# Introduction to Artificial Intelligence

## Lecture 20 – Probabilistic first order logic

CS/CNS/EE 154

Andreas Krause

# Announcements

- Exam:
  - 24h contiguous time period to solve the exam
  - Will update details on webpage
  - **Recitation this Thursday**

# Logic and probability

- In this course we covered
- First order logic:
  - Very expressive formal language
  - Very fragile, need to specify all special cases, etc.
  - Does not handle uncertainty well
- Bayesian networks:
  - Allow to quantify uncertainty
  - Can be learnt from data
  - Not very expressive

- Can we have the best of both worlds??

# Probabilistic propositional logic

- Suppose we would like to express uncertainty about *logical propositions*
- Birds can typically fly $P(Bird \Rightarrow CanFly) = .95$
- Propositional symbols ➔ Bernoulli random variables
  - Specify $P(Bird = b, CanFly = f)$
    for all $b, f \in \{true, false\}$
- Probability of a proposition ϕ is the probability mass of all models of ϕ (i.e., all ω that make ϕ true)
- Allows us to avoid specifying large numbers of excepts ("Birds can fly unless X and …")

# Joint distributions

- Instead of random variable, have random vector
$$\mathbf{X}(\omega) = [X_1(\omega), \ldots, X_n(\omega)] \in D^n$$
- Can specify $P(X_1=x_1,\ldots,X_n=x_n)$ directly
(atomic events are assignments $x_1,\ldots,x_n$)
- Joint distribution describes relationship among all variables
- Example:

| | *toothache* | | *¬ toothache* | |
|---|---|---|---|---|
| | *catch* | *¬ catch* | *catch* | *¬ catch* |
| *cavity* | .108 | .012 | .072 | .008 |
| *¬ cavity* | .016 | .064 | .144 | .576 |

Can represent complex distributions using Bayes Nets!

# Limitations of probabilistic prop. logic

- Need one variable for each possible propositional symbol
- Cannot deal with variable number of objects

- First order logic much more expressive
  - Objects, relations, functions
  - Quantifiers allow to express sentences involving many objects

# Probabilistic first order logic

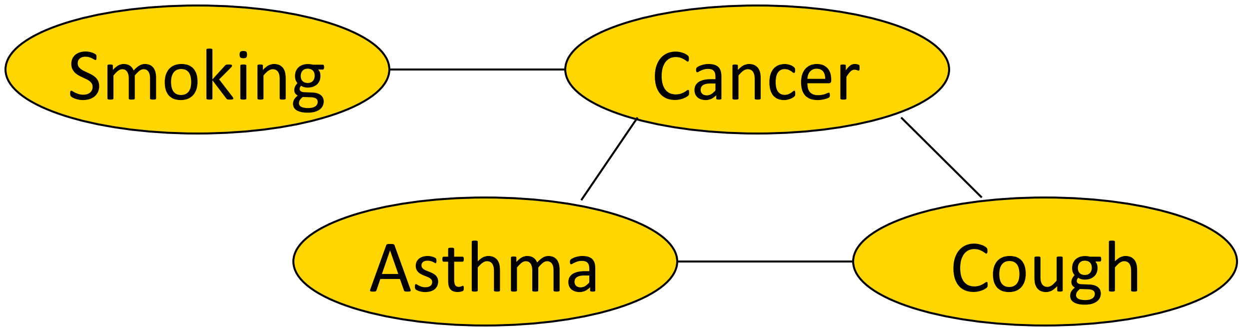
- Several approaches for combining FOL and probability
  - Bayesian logic (Milch, Russell et al.)
  - Markov logic networks (Domingos & Richardson)

➔ Slides about Markov Logic Networks from Pedro Domingos (University of Washington)

http://www.cs.washington.edu/homes/pedrod/803/

# Markov Networks

- **Undirected** graphical models




- Potential functions defined over cliques

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

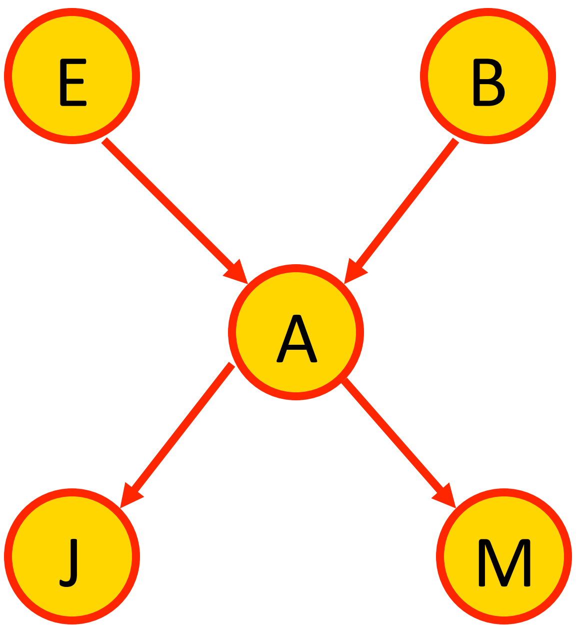
$$Z = \sum_x \prod_c \Phi_c(x_c)$$

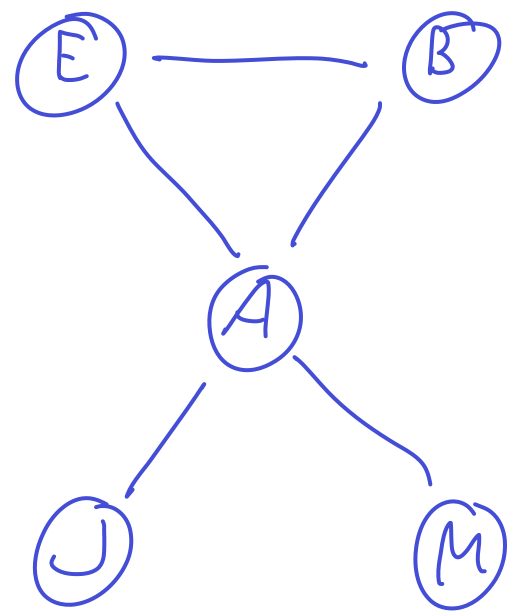
| Smoking | Cancer | Φ(S,C) |
|---|---|---|
| False | False | 4.5 |
| False | True | 4.5 |
| True | False | 2.7 |
| True | True | 4.5 |

# Converting Bayesian Nets to Markov Nets

$$P(A,B,E,J,M) = P(E)\,P(B)\,P(A|E,B)\,P(J|A)\,P(M|A)$$

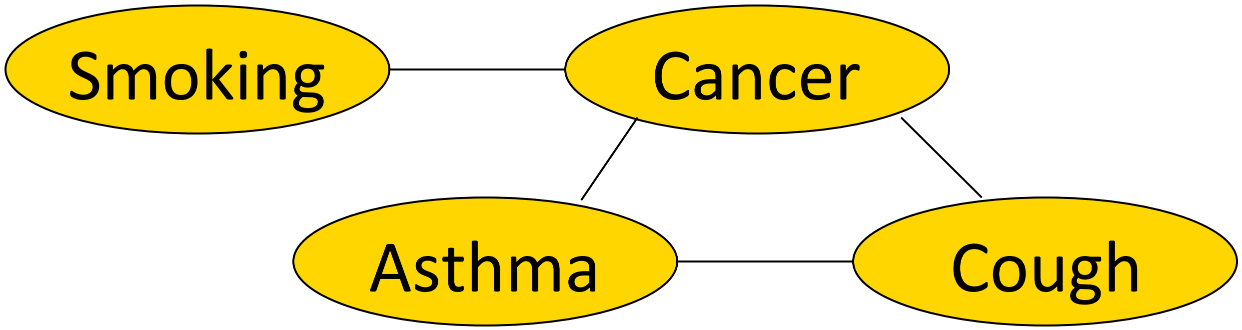
$\phi_E(E)$ $\phi_B(B)$ $\phi_A(A,E,B)$ $\phi_J(A,J)$ $\phi_M(A,M)$



# Markov Networks

- **Undirected** graphical models




- Potential functions defined over cliques

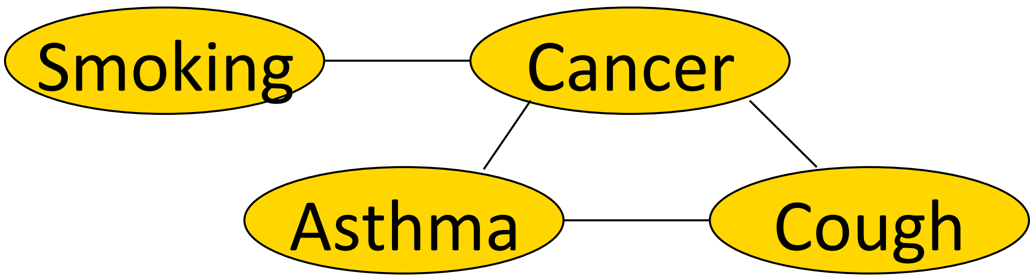
$$P(x) = \frac{1}{Z} \prod_c \Phi_c(x_c)$$

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

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

# Log-linear Markov Networks

- **Undirected** graphical models




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

$$\phi_c(x_c) = \exp(\log \phi_c(x_c))$$
$$= \exp\Big(\sum_j w_j f_j(x_c)\Big)$$

- Log-linear model:

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

$$P(x) = \frac{1}{Z} \prod \phi_c(x_c)$$
$$= \frac{1}{Z} \exp\Big(\sum_i w_i f_i(x)\Big)$$

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

$$w_1 = 1.5$$

# Markov Nets vs. Bayes Nets

| **Property** | **Markov Nets** | **Bayes Nets** |
| --- | --- | --- |
| Form | Prod. potentials | Prod. potentials |
| Potentials | Arbitrary | Cond. probabilities |
| Cycles | Allowed | Forbidden |
| Partition func. | Z = ? | Z = 1 |
| Indep. check | Graph separation | D-separation |

# Inference in Markov Nets

- Want to compute marginals & conditionals of

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

- Exact inference is #P-complete
- Approximate inference
  - Belief propagation
  - Sampling-based methods (e.g., Gibbs sampling)

  - *These work exactly the same as for Bayes nets!*

# Markov Logic: Intuition

- Recall: 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 $\Rightarrow$ Stronger constraint)

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

# 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 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: Friends & Smokers

Smoking causes cancer.

Friends have similar smoking habits.

# Example: Friends & Smokers

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

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

# Example: Friends & Smokers

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

# Example: Friends & Smokers

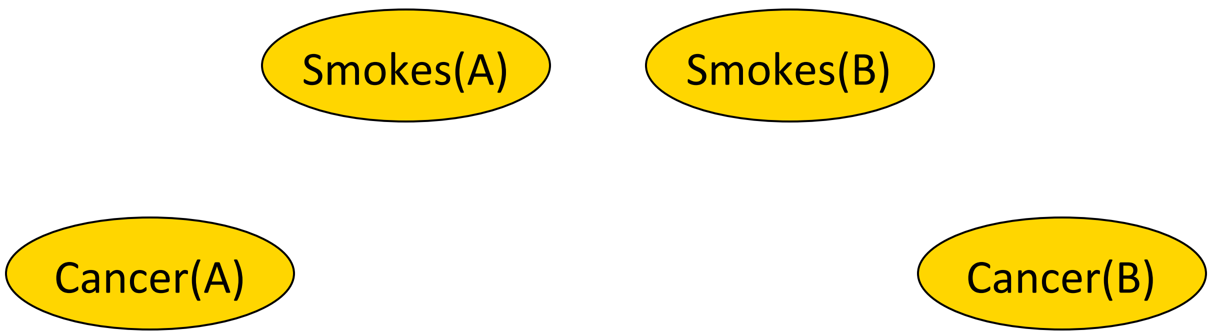
| | |
|---|---|
| 1.5 | $\forall x \; Smokes(x) \Rightarrow Cancer(x)$ |
| 1.1 | $\forall x, y \; Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$ |

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

# Example: Friends & Smokers

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

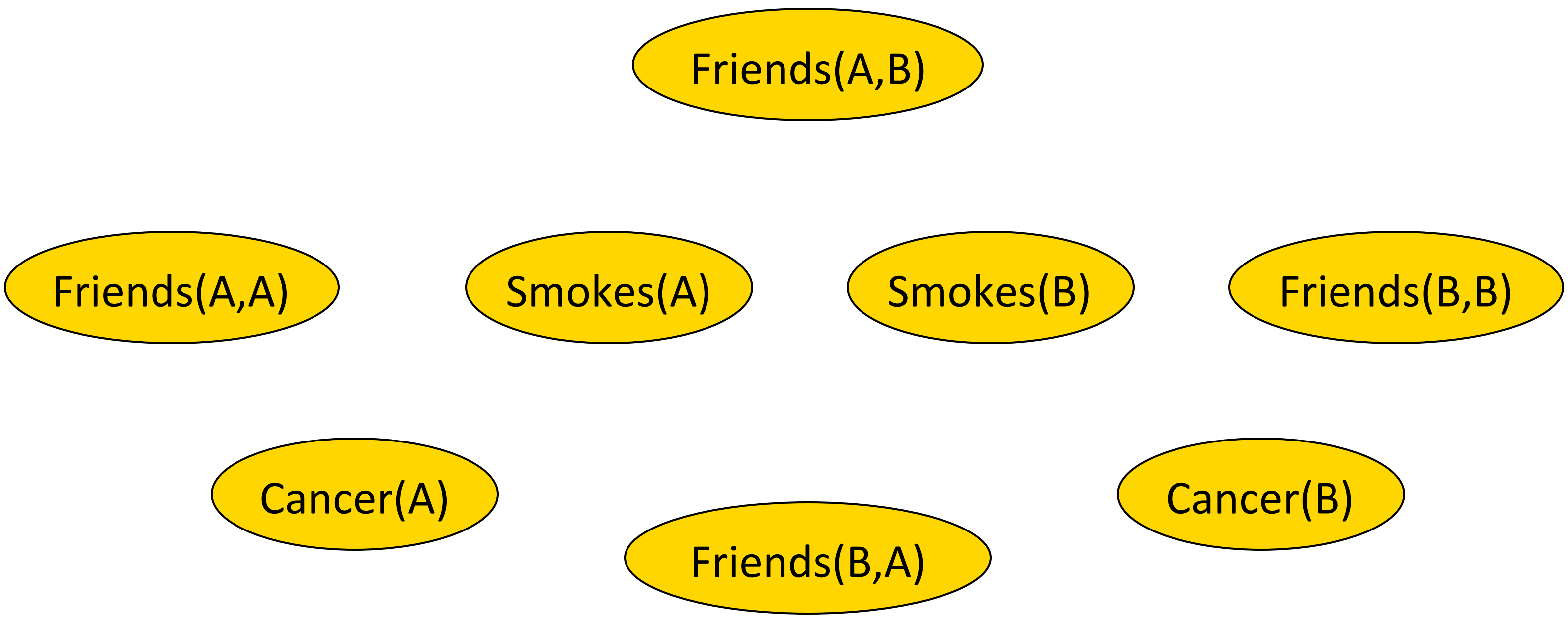
Two constants: **Anna** (A) and **Bob** (B)



# Example: Friends & Smokers

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

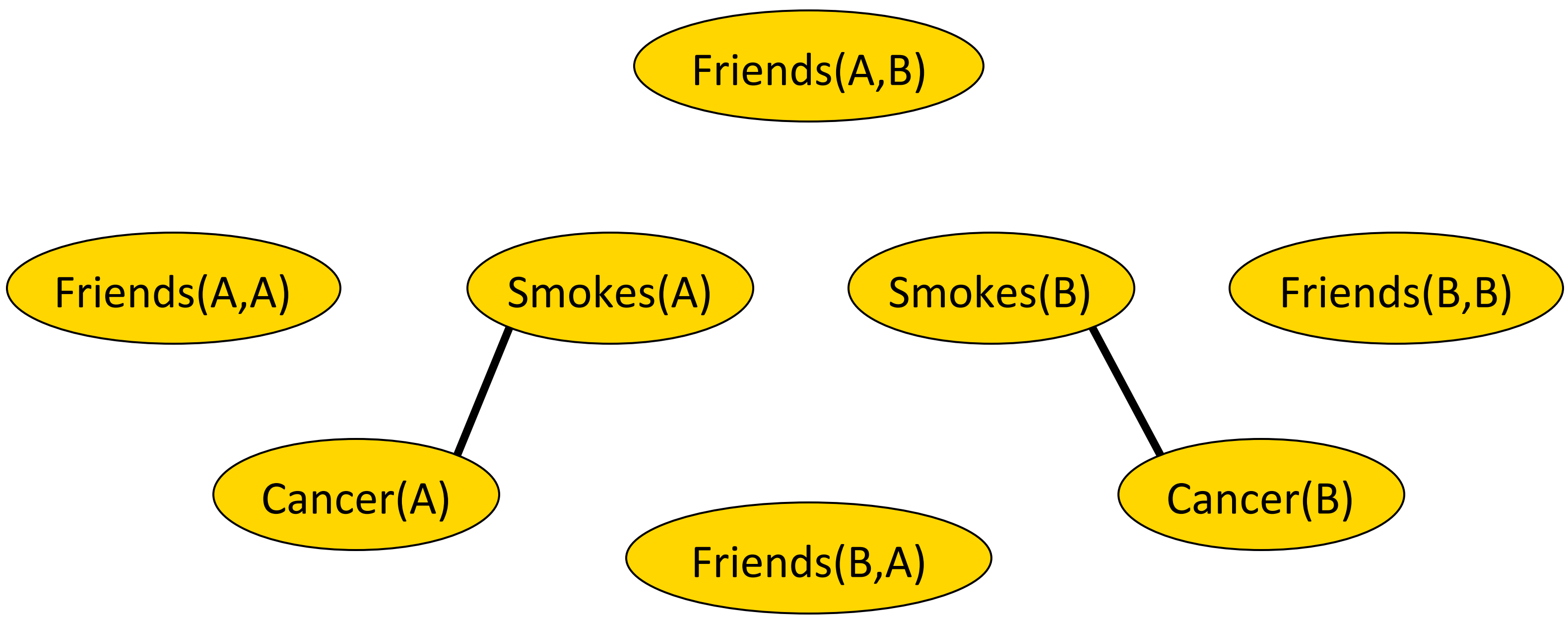
Two constants: **Anna** (A) and **Bob** (B)



# Example: Friends & Smokers

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

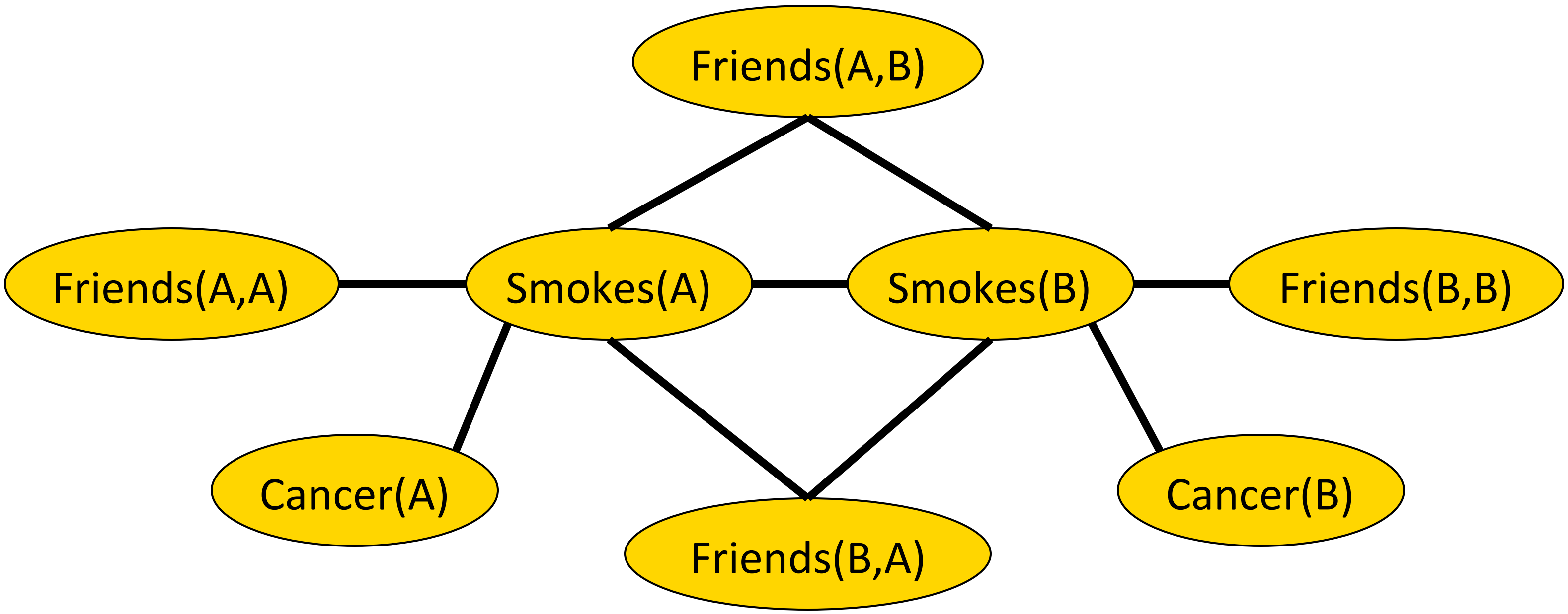
Two constants: **Anna** (A) and **Bob** (B)



# 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)



# Markov Logic Networks

- MLN is **template** for ground Markov nets
- Probability of a world $x$:

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

Weight of formula *i*

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

# Relation to Statistical Models

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

# Relation to First-Order Logic

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

# MAP Inference

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

$$\arg\max_{y} \; P(\underbrace{y}_{\text{Query}} \mid \underbrace{x}_{\text{Evidence}})$$

# 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]

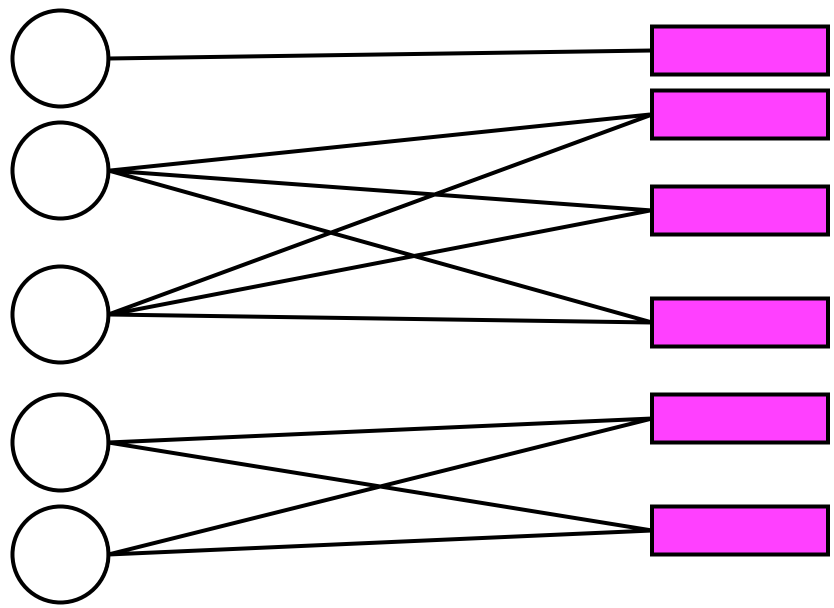
# Lifted Inference

- We can do inference in first-order logic without grounding the KB (e.g.: resolution)
- Can do the same for inference in MLNs
- Group atoms and clauses into “indistinguishable” sets
- Do inference over those
- One example: Lifted belief propagation

# Belief Propagation

$$\mu_{x \to f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \to x}(x)$$

**Nodes (x)**

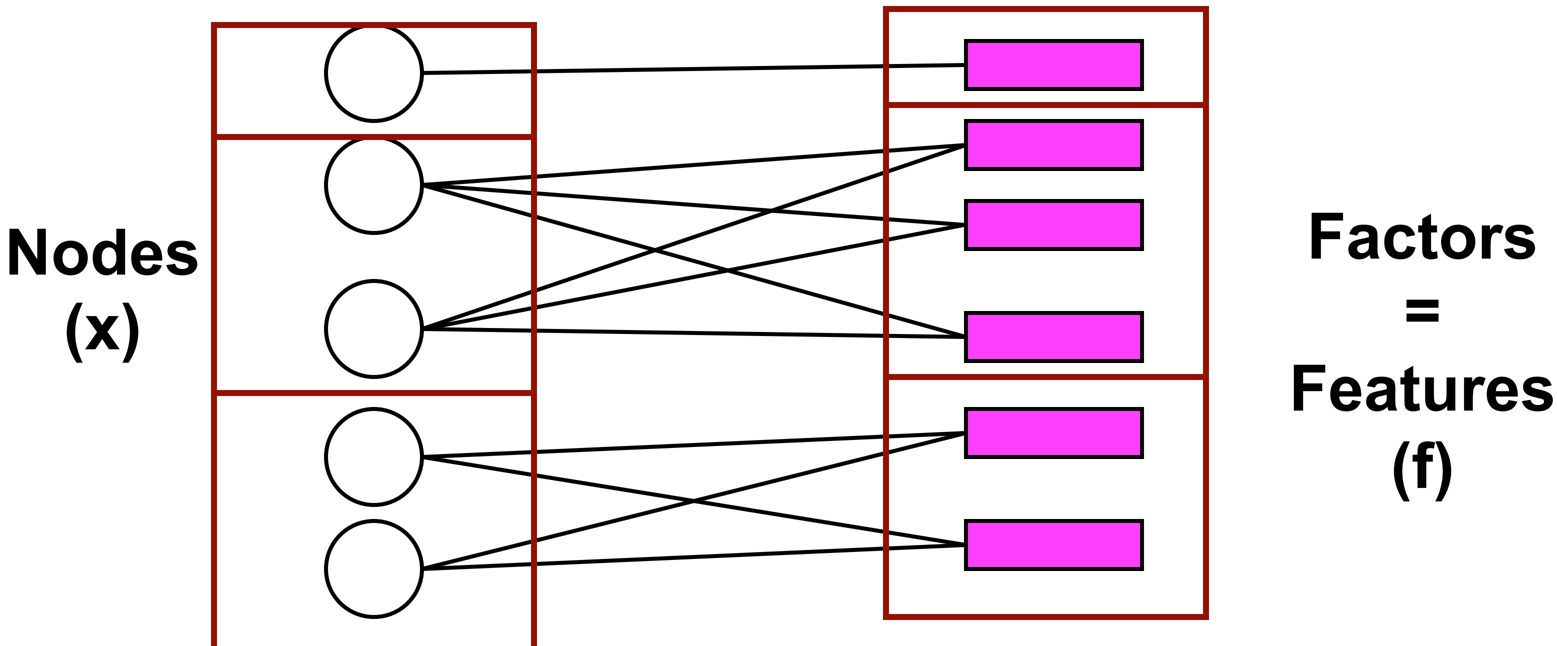


**Factors = Features (f)**

$$\mu_{f \to x}(x) = \sum_{\sim \{x\}} \left( e^{wf(x)} \prod_{y \in n(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$

# Lifted Belief Propagation

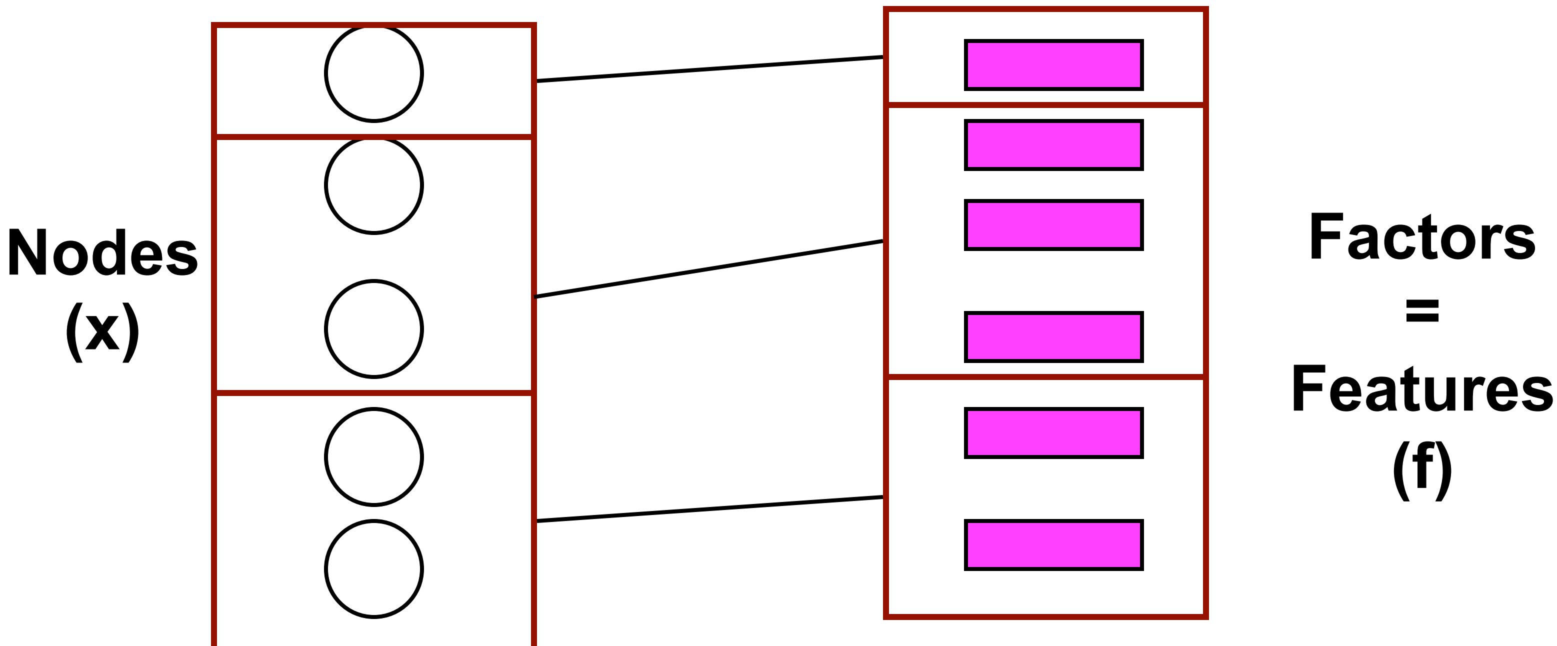
$$\mu_{x\to f}(x) = \prod_{h\in n(x)\setminus\{f\}} \mu_{h\to x}(x)$$




$$\mu_{f\to x}(x) = \sum_{\sim\{x\}}\left(e^{wf(x)} \prod_{y\in n(f)\setminus\{x\}} \mu_{y\to f}(y)\right)$$

# Lifted Belief Propagation

$$\mu_{x \to f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \to x}(x)$$



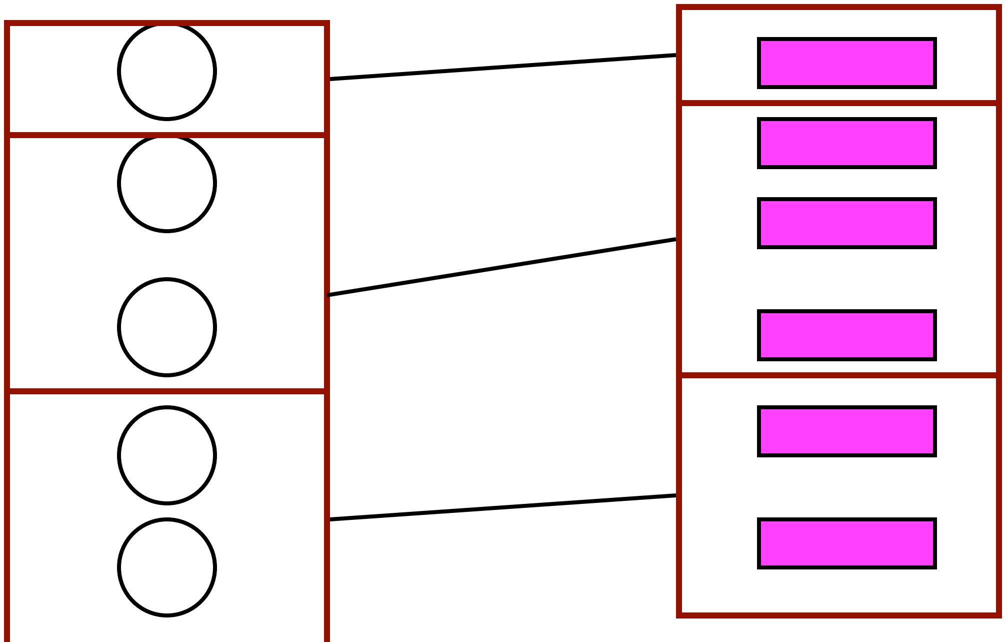

$$\mu_{f \to x}(x) = \sum_{\sim\{x\}} \left( e^{wf(x)} \prod_{y \in n(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$

# Lifted Belief Propagation

**α,β : Functions of edge counts**

$$\mu_{x\to f}(x) = \beta \prod_{h\in n(x)\setminus\{f\}} \mu_{h\to x}(x)^{\alpha}$$

**Nodes (x)**



**Factors = Features (f)**

$$\mu_{f\to x}(x) = \sum_{\sim\{x\}} \left( e^{wf(x)} \prod_{y\in n(f)\setminus\{x\}} \mu_{y\to f}(y) \right)$$

# Lifted Belief Propagation

- Form lifted network composed of supernodes and superfeatures
  - Supernode: Set of ground atoms that all send and receive same messages throughout BP
  - Superfeature: Set of ground clauses that all send and receive same messages throughout BP
- Run belief propagation on lifted network
- Guaranteed to produce same results as ground BP
- Time and memory savings can be huge

# Example

$$\boxed{\forall x, y\ Smokes(x) \wedge Friends(x, y) \Rightarrow Smokes(y)}$$

**Evidence:** Smokes(Ana)
Friends(Bob,Charles), Friends(Charles,Bob)

**N people in the domain**

# Example

$$\forall x, y\ Smokes(x) \wedge Friends(x, y) \Rightarrow Smokes(y)$$

**Evidence:** Smokes(Ana)
Friends(Bob,Charles), Friends(Charles,Bob)

**Intuitive Grouping :**

| | | |
|---|---|---|
| Smokes(Ana) | Smokes(Bob)<br>Smokes(Charles) | Smokes(James)<br>Smokes(Harry)<br>… |

# Alchemy

Open-source software including:

- Full first-order logic syntax
- Inference (MAP and conditional probabilities)
- Weight learning (generative and discriminative)
- Structure learning
- Programming language features

**alchemy.cs.washington.edu**

# Running Alchemy

- Programs
  - Infer
  - Learnwts
  - Learnstruct
- Options
- MLN file
  - Types (optional)
  - Predicates
  - Formulas
- Database files

# Uniform Distribn.: Empty MLN

**Example:** Unbiased coin flips

**Type:** `flip = { 1, … , 20 }`

**Predicate:** `Heads(flip)`

$$P(Heads(f)) = \frac{\frac{1}{Z}e^{0}}{\frac{1}{Z}e^{0} + \frac{1}{Z}e^{0}} = \frac{1}{2}$$

# Multinomial Distrib.: ! Notation

**Example:** Throwing die

**Types:** `throw = { 1, … , 20 }`

`face = { 1, … , 6 }`

**Predicate:** `Outcome(throw,face!)`

**Semantics:** Arguments without "!" determine arguments with "!".

Also makes inference more efficient (triggers blocking).

# Multinomial Distrib.: + Notation

**Example:** Throwing biased die

**Types:** `throw = { 1, … , 20 }`

`face = { 1, … , 6 }`

**Predicate:** `Outcome(throw,face!)`

**Formulas:** `Outcome(t,+f)`

**Semantics:** Learn weight for each grounding of args with "+".

# Text Classification

```
page = { 1, … , n }
word = { … }
topic = { … }

Topic(page,topic!)
HasWord(page,word)

HasWord(p,+w) => Topic(p,+t)
```

# Hypertext Classification

```
Topic(page,topic!)
HasWord(page,word)
Links(page,page)

HasWord(p,+w) => Topic(p,+t)
Topic(p,t) ^ Links(p,p') => Topic(p',t)
```

***Cf.*** S. Chakrabarti, B. Dom & P. Indyk, "Hypertext Classification Using Hyperlinks," in *Proc. SIGMOD-1998*.

# Information Retrieval

```
InQuery(word)
HasWord(page,word)
Relevant(page)

InQuery(w+) ^ HasWord(p,+w) => Relevant(p)
Relevant(p) ^ Links(p,p') => Relevant(p')
```

*Cf.* L. Page, S. Brin, R. Motwani & T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Tech. Rept., Stanford University, 1998.

# Entity Resolution

**Problem: Given database, find duplicate records**

```
HasToken(token,field,record)
SameField(field,record,record)
SameRecord(record,record)

HasToken(+t,+f,r) ^ HasToken(+t,+f,r’)
   => SameField(f,r,r’)
SameField(f,r,r’) => SameRecord(r,r’)
SameRecord(r,r’) ^ SameRecord(r’,r”)
   => SameRecord(r,r”)
```

*Cf.* A. McCallum & B. Wellner, “Conditional Models of Identity Uncertainty with Application to Noun Coreference,” in *Adv. NIPS 17*, 2005.

# Summary

- Probabilistic first order logic combines (most of the) advantages of FOL and probability
- Formulas provide "templates" for probabilistic models (Markov Networks)
- Can learn parameters (and formulas) from data
- Extensions to deal with continuous variables

# You've learned a lot!

Deterministic environments:

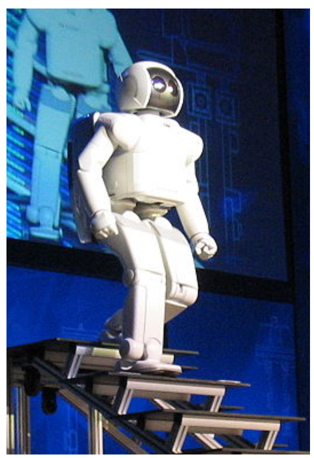
- DFS, BFS, A*, IDS, α-β-search, CSPs, Constraint propagation, Arc consistency, Propositional Logic, Forward/backward chaining, resolution proofs, first order logic, …

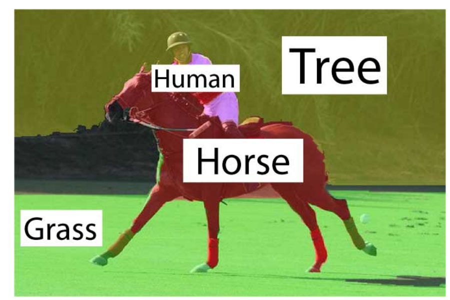
Dealing with uncertainty:

- Bayes Nets, D-separation, Variable elimination, Belief propagation, Gibbs sampling, Value of information, Hidden Markov Models, Kalman Filters, Dynamic Bayesian Networks, Markov Decision processes, value iteration, policy iteration, linear regression, logistic regression, regularization, approximate dynamic programming, Q-learning, Markov Nets, Markov logic networks, …

# Summary

- Build systems (agents) that act rationally
- Act rationally = "perform well on some task"
- Amenable to mathematical analysis, empirical evaluation
- Involves / builds on
  - Logic, optimization, control theory, statistics, game theory, engineering, …





# Acknowledgments

- Slides about Markov Logic Networks from Pedro Domingos (University of Washington)
- http://www.cs.washington.edu/homes/pedrod/803/