Markov Logic: Combining Logic and Probability

Parag Singla
Dept. of Computer Science & Engineering
Indian Institute of Technology Delhi

Overview

- Motivation & Background
- Markov logic
- Inference & Learning
- Abductive Plan Recognition

Social Network and Smoking Behavior

Smoking

Cancer

Social Network and Smoking Behavior

Smoking **leads to** **Cancer**

Social Network and Smoking Behavior

Smoking leads to Cancer

Friendship

Similar Smoking Habits

Social Network and Smoking Behavior

Smoking leads to Cancer

Friendship leads to Similar Smoking Habits

Statistical Relational AI

- Real world problems characterized by
  - Entities and Relationships
  - Uncertain Behavior
- Relational Models
  - Horn clauses, SQL queries, first-order logic
- Statistical Models
  - Markov networks, Bayesian networks
- How to combine the two?
- Markov Logic
  - Markov Networks + First Order Logic

Statistical Relational AI

- Probabilistic logic [Nilsson, 1986]
- Statistics and beliefs [Halpern, 1990]
- Knowledge-based model construction [Wellman et al., 1992]
- Stochastic logic programs [Muggleton, 1996]
- Probabilistic relational models [Friedman et al., 1999]
- Bayesian Logic Programs [Kersting and De Raedt 2001]
- Relational Markov networks [Taskar et al., 2002]
- BLOG [Milch et al., 2005]
- **Markov logic** [Richardson & Domingos, 2006]

First-Order Logic

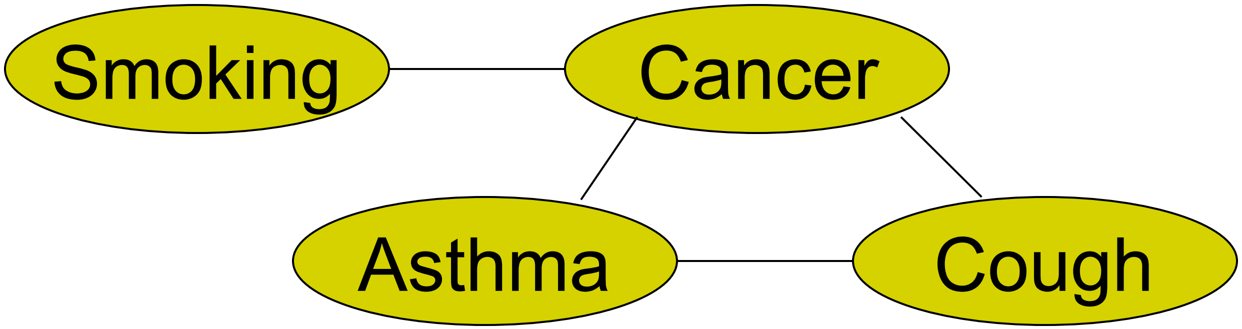
- Constants, variables, functions, predicates
  - Anil, x, MotherOf(x), Friends(x,y)
- Grounding: Replace all variables by constants
  - Friends (Anna, Bob)
- Formula: Predicates connected by operators
  - Smokes(x) $\Rightarrow$ Cancer(x)
- Knowledge Base (KB): A set of formulas
  - Can be equivalently converted into a clausal form
- World: Assignment of truth values to all ground atoms

First-Order Logic

- Deal with finite first-order logic
- Assumptions
  - Unique Names
  - Domain Closure
  - Known Functions

Markov Networks

- **Undirected** graphical models



- Log-linear model:

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

$$f_1(\text{Smoking, Cancer}) = \begin{cases} 1 & \text{if Smoking} \Rightarrow \text{Cancer} \\ 0 & \text{otherweise} \end{cases}$$

$$w_1 = 1.5$$

Overview

- Motivation & Background
- Markov logic
- Inference & Learning
- Abductive Plan Recognition

Markov Logic
[Richardson & Domingos 06]

- 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)

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

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 finite 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

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

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

Example: Friends & Smokers

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

Example: Friends & Smokers

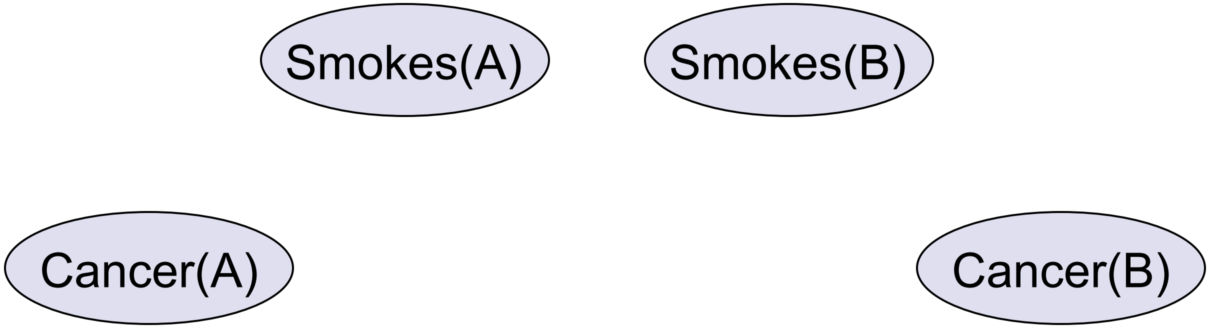
1.5	$\forall x \; Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y \; Friends(x, y) \wedge Smokes(x) \Rightarrow Smokes(y)$

Two constants: **Anil** (A) and **Bunty** (B)

Example: Friends & Smokers

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

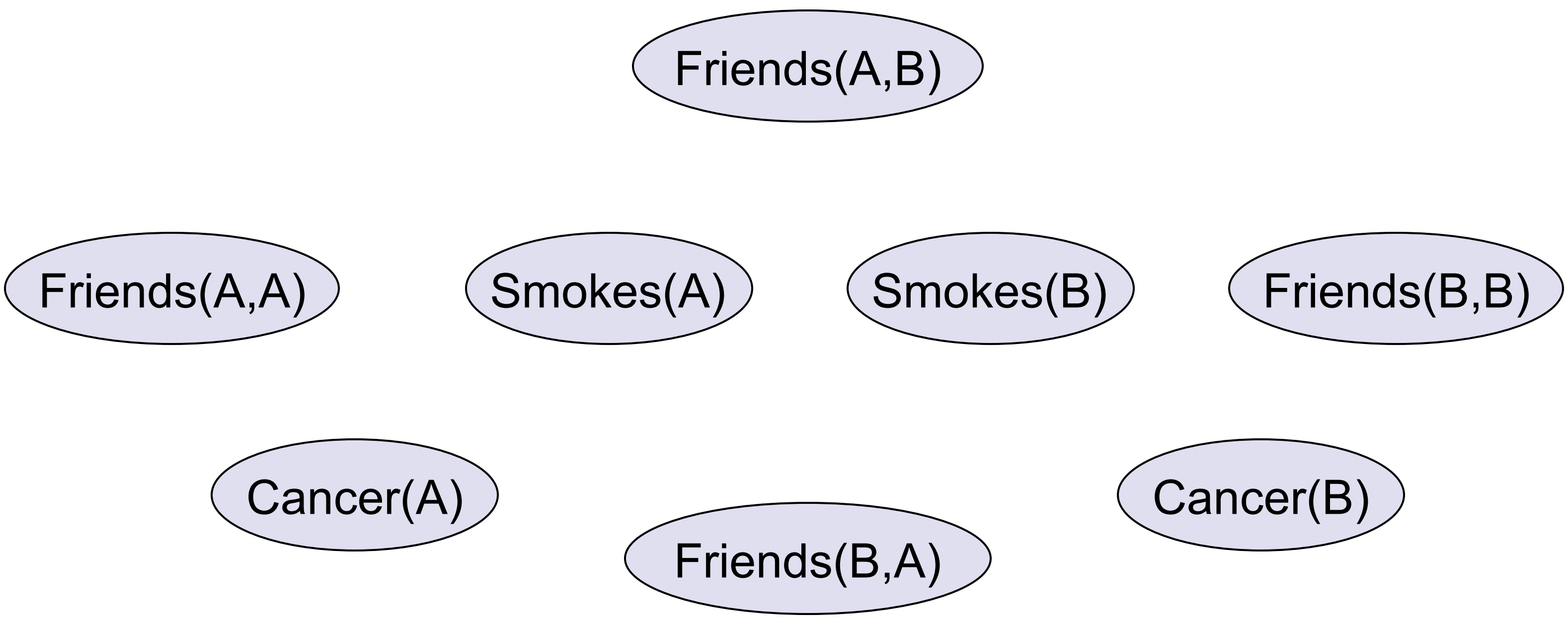
Two constants: **Anil** (A) and **Bunty** (B)



Example: Friends & Smokers

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

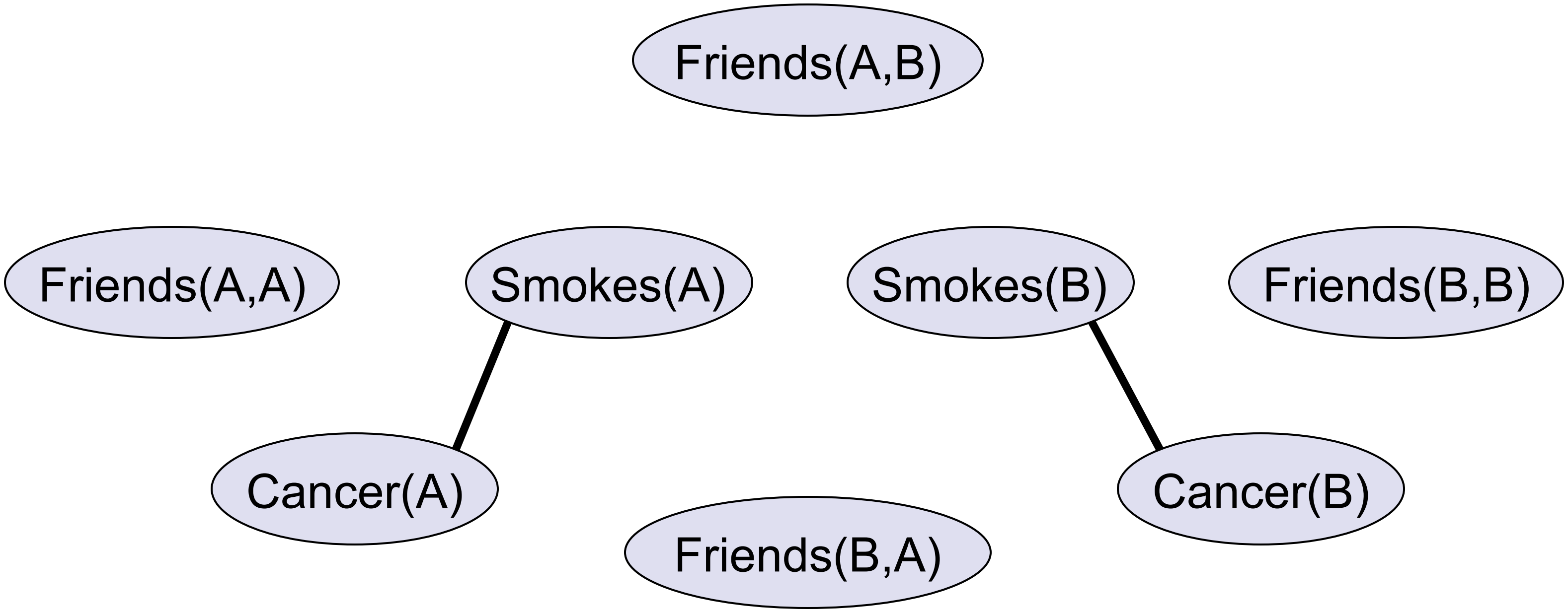
Two constants: **Anil** (A) and **Bunty** (B)



Example: Friends & Smokers

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

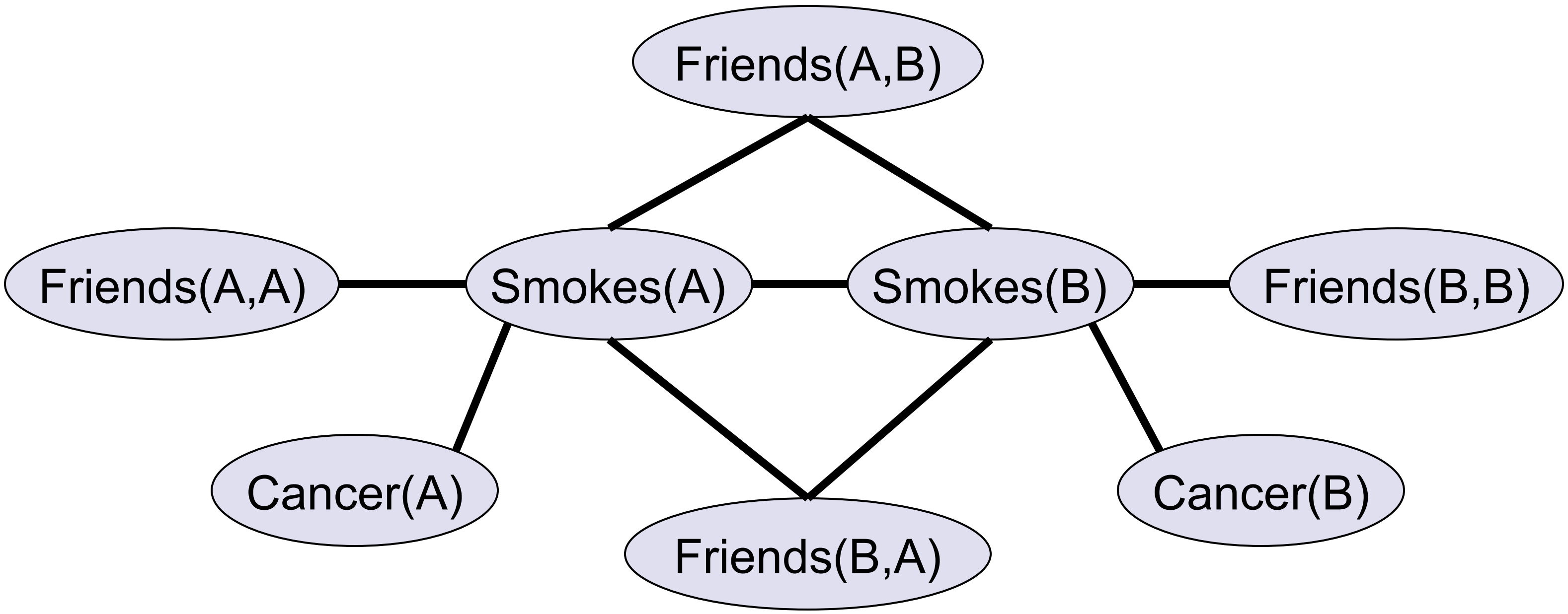
Two constants: **Anil** (A) and **Bunty** (B)



Example: Friends & Smokers

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

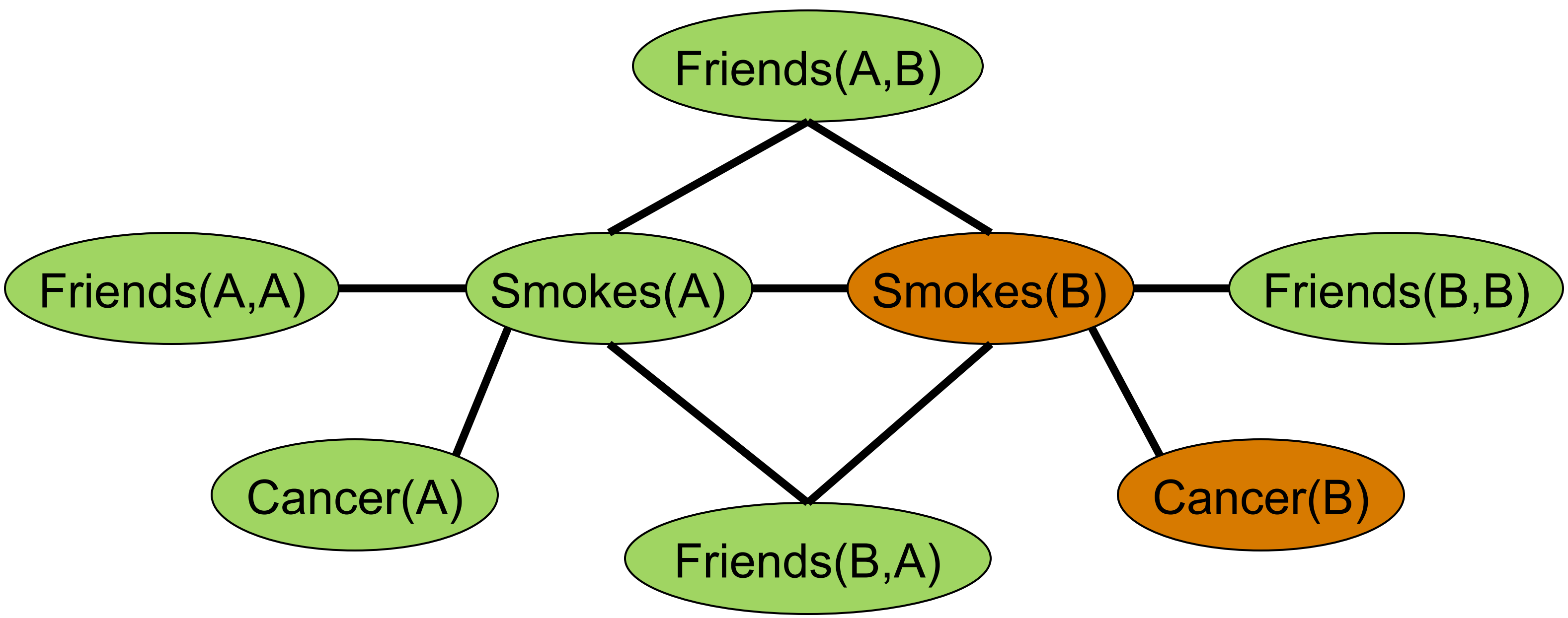
Two constants: **Anil** (A) and **Bunty** (B)



Example: Friends & Smokers

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

Two constants: **Anil** (A) and **Bunty** (B)



State of the World ≡ {0,1} Assignment to the nodes

Markov Logic Networks

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

$$P(x) = \frac{1}{Z}\exp\left(\sum_{k \in ground\ formulas} w_k f_k(x)\right)$$

Markov Logic Networks

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

$$P(x) = \frac{1}{Z} \exp\left(\sum_{k \in ground\ formulas} w_k f_k(x) \right)$$

$$P(x) = \frac{1}{Z} \exp\left(\sum_{i \in \mathrm{MLN\ formulas}} \boxed{w_i} \boxed{n_i(x)} \right)$$

Weight of formula *i*

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

Relation to Statistical Models

- Special cases:
  - Markov networks
  - Markov random fields
  - Bayesian networks
  - Log-linear models
  - Exponential models
  - Logistic regression
  - Hidden Markov models
  - Conditional random fields
- Obtained by making all predicates zero-arity

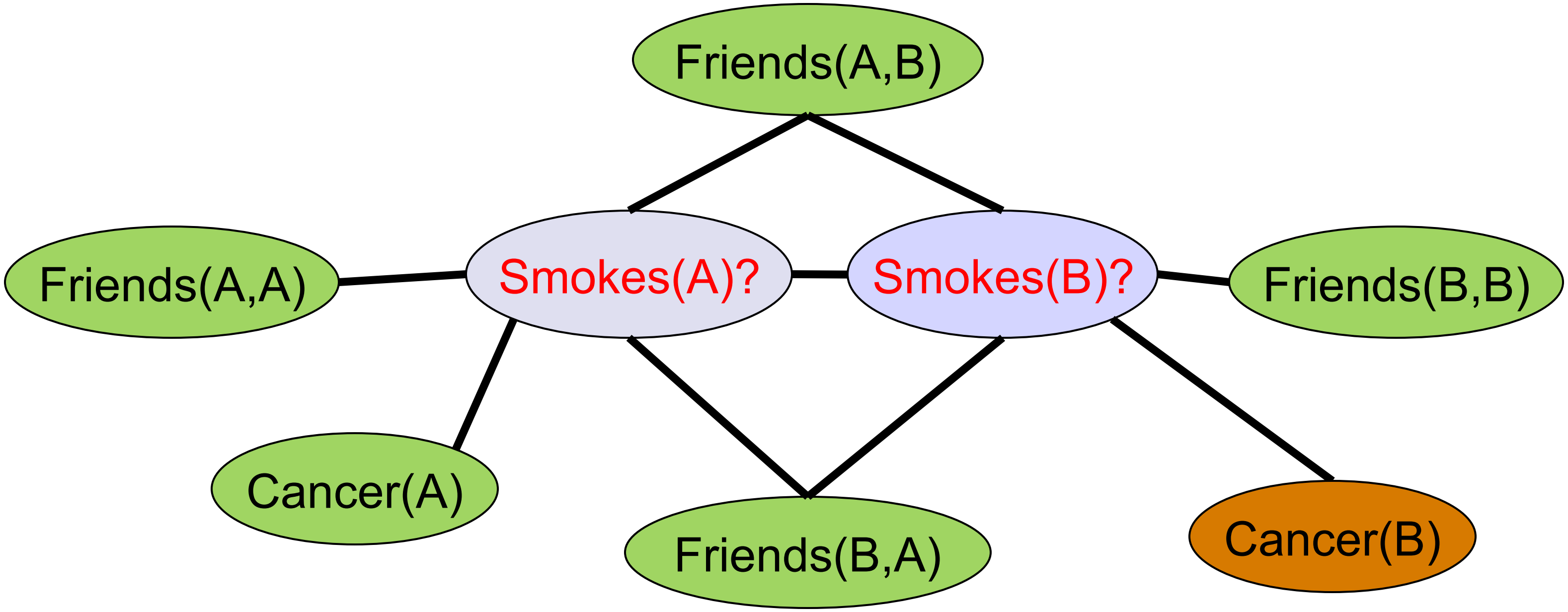
Relation to First-Order Logic

- Infinite weights $\Rightarrow$ First-order logic
- Satisfiable KB, positive weights $\Rightarrow$ Satisfying assignments = Modes of distribution
- Markov logic allows contradictions between formulas
- Relaxing Assumptions
  - Known Functions (Markov Logic in Infinite Domains) **[Singla & Domingos 07]**
  - Unique Names (Entity Resolution with Markov Logic) **[Singla & Domingos 06]**

Overview

- Motivation & Background
- Markov logic
- Inference & Learning
- Abductive Plan Recognition

Inference



blue ? – non-evidence (unknown)
green/orange – evidence (known)

MPE Inference

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

$$P(y \mid x) = \frac{1}{Z_x} \exp\left(\sum_i w_i n_i(x, y) \right)$$

Query

Evidence

MPE Inference

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

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

MPE Inference

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

$$\arg\max_{y} \sum_{i} w_i n_i(x, y)$$

MPE 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. 97])

Lazy Grounding of Clauses: LazySAT [Singla & Domingos 06]

Marginal Inference

- **Problem:** Find the probability of query atoms given evidence

$$P(y \mid x) = \frac{1}{Z_x} \exp\left(\sum_i w_i n_i(x, y) \right)$$

Query

Evidence

Marginal Inference

- **Problem:** Find the probability of query atoms given evidence

$$P(\underbrace{y}_{\text{Query}} \mid \underbrace{x}_{\text{Evidence}}) = \frac{1}{Z_x} \exp\left(\sum_i w_i n_i(x, y) \right)$$

Query

Evidence

Computing Z_x takes exponential time!

Marginal Inference

- **Problem:** Find the probability of query atoms given evidence

$$P(y \mid x) = \frac{1}{Z_x} \exp\left(\sum_i w_i n_i(x, y) \right)$$

Query

Evidence

Approximate Inference: Gibbs Sampling, Message Passing
[Richardson & Domingos 06, Poon & Domingos 06, Singla & Domingos 08]

Learning Parameters

$w_1?$	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
$w_2?$	$\forall x, y\ Friends(x, y) \wedge Smokes(x) \Rightarrow Smokes(y)$

Three constants: **Anil, Bunty, Chaya**

Learning Parameters

$w_1?$	$\forall x \; Smokes(x) \Rightarrow Cancer(x)$
$w_2?$	$\forall x, y \; Friends(x, y) \wedge Smokes(x) \Rightarrow Smokes(y)$

Three constants: **Anil, Bunty, Chaya**

Smokes
Smokes(Anil)
Smokes(Bunty)

Cancer
Cancer(Anil)
Cancer(Bunty)

Friends
Friends(Anil, Bunty)
Friends(Bunty, Anil)
Friends(Anil, Chaya)
Friends(Chaya, Anil)

Closed World Assumption:
Anything not in the database is assumed false.

Learning Parameters

$w_1?$	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
$w_2?$	$\forall x, y\ Friends(x, y) \wedge Smokes(x) \Rightarrow Smokes(y)$

Three constants: **Anil, Bunty, Chaya**

Smokes
Smokes(Anil)
Smokes(Bunty)

Cancer
Cancer(Anil)
Cancer(Bunty)

Friends
Friends(Anil, Bunty)
Friends(Bunty, Anil)
Friends(Anil, Chaya)
Friends(Chaya, Anil)

Maximize the Likelihood: Use Gradient Based Approaches
[Singla & Domingos 05, Lowd & Domingos 07]

Learning Structure

Three constants: **Anil, Bunty, Chaya**

Smokes
Smokes(Anil)
Smokes(Bunty)

Cancer
Cancer(Anil)
Cancer(Bunty)

Friends
Friends(Anil, Bob)
Friends(Bunty, Anil)
Friends(Anil, Chaya)
Friends(Chaya, Anil)

Can we learn the set of the formulas in the MLN?

Learning Structure

$w_1?$	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
$w_2?$	$\forall x, y\ Friends(x, y) \wedge Smokes(x) \Rightarrow Smokes(y)$

Three constants: **Anil, Bunty, Chaya**

Smokes
Smokes(Anil)
Smokes(Bunty)

Cancer
Cancer(Anil)
Cancer(Bunty)

Friends
Friends(Anil, Bob)
Friends(Bunty, Anil)
Friends(Anil, Chaya)
Friends(Chaya, Anil)

Can we refine the set of the formulas in the MLN?

Learning Structure

$w_1?$	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
$w_2?$	$\forall x, y\ Friends(x, y) \land Smokes(x) \Rightarrow Smokes(y)$
$w_3?$	$\forall x, y\ Friends(x, y) \Rightarrow Friends(y, x)$

Three constants: **Anil, Bunty, Chaya**

Smokes
Smokes(Anil)
Smokes(Bunty)

Cancer
Cancer(Anil)
Cancer(Bunty)

Friends
Friends(Anil, Bob)
Friends(Bunty, Anil)
Friends(Anil, Chaya)
Friends(Chaya, Anil)

Can we refine the set of the formulas in the MLN?

Learning Structure

$w_1?$	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
$w_2?$	$\forall x, y\ Friends(x,y) \wedge Smokes(x) \Rightarrow Smokes(y)$
$w_3?$	$\forall x, y\ Friends(x,y) \Rightarrow Friends(y,x)$

Three constants: **Anil, Bunty, Chaya**

Smokes
Smokes(Anil)
Smokes(Bunty)

Cancer
Cancer(Anil)
Cancer(Bunty)

Friends
Friends(Anil, Bob)
Friends(Bunty, Anil)
Friends(Anil, Chaya)
Friends(Chaya, Anil)

ILP style search for formuals
[Kok & Domingos 05, 07, 09, 10]

Alchemy

Open-source software including:

- Full first-order logic syntax
- Inference algorithms
- Parameter & structure learning algorithms

alchemy.cs.washington.edu

Overview

- Motivation & Background
- Markov logic
- Inference & Learning
- Abductive Plan Recognition

Applications

- Web-mining
- Collective Classification
- Link Prediction
- Information retrieval
- Entity resolution
- Activity Recognition
- Image Segmentation & De-noising
- Social Network Analysis
- Computational Biology
- Natural Language Processing
- Robot mapping
- Abductive Plan Recognition
- More..

Abduction

- **Abduction:** Given the observations and the background, find the best explanation
- Given:
  - Background knowledge (B)
  - A set of observations (O)
- To Find:
  - A hypothesis, H, a set of assumptions
- $B \cup H \not\models \bot$, $B \cup H \models O$

Plan Recognition

- Given planning knowledge and a set of low-level actions, identify the top level plan
- Involves abductive reasoning

B: Planning Knowledge (Background)

O: Set of low-level Actions (Observations)

H: Top Level Plan (Hypothesis)

$B \cup H \neq \perp$, $B \cup H \models O$

Plan Recognition Example

- Emergency Response Domain [Blaylock & Allen 05]
- Background Knowledge

  heavy_snow(loc) ∧ drive_hazard(loc) ⇒ block_road(loc)

  accident(loc) ∧ clear_wreck(crew,loc) ⇒ block_road(loc)
- Observation

  block_road(Plaza)
- Possible Explanations
  - Heavy Snow?
  - Accident?

Abduction using Markov logic

- Given

  heavy_snow(loc) ∧ drive_hazard(loc) ⇒ block_road(loc)

  accdent(loc) ∧ clear_wreck(crew, loc) ⇒ block_road(loc)

  Observation: block_road(plaza)

Abduction using Markov logic

- Given

  heavy_snow(loc) $\wedge$ drive_hazard(loc) $\Rightarrow$ block_road(loc)

  accdent(loc) $\wedge$ clear_wreck(crew, loc) $\Rightarrow$ block_road(loc)

  Observation: block_road(plaza)

  Does not work!

- Rules are true independent of antecedents
- Need to go from effect to cause

Introducing Hidden Cause

heavy_snow(loc) ∧ drive_hazard(loc) ⇒ block_road(loc)

rb_C1(loc) → Hidden Cause

heavy_snow(loc) ∧ drive_hazard(loc) ⇔ rb_C1(loc)

Introducing Hidden Cause

heavy_snow(loc) ∧ drive_hazard(loc) ⇒ block_road(loc)

rb_C1(loc) → Hidden Cause

heavy_snow(loc) ∧ drive_hazard(loc) ⇔ rb_C1(loc)

rb_C1(loc) ⇒ block_road(loc)

Introducing Hidden Cause

heavy_snow(loc) ∧ drive_hazard(loc) ⇒ block_road(loc)

rb_C1(loc) → Hidden Cause

heavy_snow(loc) ∧ drive_hazard(loc) ⇔ rb_C1(loc)

rb_C1(loc) ⇒ block_road(loc)

accident(loc) ∧ clear_wreck(loc, crew) ⇒ block_road(loc)

rb_C2(loc, crew)

accident(loc) ∧ clear_wreck(loc) ⇔ rb_C2(loc, crew)

rb_C2(loc,crew) ⇒ block_road(loc)

Introducing Reverse Implication

Explanation 1: heavy_snow(loc) ∧ clear_wreck(loc) ⇔ rb_C1(loc)

Explanation 2: accident(loc) ∧ clear_wreck(loc) ⇔ rb_C2(loc, crew)

Multiple causes combined via reverse implication

Existential quantification

block_road(loc) ⇒ rb_C1(loc) v (∃ crew rb_C2(loc, crew))

Low-Prior on Hidden Causes

Explanation 1: heavy_snow(loc) ∧ clear_wreck(loc) ⇔ rb_C1(loc)

Explanation 2: accident(loc) ∧ clear_wreck(loc) ⇔ rb_C2(loc, crew)

Multiple causes combined via reverse implication

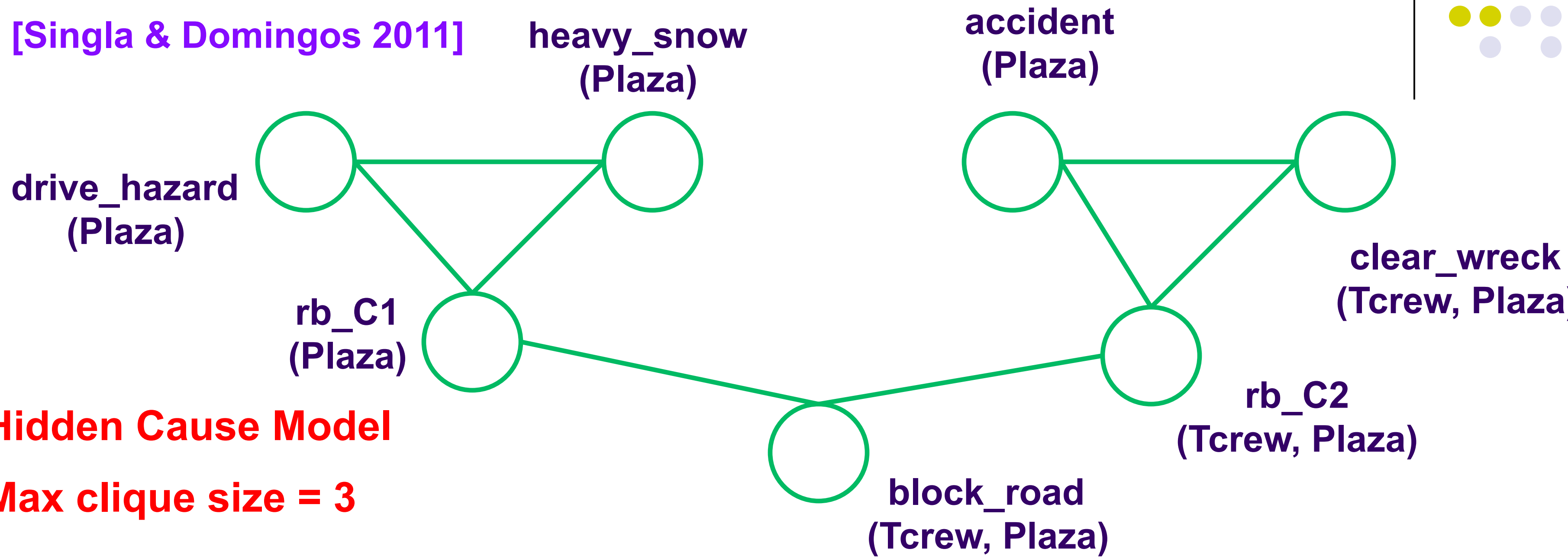
Existential quantification

block_road(loc) ⇒ rb_C1(loc) v (∃ crew rb_C2(loc, crew))

-w1 rb_C1(loc)
-w2 rb_C2(loc, crew)

Hidden Causes: Avoiding Blow-up

[Singla & Domingos 2011]

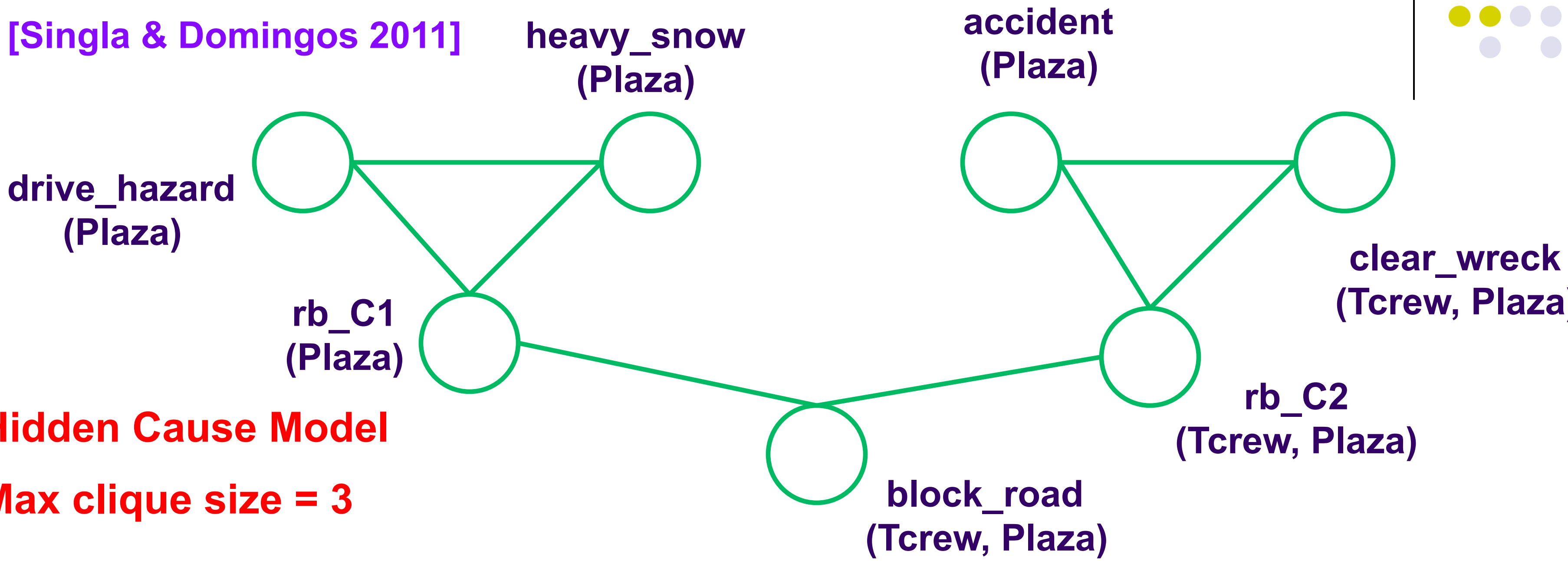


Hidden Cause Model

Max clique size = 3

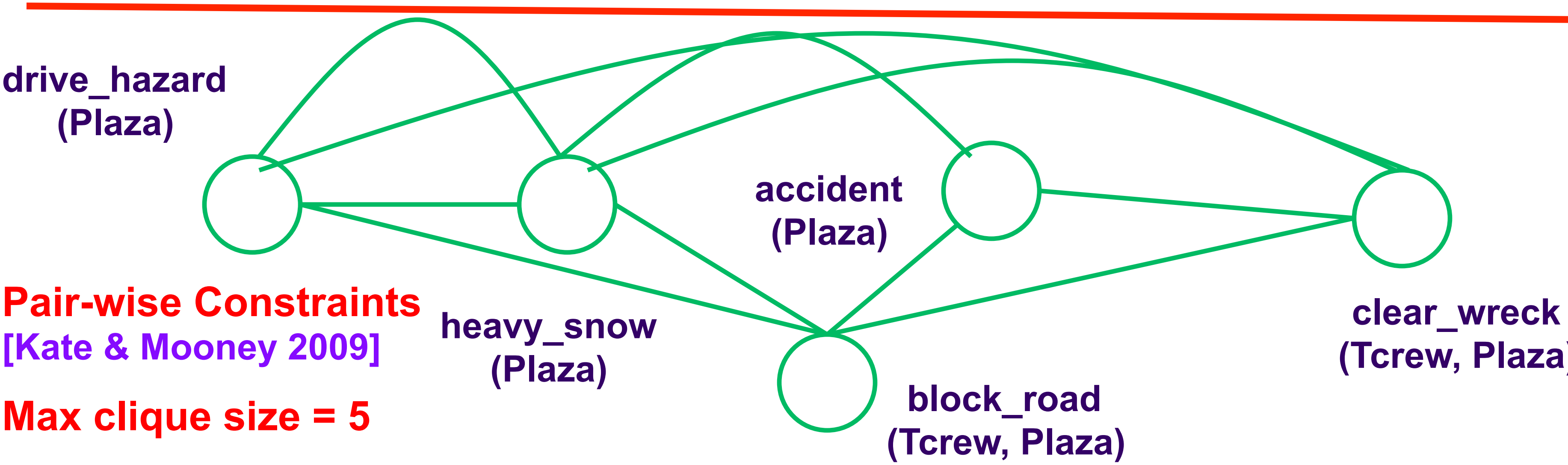
Hidden Causes: Avoiding Blow-up

[Singla & Domingos 2011]



Hidden Cause Model

Max clique size = 3



Pair-wise Constraints

[Kate & Mooney 2009]

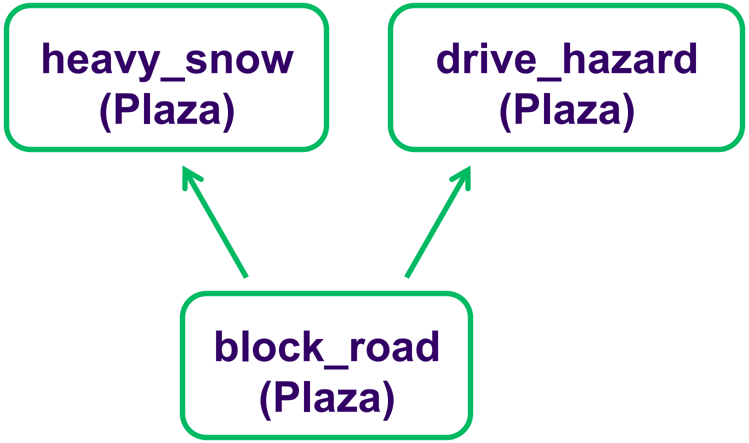
Max clique size = 5

Second Issue: Ground Network Too Big!

- Grounding out the full network may be costly
- Many irrelevant nodes/clauses are created
- Complicates learning/inference
- Can focus the grounding (KBMC)

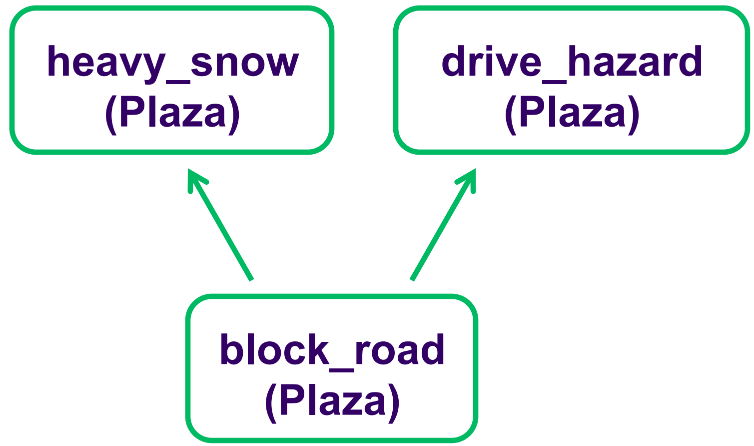
Abductive Model Construction

Observation:
block_road(Plaza)

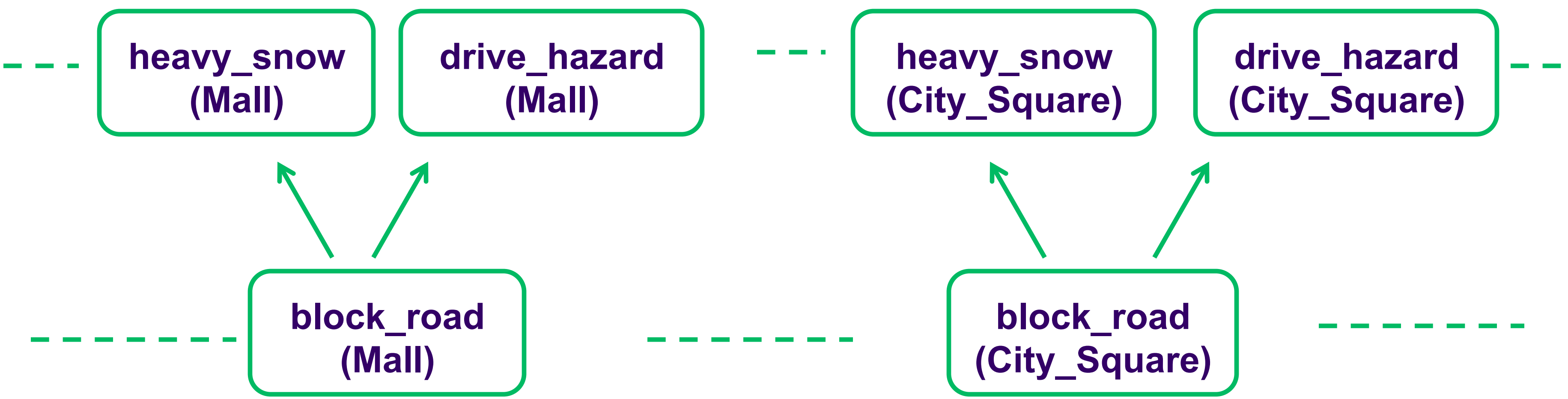


Abductive Model Construction

Observation:
block_road(Plaza)



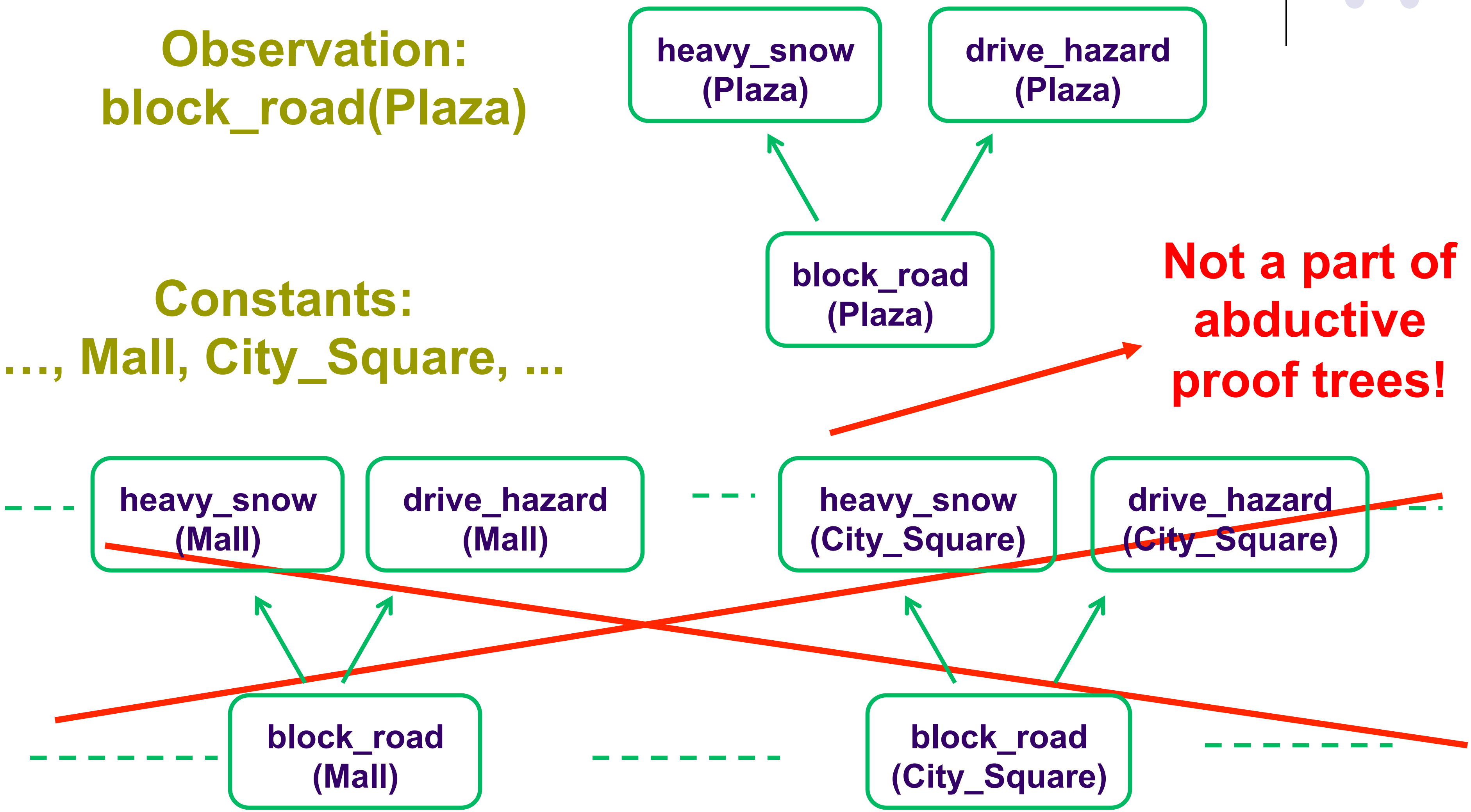
Constants:
…, Mall, City_Square, ...



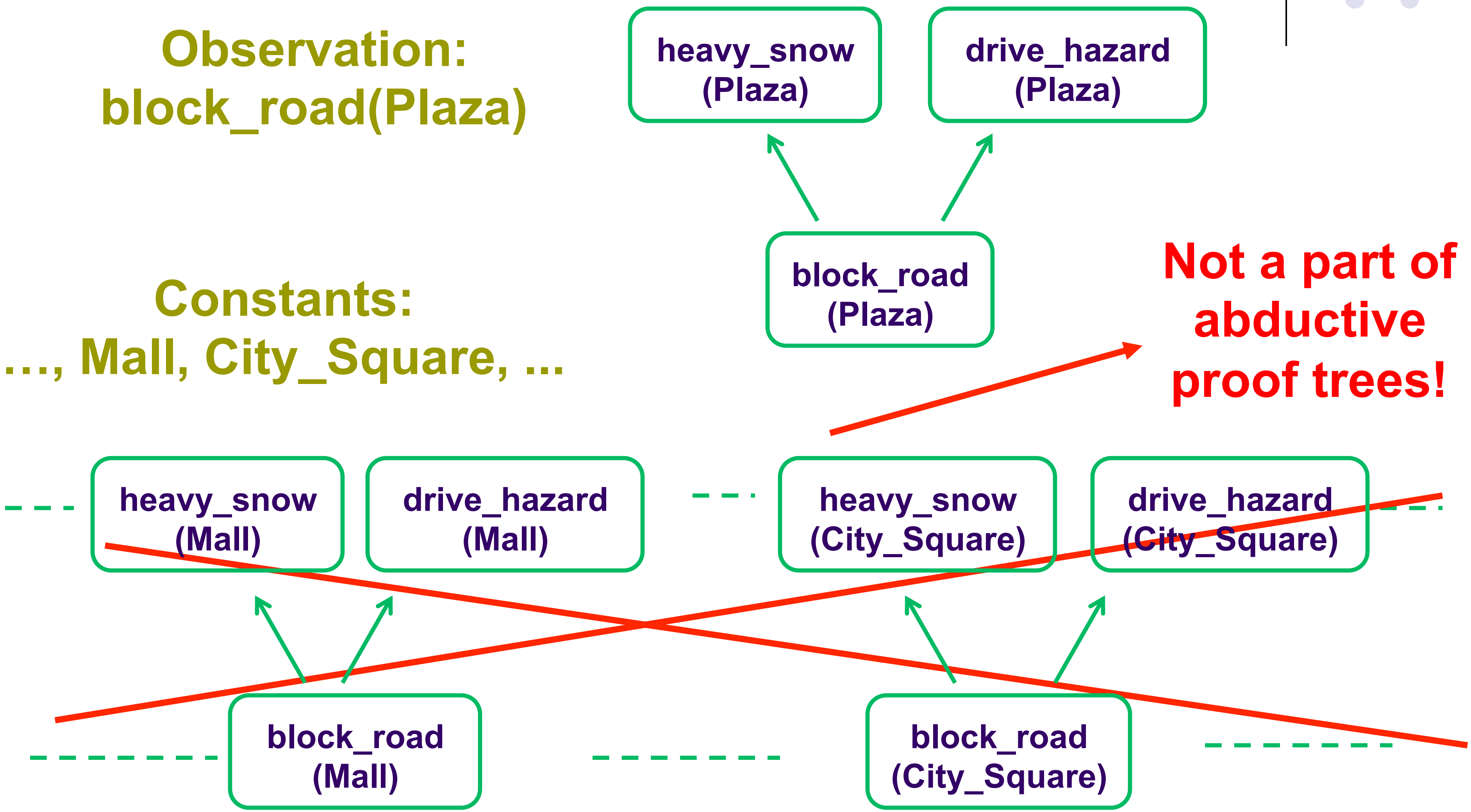
Abductive Model Construction

Observation:
block_road(Plaza)

Constants:
…, Mall, City_Square, ...



Abductive Model Construction



Backward chaining to get proof trees [Stickel 1988]

Abductive Markov Logic
[Singla & Domingos 11]

- Re-encode the MLN rules
  - Introduce reverse implications
- Construct ground Markov network
  - Use abductive model construction
- Perform learning and inference

Summary

- Real world applications
  - Entities and Relations
  - Uncertainty
- Unifying logical and statistical AI
- Markov Logic – simple and powerful model
- Need to do to efficient learning and inference
- Applications: Abductive Plan Recognition