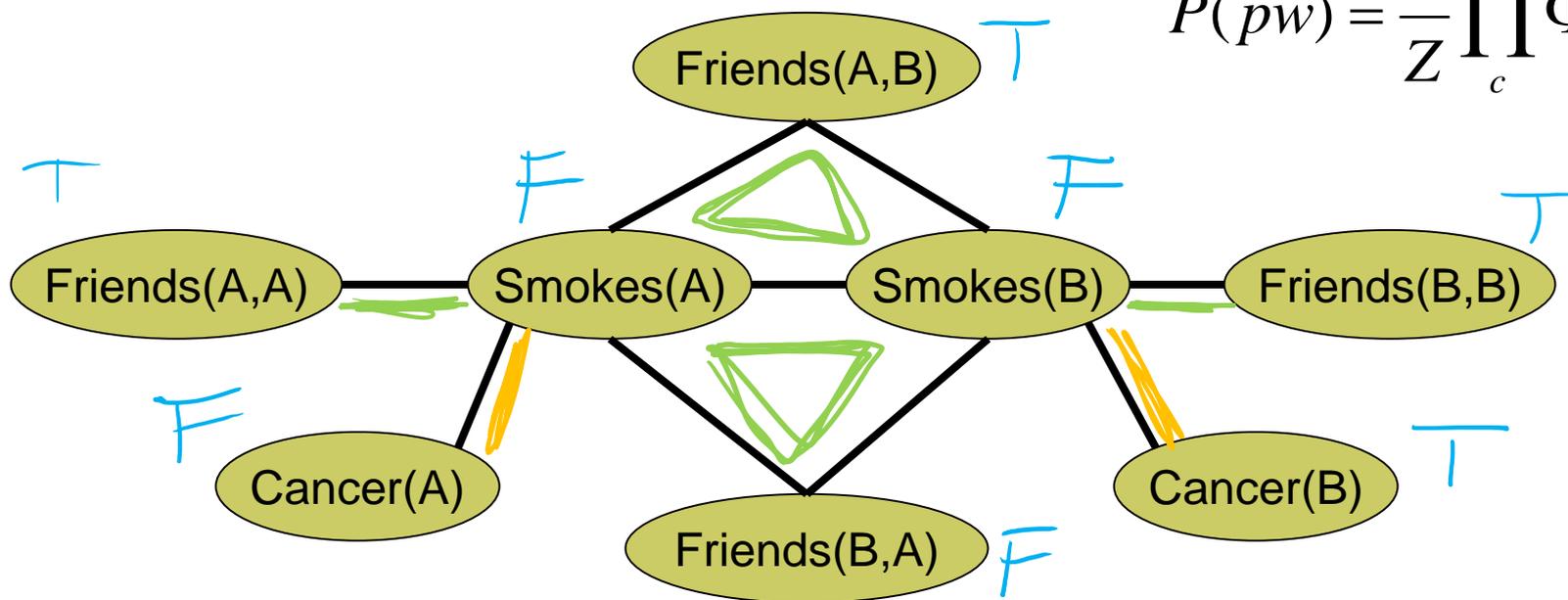
MLN: prob. of possible world



- 1.5 $\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$
- 1.1 $\forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$

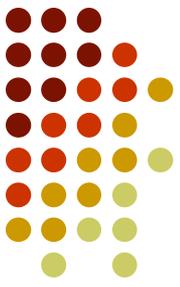
Two constants: **Anna (A)** and **Bob (B)**

$$P(pw) = \frac{1}{Z} \prod_c \Phi_c(pw_c)$$



$$\left(e^{1.1} * e^{1.1} * e^{1.1} * e^{1.1} * e^{1.5} * e^{1.5} \right) / Z$$

MLN: prob. Of possible world

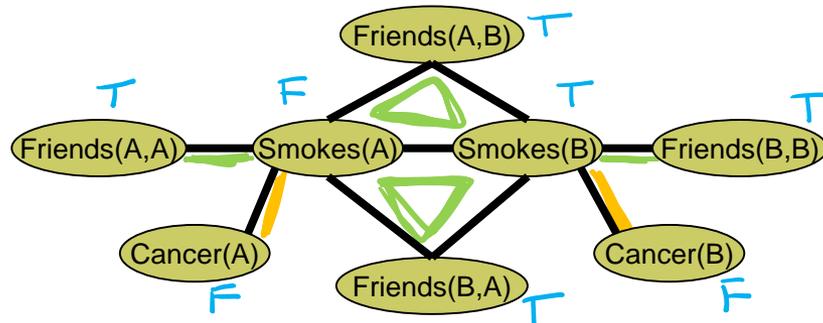


- Probability of a world p_w :

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

Weight of formula i

No. of true groundings of formula i in p_w



$e^{(1.1 * 2)} + (1.5 * 1)$

$$P(p_w) = (e^{1.1} * e^{1.1} * e^0 * e^0 * e^{1.5} * e^0) / Z$$

$$n_2(p_w) = 2 \quad n_1(p_w) = 1$$

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

Learning Goals for today's class

You can:

- Describe the intuitions behind the design of a Markov Logic
- Define and Build a Markov Logic Network
- Justify and apply the formula for computing the probability of a possible world

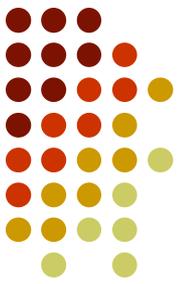
Next class on Mon

Markov Logic

- Relation to FOL
- Inference (MAP and Cond. Prob)

Assignment-3 due on Mon

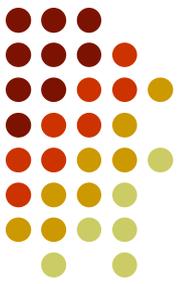
Assignment-4 will be out on the same day



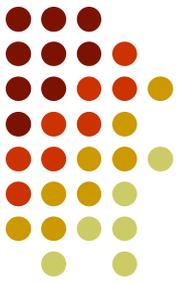
Relation to First-Order Logic

- Example pag 17
- Infinite weights \Rightarrow First-order logic
- Satisfiable KB, positive weights \Rightarrow
Satisfying assignments = Modes of distribution
- Markov logic allows contradictions between formulas

Relation to Statistical Models

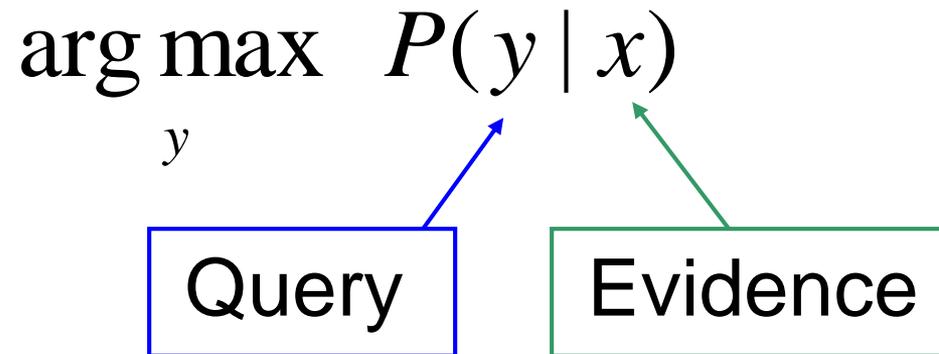


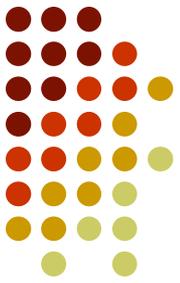
- Special cases:
 - Markov networks
 - Markov random fields
 - Bayesian networks
 - Log-linear models
 - Exponential models
 - Max. entropy models
 - Gibbs distributions
 - Boltzmann machines
 - Logistic regression
 - Hidden Markov models
 - Conditional random fields
- Obtained by making all predicates zero-arity
- Markov logic allows objects to be interdependent (non-i.i.d.)



MAP Inference

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

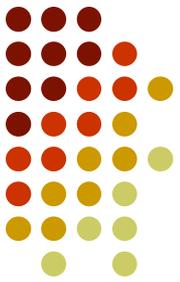




MAP Inference

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

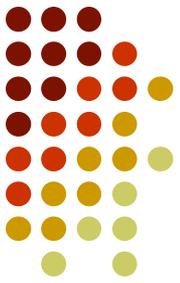
$$\arg \max_y \frac{1}{Z_x} \exp \left(\sum_i w_i n_i(x, y) \right)$$



MAP Inference

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

$$\arg \max_y \sum_i w_i n_i(x, y)$$



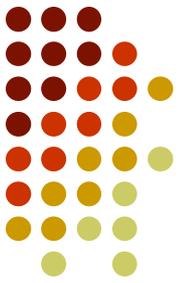
MAP Inference

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

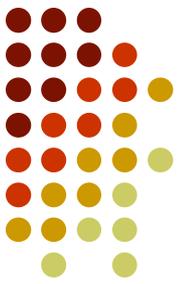
$$\arg \max_y \sum_i w_i n_i(x, y)$$

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

The MaxWalkSAT Algorithm



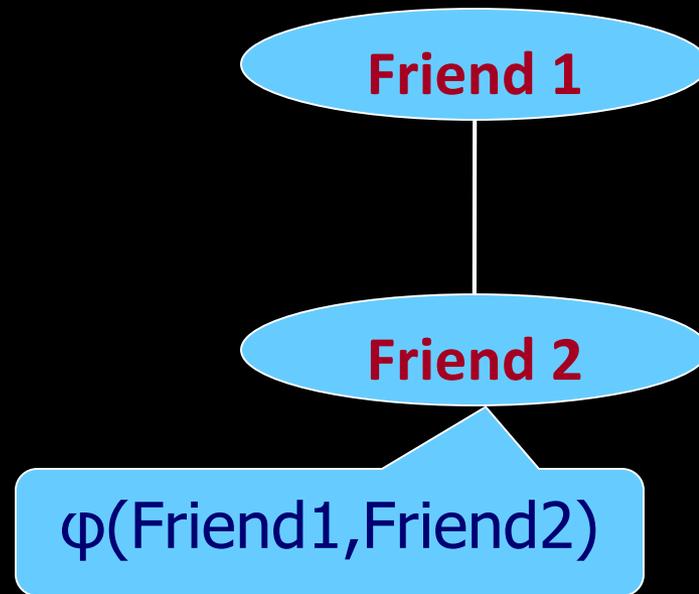
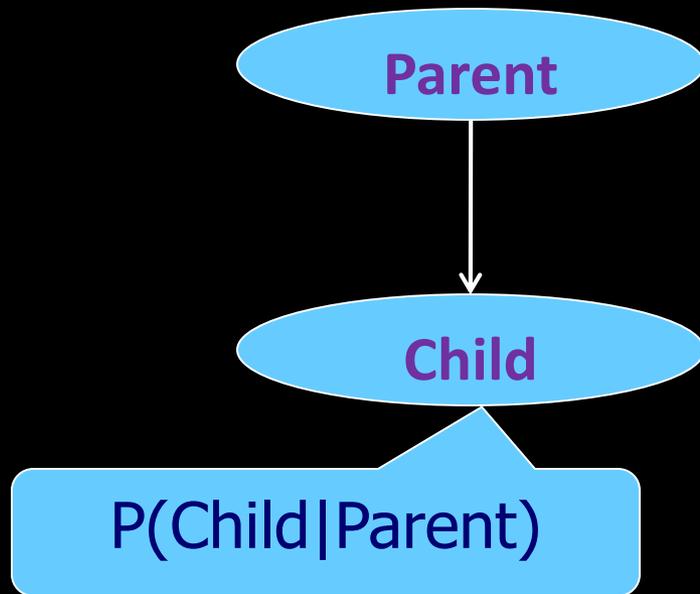
```
for  $i \leftarrow 1$  to max-tries do
  solution = random truth assignment
  for  $j \leftarrow 1$  to max-flips do
    if  $\sum \text{weights}(\text{sat. clauses}) > \text{threshold}$  then
      return solution
     $c \leftarrow$  random unsatisfied clause
    with probability  $p$ 
      flip a random variable in  $c$ 
    else
      flip variable in  $c$  that maximizes
         $\sum \text{weights}(\text{sat. clauses})$ 
  return failure, best solution found
```



Computing Probabilities

- $P(\text{Formula} | \text{MLN}, \text{C}) = ?$
- Brute force: Sum probs. of worlds where formula holds
- MCMC: Sample worlds, check formula holds
- $P(\text{Formula1} | \text{Formula2}, \text{MLN}, \text{C}) = ?$
- 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
 - **Soften** the constraints so that when a constraint is violated, the world is less probably, not impossible
 - **Higher** weight \Rightarrow **Stronger** constraint
 - Weight of \Rightarrow first-order logic

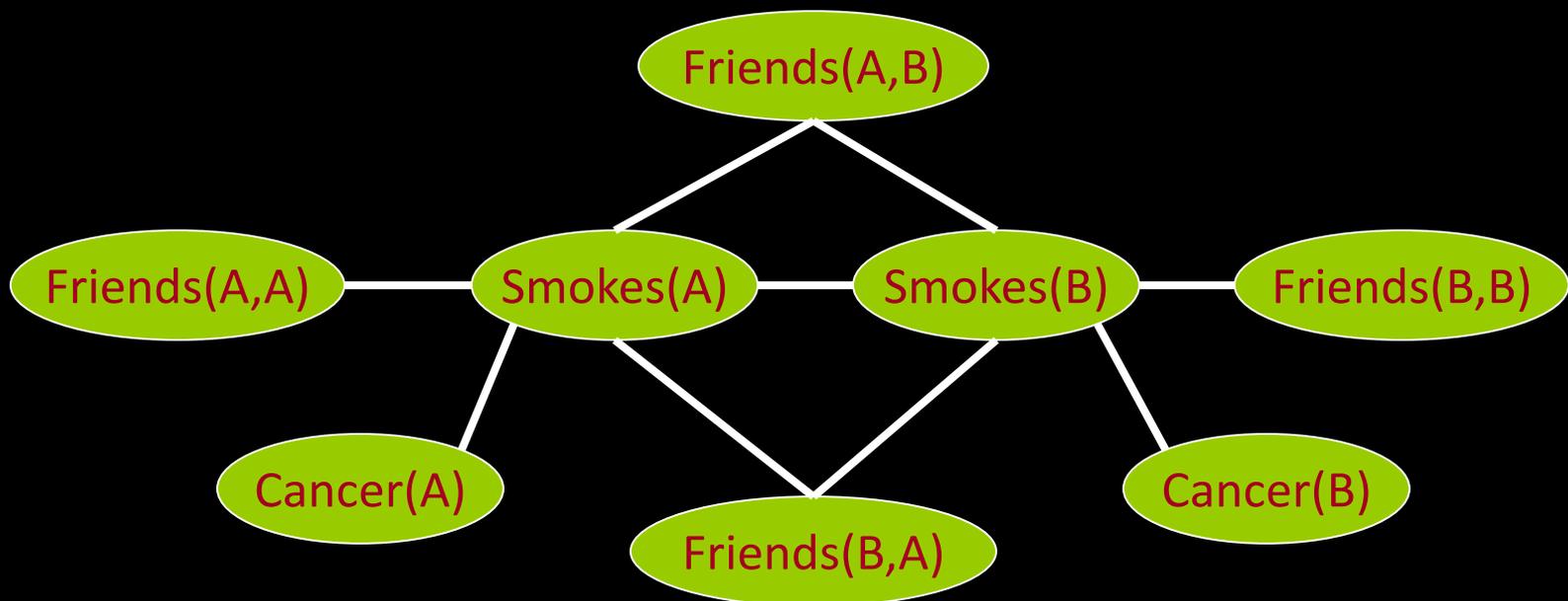
$$\text{Probability}(\text{World } S) = (1 / Z) \times \exp \{ \sum \text{weight}_i \times \text{numberTimesTrue}(f_i, S) \}$$

Example: Friends & Smokers

1.5 $\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$

1.1 $\forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$

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

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, scalable** 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