Markov Logic

And other SRL Approaches

Overview

- Statistical relational learning
- Markov logic
- Basic inference
- Basic learning

Statistical Relational Learning

Goals:

- Combine (subsets of) logic and probability into a single language
- 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!

Key Dimensions

- **Logical language**
  First-order logic, Horn clauses, frame systems
- **Probabilistic language**
  Bayes nets, Markov nets, PCFGs (Prob. Context-free grammar)
- **Type of learning**
  - Generative / Discriminative
  - Structure / Parameters
  - Knowledge-rich / Knowledge-poor
- **Type of inference**
  - MAP / Marginal
  - Full grounding / Partial grounding / Lifted

Knowledge-Based Model Construction

- **Logical language:** Horn clauses
- **Probabilistic language:** Bayes nets
  - Ground atom → Node
  - Head of clause → Child node
  - Body of clause → Parent nodes
  - >1 clause w/ same head → Combining function
- **Learning:** ILP + EM
- **Inference:** Partial grounding + Belief prop.

Stochastic Logic Programs

- **Logical language:** Horn clauses
- **Probabilistic language:** Probabilistic context-free grammars
  - Attach probabilities to clauses
  - Σ Probs. of clauses w/ same head = 1
- **Learning:** ILP + “Failure-adjusted” EM
- **Inference:** Do all proofs, add probs.

Probabilistic Relational Models

- **Logical language:** Frame systems
- **Probabilistic language:** Bayes nets
  - Bayes net template for each class of objects
  - Object’s attrs. can depend on attrs. of related objs.
  - Only binary relations
  - No dependencies of relations on relations
- **Learning:**
  - Parameters: Closed form (EM if missing data)
  - Structure: “Tiered” Bayes net structure search
- **Inference:** Full grounding + Belief propagation

Relational Markov Networks

- **Logical language:** SQL queries
- **Probabilistic language:** Markov nets
  - SQL queries define cliques
  - Potential function for each query
  - No uncertainty over relations
- **Learning:**
  - Discriminative weight learning
  - No structure learning
- **Inference:** Full grounding + Belief prop.

Bayesian Logic

- **Logical language:** First-order semantics
- **Probabilistic language:** Bayes nets
  - BLOG program specifies how to generate relational world
  - Parameters defined separately in Java functions
  - Allows unknown objects
  - May create Bayes nets with directed cycles
- **Learning:** None to date
- **Inference:**
  - MCMC with user-supplied proposal distribution
  - Partial grounding

Markov Logic

- **Logical language:** First-order logic
- **Probabilistic language:** Markov networks
  - **Syntax:** First-order formulas with weights
  - **Semantics:** Templates for Markov net features
- **Learning:**
  - **Parameters:** Generative or discriminative
  - **Structure:** ILP with arbitrary clauses and MAP score
- **Inference:**
  - **MAP:** Weighted satisfiability
  - **Marginal:** MCMC with moves proposed by SAT solver
  - Partial grounding + Lazy inference / Lifted inference

Markov Logic

- Most developed approach to date
- Many other approaches can be viewed as special cases
- Main focus of rest of this class

Markov Logic: Intuition

- 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

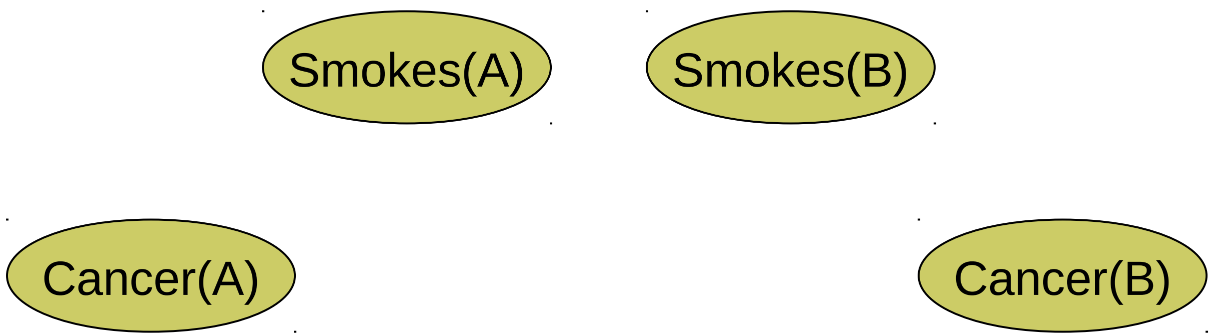
1.5	$\forall x \; Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y \; Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

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

Example: Friends & Smokers

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

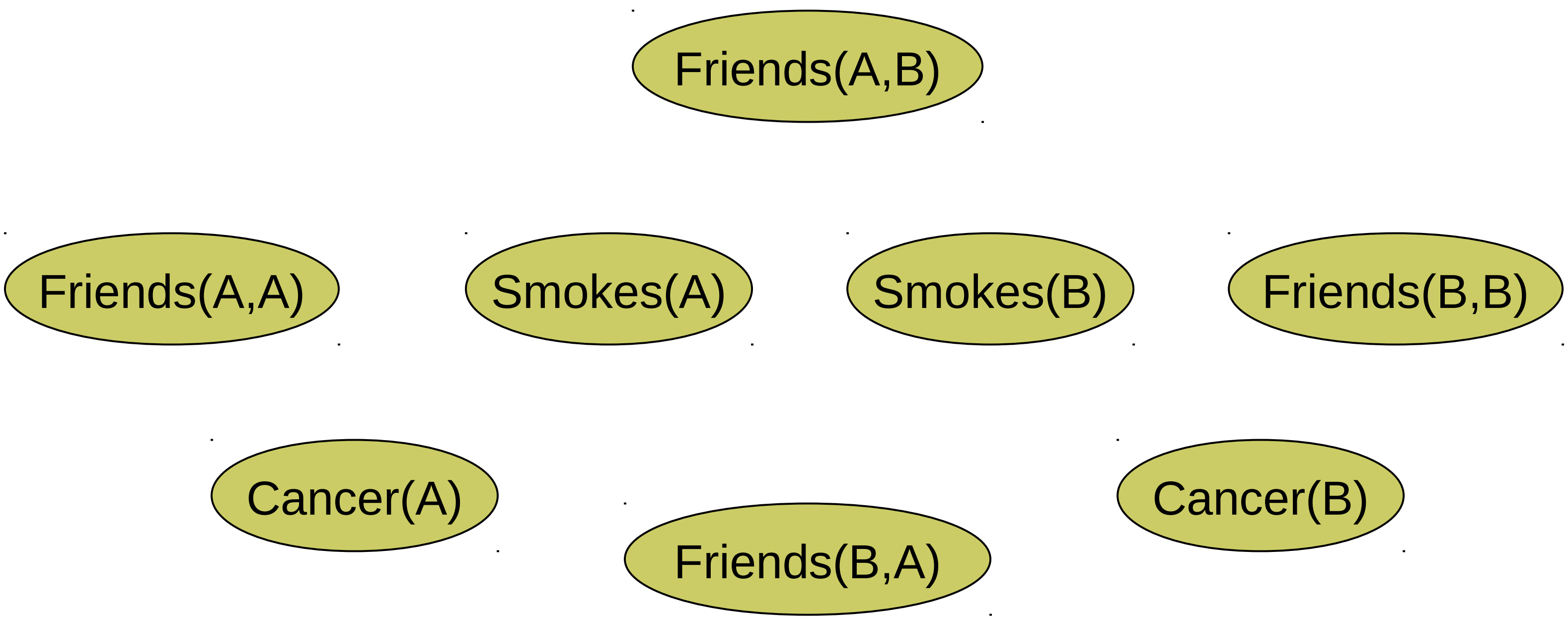
Two constants: **Anna** (A) and **Bob** (B)



Example: Friends & Smokers

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

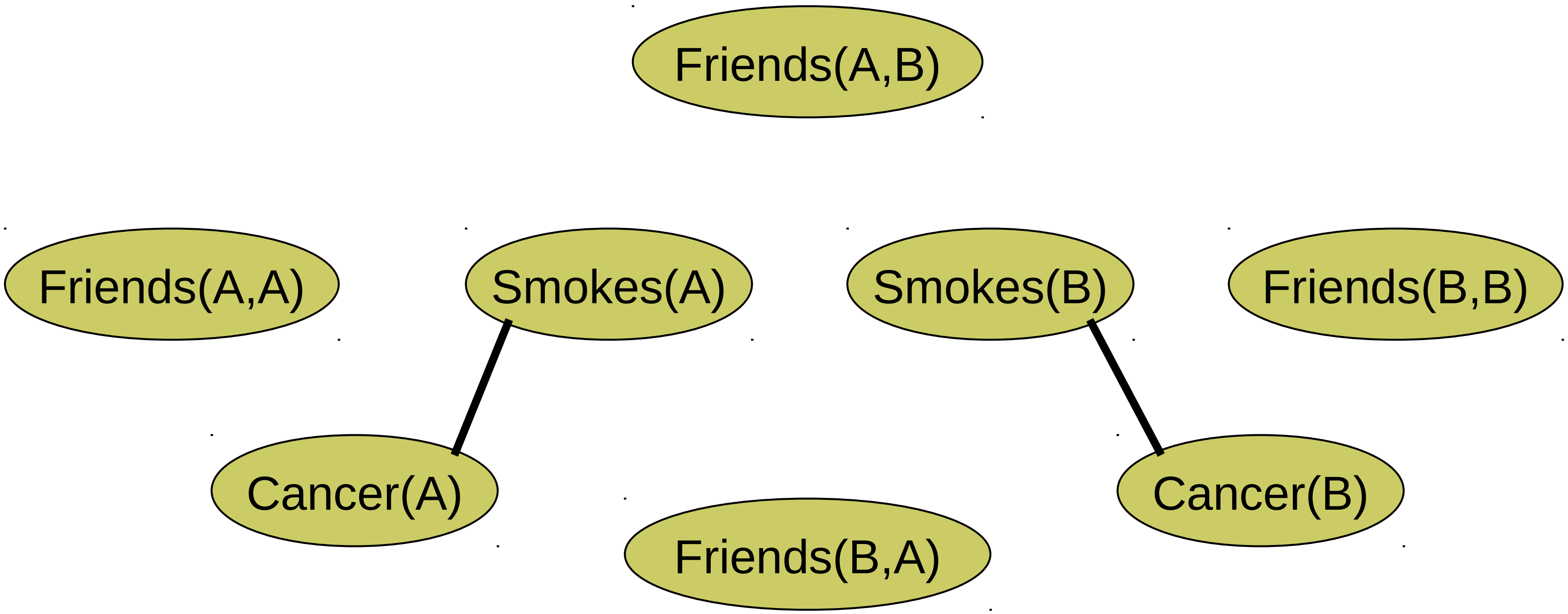
Two constants: **Anna** (A) and **Bob** (B)



Example: Friends & Smokers

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

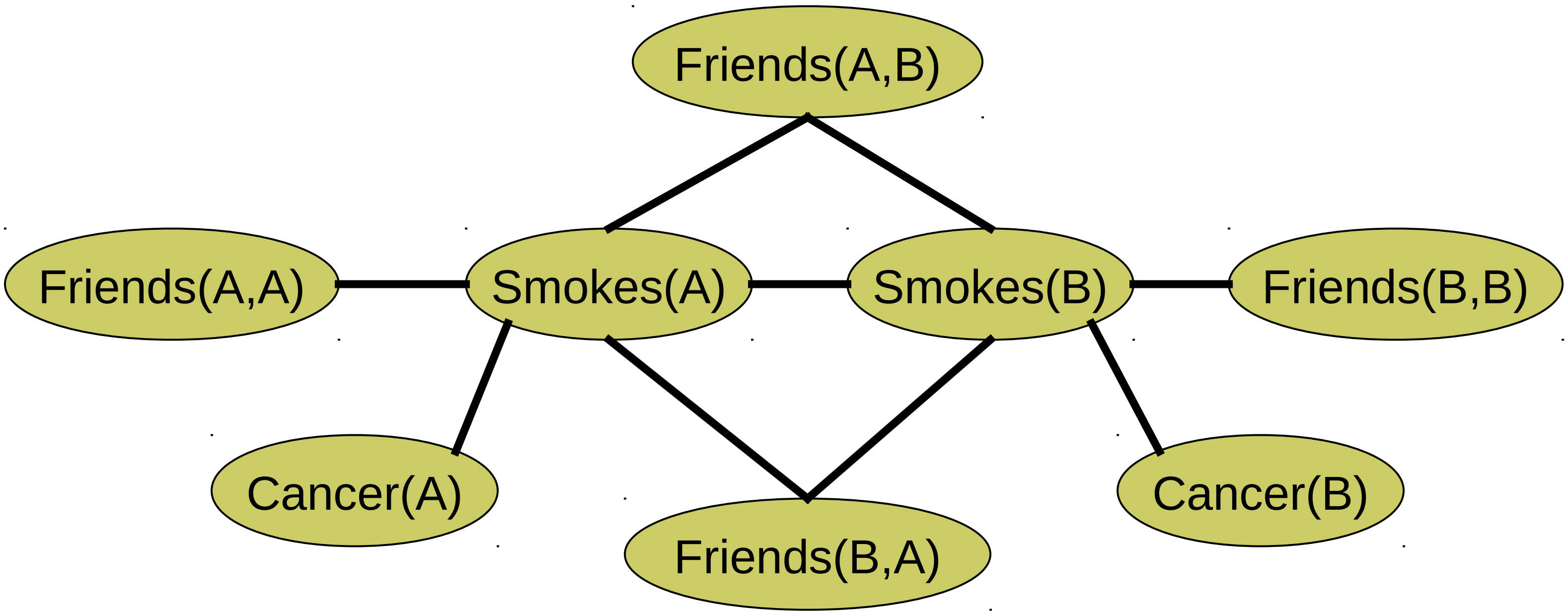
Two constants: **Anna** (A) and **Bob** (B)



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

- **Typed** variables and constants greatly reduce size of ground Markov net
- Functions, existential quantifiers, etc.
- Infinite and continuous domains

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.)

Relation to First-Order Logic

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

MAP Inference

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

$$\operatorname*{argmax}_{y} P(y \mid x)$$

Query

Evidence

MAP Inference

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

$$\underset{y}{\operatorname{argmax}} \ \frac{1}{Z_x} \exp\left(\sum_i w_i n_i(x, y)\right)$$

MAP Inference

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

$$\operatorname*{argmax}_{y} \sum_{i} w_i n_i(x, y)$$

MAP Inference

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

$$\underset{y}{\operatorname{argmax}} \sum_i w_i n_i(x, y)$$

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

The MaxWalkSAT Algorithm

for *i* ← 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

Computing Probabilities

- P(Formula|MLN,C) = ?
- Brute force: Sum probs. of worlds where formula holds
- MCMC: Sample worlds, check formula holds
- P(Formula1|Formula2,MLN,C) = ?
- Discard worlds where Formula 2 does not hold
- In practice: More efficient alternatives

Learning

- Data is a relational database
- For now: Closed world assumption (if not: EM)
- Learning parameters (weights)
  - Similar to learning weights for Markov networks
- Learning structure (formulas)
  - A form of inductive logic programming
  - Also related to learning features for Markov nets

Weight Learning

- Parameter tying: Groundings of same clause

$$\frac{\partial}{\partial w_i} \log P_w(x) = \boxed{n_i(x)} - \boxed{E_w\left[n_i(x)\right]}$$

No. of times clause *i* is true in data

Expected no. times clause *i* is true according to MLN

- Generative learning: Pseudo-likelihood
- Discriminative learning: Cond. likelihood, use MaxWalkSAT for inference

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

	Alchemy	**Prolog**	**BUGS**
Represent-ation	F.O. Logic + Markov nets	Horn clauses	Bayes nets
Inference	Satisfiability, MCMC, BP	Theorem proving	Gibbs sampling
Learning	Parameters & structure	No	Params.
Uncertainty	Yes	No	Yes
Relational	Yes	Yes	No