Markov Logic Networks

Matthew Richardson, Pedro Domingos, UW-Seattle

Yue Gao, Computer Science

Joel R. Landry, Applied Economists and Management

March 16th 2010

Road Map

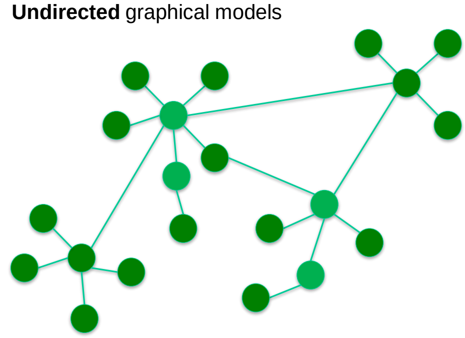
- Markov Random Field
- First Order Logic
- Markov Logic Network
- Inference
- Learning
- Comparisons
- Experiments and Software Alchemy

Background

- Markov Networks (Markov Random Field)
  - Definition
  - Example
- First-order logic
  - Definition
  - Example

Markov Networks

Undirected graphical models



Markov Networks

- Potential functions defined over cliques

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

- Log-linear model:

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

Weight of Feature *i* — Feature *i*

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

Example of First Order Logic

Smoking causes cancer.
Friends have similar smoking habits.

Example

$\forall x\ Smok$ ...

$\neg Smokes$ ...

$\forall x, y\ Fri$ ... $okes(y)$

$\neg Friends$ ... (y)

$\neg Friends$ ... (y)

First Order Logic

Knowledge base: $KB(C) \equiv \{F_1, \ldots, F_N\}$

a set of formulas that describe the relationships between ‘objects’ in a ‘world’ (a domain of interest, C).

For example, suppose we wish to explain the dating patterns of graduate students.

Our toy world would include graduate students (objects) as its domain, C. These could be specific objects (*constants*) or could range over the whole domain (*variables*), perhaps indexed by *type* (CS, Econ, etc.).

We would wish to characterize the set of important relationships between these students (*formulas*) that would allow us to evaluate why certain graduate students date.

First Order Logic

Formulas: F_i

Statements about the world.

“If X likes movies, and Y likes movies, X and Y would like to go on a date together.”

PredicateA(X)∧PredicateA(Y) ➔ PredicateB(X,Y)

First Order Logic

- *Predicates*:

  A mapping over constants or variables that returns True or False.

  LikeMovies(Bob) =1 if Bob likes movies; 0, otherwise.

- *Functions*:

  A mapping over objects that returns an object.

  Roomate(Sara)=Karen

First Order Logic

- *Terms*:

  Any predicate, function, constant or variable.

- *Atomic formula* (*atoms*, or *positive literal*):

  Any predicate over multiple terms.

  EnjoyedDate(Bob, Roomate(Sara));

  Also EnjoyedDate(Bob,Bob)

  Formulas are constructed from atomic formulas using logical connectives and qualifiers (∧,v,~,→,←→, universal and existential qualification).

  A negated atomic formulas is called a *negative literal*.

First Order Logic

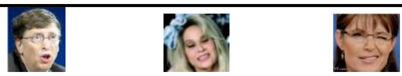
- *Clausal Form* (also clausal normal form of CNF)

  Regularization of a formula on the basis of forming conjunctions of clauses (themselves disjunctions of literals)

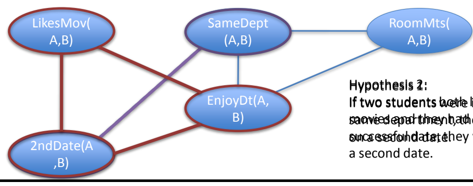
  E.g. CNF(F_i(A,B,C))=(AVB)∧(C)

- Inference in FOL is semi-decidable

  [(AVB)∧(C) is equivalent to (AVB)∧~(~C))], so we restrict ourselves to *Horn clauses* (clauses containing at most one positive literal).



X_{Bill}	X_{Karen}	X_{Sara}
X_1=1 (likes movies)	X_1=1 (likes movies)	X_1=0 (does not like movies)
X_2=1 (is roommates with himself)	X_2=0 (is not roommates with Bill)	X_2=0 (is not roommates with Bill)
X_3=0 (not roommates with Sara)	X_3=1 (is roommates with Sara)	X_3=1 (is roommates with herself)
X_4= 0 (not roommates with Karen)	X_4= 1 (is roommates with herself)	X_4= 1 (is roommates with Karen)
X_5=1 (in same dept. with himself)	X_5=0 (in diff dept. than Bill)	X_5=1 (in same dept. with Bill)
X_6=1 (in same dept. with Sara)	X_6=0 (in diff dept. than Sara)	X_6=1 (in same dept. with herself)
X_7=1 (in diff dept. than Karen)	X_7=1 (in same dept. with herself)	X_7=0 (in diff dept. than Karen)
X_8=1 (enjoyed date with himself)	X_8=1 (enjoyed date with Bill)	X_8=1 (enjoyed date with Bill)
X_9=1 (enjoyed date with Karen)	X_9=1 (enjoyed date with Karen)	X_9=0 (did not enjoy date with Karen)
X_{10}=0 (did not enjoy date with Sara)	X_{10}=0 (did not enjoy date with Sara)	X_{10}=1 (enjoyed date with herself)
X_{11}=1 (2nd date with himself)	X_{11}=1 (2nd date with Bill)	X_{11}=0 (did not mate with Bill)
X_{12}=1 (2nd date with Karen)	X_{12}=1 (2nd date with herself)	X_{12}=0 (did not mate with Karen)
X_{13}=0 (no 2nd date with Sara)	X_{13}=0 (no 2nd date with Sara)	X_{13}=1 (no 2nd date with herself)



Hypothesis 1:
If two students were in the same department, they went on a second date.

Hypothesis 2:
If two students both like the movies and they had a successful date, they went on a second date.

Example 1: Love Triangle

- Hypothesis 1:
  - If two students were in the same department and they had a successful date, they will have a 2ndDate.
    - F_1=SameDept(A,B) ∧ EnjoyDt(A,B)→2ndDate(A,B)
    - CNF(F_1)=~SameDept(A,B) V~EnjoyDt(A,B) V2ndDate(A,B)
- Hypothesis 2:
  - If two students both like movies and they had a successful date, they will have a 2ndDate.
    - F_2=LikesMov(A)∧LikesMov(B)∧EnjoyDt(A,B)→2ndDate(A,B)
    - CNF(F_2)=(~LikesMov(A) V ~LikesMov(B) V ~EnjoyDt(A,B))V2ndDate(A,B)

First Order Logic

- There is no uncertainty in first-order logic:
  - If a statement about our toy world is true it is ALWAYS true, so soften to *possible worlds*

First-Order Logic
If a formula describing the world is true it is always true.

Prob{F_i=T}=1.

Markov Logic Networks
If a formula is true in one world it may not be true in others. The more worlds it is true in, the more probable the formula is true.

Prob{F_i=T}=0.7.

Example 1: Love Triangle

- Hypothesis 1:
  - Under FOL:
    - Each is in the same department with themselves. Also, Karen and Sara/Bill are in different departments and did not go on a 2nd date. However, Bill is in the same department with Sara, and they did not go on a 2nd date→ so the statement is violated.
  - Under MLN:
    - In five out of six cases the claim is true→ so we assign a weight that reflects this reality.
- Hypothesis 2:
  - Under FOL:
    - This is true in all six cases.
  - Under MLN:
    - In six of six cases the claim is true→ so we we assign a weight that reflects this reality.

Markov Logic Network Overview

- Intuition and Definition
- Setup of MLN
- Examples
- Inference
- Learning
- Software

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 → 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

Markov Logic Networks

MLN is **template** for ground Markov nets

Probability of a world x:

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

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

MLN Assumptions

- **Unique names**. *Different constants refer to different objects*
- **Domain closure**. *The only objects in the domain are those representable using the constant and function symbols in (L,C)*
- **Known functions**. *For each function appearing in L, the value of that function applied to every possible tuple of arguments is known, and is an element of C.*

Grounding of MLN with Assumptions 1-3.

function Ground(F, C)
  inputs: F, a formula in first-order logic
      C, a set of constants
  output: G_F, a set of ground formulas
  calls: $CNF(F, C)$, which converts F to conjunctive normal form, replacing existentially quantified formulas by disjunctions of their groundings over C
$F \leftarrow CNF(F, C)$
$G_F = \emptyset$
for each clause $F_j \in F$
  $G_j = \{F_j\}$
  for each variable x in F_j
    for each clause $F_k(x) \in G_j$
      $G_j \leftarrow (G_j \setminus F_k(x)) \cup \{F_k(c_1), F_k(c_2), \ldots, F_k(c_{|C|})\}$,
        where $F_k(c_i)$ is $F_k(x)$ with x replaced by $c_i \in C$
  $G_F \leftarrow G_F \cup G_j$
for each ground clause $F_j \in G_F$
  repeat
    for each function $f(a_1, a_2, \ldots)$ all of whose arguments are constants
      $F_j \leftarrow F_j$ with $f(a_1, a_2, \ldots)$ replaced by c, where $c = f(a_1, a_2, \ldots)$
  until F_j contains no functions
return G_F

Example: 2nd Date

Same Department (X, Y) $\rightarrow$ 2ndDate(X, Y)

LikesMov(X) $\wedge$ LikesMov(Y) $\rightarrow$ 2ndDate(X, Y)

$\neg(LikesMov(X) \wedge LikesMov(Y) \wedge EnjoyDt(X,Y)) \vee 2ndDate(X,Y)$

(De morgan' s law)

$\neg LikesMov(X) \vee \neg LikesMov(Y) \vee \neg EnjoyDt(X,Y) \vee 2ndDate(X,Y)$

Example: 2nd Date

Constant set is { A, B, C}

$F = \neg LikesMov(X) \vee \neg LikesMov(Y) \vee \neg EnjoyDt(X,Y) \vee 2ndDate(X,Y)$

$G_f = \Phi$

Step 1

$F_1 = LikesMov(X)$

$G_1 = \{\neg LikesMov(A), \neg LikesMov(B), \neg LikesMov(C)\}$

$G_f = \{\neg LikesMov(A), \neg LikesMov(B), \neg LikesMov(C)\}$

Example: 2^{nd} Date

Constant set is { A, B, C}

$F = \neg LikesMov(X) \vee \neg LikesMov(Y) \vee \neg EnjoyDt(X,Y) \vee 2ndDate(X,Y)$

$G_f = \{\neg LikesMov(A), \neg LikesMov(B), \neg LikesMov(C)\}$

Step 2

$F_2 = \text{LikesMov}(Y)$

$G_2 = \{\neg LikesMov(A), \neg LikesMov(B), \neg LikesMov(C)\}$

$G_f = \{\neg LikesMov(A), \neg LikesMov(B), \neg LikesMov(C)\}$

Example: 2^{nd} Date

Constant set is { A, B, C}

$F = \neg LikesMov(X) \vee \neg LikesMov(Y) \vee \neg EnjoyDt(X,Y) \vee 2ndDate(X,Y)$

$G_f = \{\neg LikesMov(A), \neg LikesMov(B), \neg LikesMov(C)\}$

Step 3

$F_3 = \neg EnjoyDt(X,Y)$

$$G_3 = \{\neg EnjoyDt(A,A), \neg EnjoyDt(A,B), \neg EnjoyDt(A,C), \neg EnjoyDt(B,A), \neg EnjoyDt(B,B), \neg EnjoyDt(B,C), \neg EnjoyDt(C,A), \neg EnjoyDt(C,B), \neg EnjoyDt(C,A)\}$$

$$G_f = \{\neg LikesMov(A), \neg LikesMov(B), \neg LikesMov(C), \neg EnjoyDt(A,A), \neg EnjoyDt(A,B), \neg EnjoyDt(A,C), \neg EnjoyDt(B,A), \neg EnjoyDt(B,B), \neg EnjoyDt(B,C), \neg EnjoyDt(C,A), \neg EnjoyDt(C,B), \neg EnjoyDt(C,A)\}$$

Example: 2^{nd} Date

Constant set is { A, B, C}

$F = \neg LikesMov(X) \vee \neg LikesMov(Y) \vee \neg EnjoyDt(X,Y) \vee 2ndDate(X,Y)$

$$G_f = \{\neg LikesMov(A), \neg LikesMov(B), \neg LikesMov(C), \neg EnjoyDt(A,A), \neg EnjoyDt(A,B), \neg EnjoyDt(A,C), \neg EnjoyDt(B,A), \neg EnjoyDt(B,B), \neg EnjoyDt(B,C), \neg EnjoyDt(C,A), \neg EnjoyDt(C,B), \neg EnjoyDt(C,A)\}$$

Step 4

$F_4 = \neg 2ndDate(X,Y)$

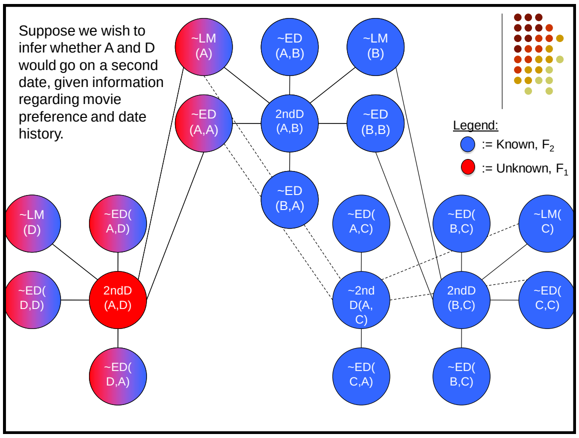
$$G_4 = \{\neg 2ndDate(A,A), \neg 2ndDate(A,B), \neg 2ndDate(A,C), \neg 2ndDate(B,A), \neg 2ndDate(B,B), \neg 2ndDate(B,C), \neg 2ndDate(C,A), \neg 2ndDate(C,B), \neg 2ndDate(C,A)\}$$

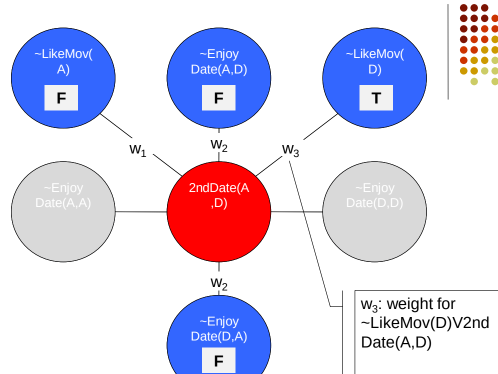
Update G_f accodingly by adding $G_f = G_f \cup G_4$

MLN Inference

- **Goal:**
$$P(F_1 \mid F_2, M_{L,C}) = \frac{P(F_1 \wedge F_2 \mid M_{L,C})}{P(F_2 \mid M_{L,C})} = \frac{\sum_{x \in \chi_{F1} \cap \chi_{F2}} P(X = x \mid M_{L,C})}{\sum_{x \in \chi_{F2}} P(X = x \mid M_{L,C})}$$
- **Phase 1:**
  - **Return the minimal subset *M* of the ground Markov network required to compute** P(F_1|F_2, $M_{L,C}$).
- **Phase 2:**
  - **Perform inference on *M*, using Gibbs sampling.**

Suppose we wish to infer whether A and D would go on a second date, given information regarding movie preference and date history.





Pr{2ndDate(A,D)=T|~LikeMov(A)=F,~EnjoyDate(A,D)=F,~LikeMov(D)=T,~EnjoyDate(D,A)=F}

$$P\left(X_l = x_l \mid B_l = b_l\right) = \frac{\exp\left(\sum_{f_i \in F_l} w_i f_i\left(X_l = x_l, B_l = b_l\right)\right)}{\exp\left(\sum_{f_i \in F_l} w_i f_i\left(X_l = 0, B_l = b_l\right)\right) + \exp\left(\sum_{f_i \in F_l} w_i f_i\left(X_l = 1, B_l = b_l\right)\right)}$$

Confusing part here is the f_i formulas…

2ndDate(A,D)=T	2ndDate(A,D)=F
f_1[~LikeMov(A)=F V 2ndDate(A,D)=T]=1	f_1[~LikeMov(A)=F V 2ndDate(A,D)=F]=0
f_2[~EnjoyDate(A,D)=F V 2ndDate(A,D)=T]=1	f_2[~EnjoyDate(A,D)=F V 2ndDate(A,D)=F]=0
f_3[~LikeMov(D)=T V 2ndDate(A,D)=T]=1	f_3[~LikeMov(D)=T V 2ndDate(A,D)=F]=1

Pr{2nd(A,D)=T | MB}=e^(w_1+2w_2+w_3)/[e^(w_1+2w_2+w_3)+e^(w_3)]

Pr{2nd(A,D)=F | MB}=e^(w_3)/[e^(w_1+2w_2+w_3)+e^(w_3)]

Learning

- Assumption (closed world; if data is missing then set to False)
- MC-MLE
- Pseudo-likelihood
- CRFs
- Structural SVMs

Generative Weight Learning

Maximize likelihood or posterior probability

Numerical optimization (gradient or 2nd order)

No local maxima

$$\frac{\partial}{\partial w_i} \log P_w(X = x) = n_i(x) - \sum_{x'} P_w\left(X = x'\right) n_i(x')$$

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

Expected no. times feature *i* is true according to model

Requires inference at each step (slow! using MC-MLE)

Generative Weight Learning

$$P_w(X = x) = \frac{1}{Z} \exp(w_j n_j(x))$$

$$\log P_w(X = x) = \sum_j \exp(w_j n_j(x)) - \log Z$$

Taking the Partial Derivative

$$\frac{\partial}{\partial w_i} \log P_w(X = x) = \frac{\partial}{\partial w_i} \sum_j \exp(w_j n_j(x)) - \frac{\partial}{\partial w_i} \log Z$$

$$= n_i(x) - \frac{1}{Z} \frac{\partial}{\partial w_i} Z$$

$$= n_i(x) - \frac{1}{Z} \sum_{x'} \frac{\partial}{\partial w_i} \exp\left(\sum_j w_j n_j(x')\right)$$

$$= n_i(x) - \frac{1}{Z} \sum_{x'} \exp\left(\sum_j w_j n_j(x')\right) n_i(x')$$

$$= n_i(x) - \sum_{x'} P_w(X = x') n_i(x')$$

Pseudo-Likelihood

$$PL(x) \equiv \prod_i P(x_i \mid neighbors(x_i))$$

- Likelihood of each variable given its neighbors in the data
- Does not require inference at each step
- Consistent estimator

Pseudo-Likelihood

$$PL(x) = \prod_l P(X_l = x_l \mid MB(x_l))$$

$$\log PL(x) = \sum_l \log P(X_l = x_l \mid MB(x_l))$$

$$P(X_l = x_l \mid MB(x_l)) = \frac{P(x)}{P(x_{[X_l=0]}) + P(x_{[X_l=1]})}$$

$$= \frac{1/Z \exp(\Sigma w_i n_i(x))}{1/Z \exp(\Sigma w_i n_i(x_{[X_l=0]})) + 1/Z \exp(\Sigma w_i n_i(x_{[X_l=1]}))}$$

$$\frac{\partial}{\partial w_j} \log PL(x) = \sum_l n_j(x) - P(X_l = 0 \mid MB(X_l))\, n_j(x_{[X_l=0]}) - P(X_l = 1 \mid MB(X_l))\, n_j(x_{[X_l=1]})$$

Pseudo-Likelihood

Efficiency tricks:

- Compute each $n_j(x)$ only once
- Skip formulas in which x_l does not appear
- Skip groundings of clauses with > 1 true literal
  e.g., (A v ¬B v C) when A=1, B=0

Optimizing pseudo-likelihood

- Pseudo-log likelihood is convex
- Standard convex optimization algorithms
  work great (e.g., L-BFGS quasi-Newton method)

Pseudo-Likelihood

Pros

- Efficient to compute
- Consistent estimator

Cons

- Works poorly with long-range dependencies

Discriminative Weight Learning

- Conditional Random Fields
- M3LN

Discriminative Weight Learning (using Conditional Random Fields)

Maximize conditional likelihood of query (y) given evidence (x)

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

No. of true groundings of clause *i* in data

Expected no. true groundings according to model

Voted perceptron: Approximate expected counts by counts in MAP state of y given x

Discriminative Weight Learning (using M3LN (Huynh and Mooney, 2009))

$$\left[\begin{array}{l} \min_{\mathbf{w},\xi \geq 0} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\xi \\ s.t. \quad \forall \bar{\mathbf{y}} \in \mathrm{Y}: \mathbf{w}^T\left[\mathbf{n}(\mathbf{x},\mathbf{y}) - \mathbf{n}(\mathbf{x},\bar{\mathbf{y}})\right] \geq \Delta(\mathbf{y},\bar{\mathbf{y}}) - \xi \end{array}\right]$$

Effectively:

$$\Psi(\mathbf{x},\mathbf{y}) = \mathbf{n}(\mathbf{x},\mathbf{y})$$

Use MaxWalkSAT for prediction, and LP-relaxation, other MPE inference algorithm.

Alchemy

- *http://alchemy.cs.washington.edu*
- Alchemy provides a series of algorithms for statistical relational learning and probabilistic logic inference, based on MLN
  - Collective classification
  - Link prediction
  - Entity resolution
  - Social network modeling
  - Information extraction

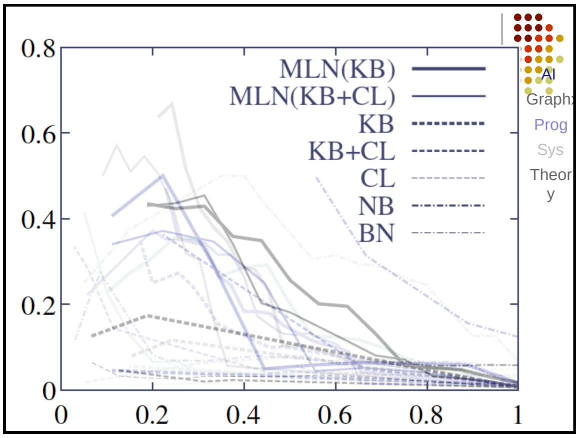
Conclusion

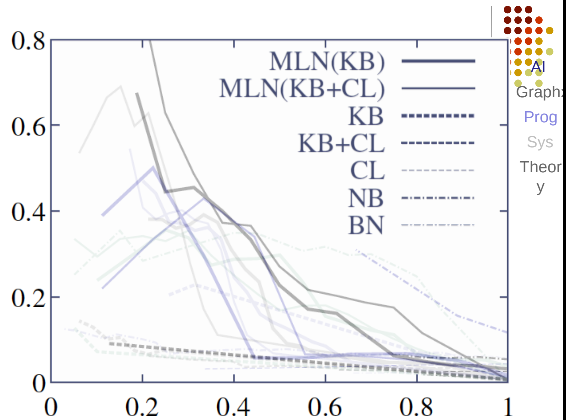
MLN are versatile ways to represent first order logic using Markov random fields

They can be used to construct a template upon which other estimation strategies, such as structural SVMs, can be attached.

If you like movies, you might get a second date with Yue or Joel. (Talk to us after class)

Appendix





Example: Friends & Smokers

Smoking causes cancer.
Friends have similar smoking habits.

Example: Friends & Smokers

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

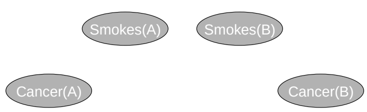
$\forall x, y\ Friends(x, y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$

Example: Friends & Smokers

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

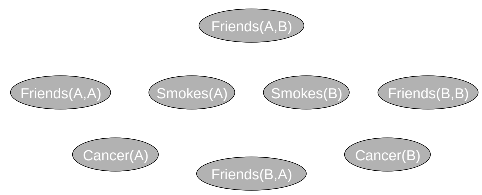
Example: Friends & Smokers

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



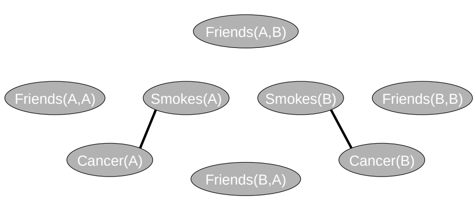
Example: Friends & Smokers

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



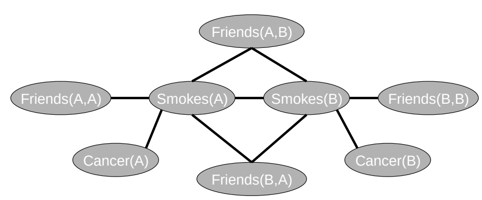
Example: Friends & Smokers

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



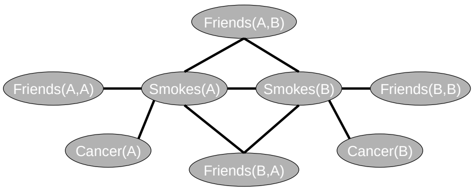
Example: Friends & Smokers

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



Example Grounding

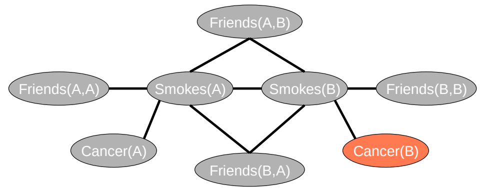
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$



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

Example Grounding

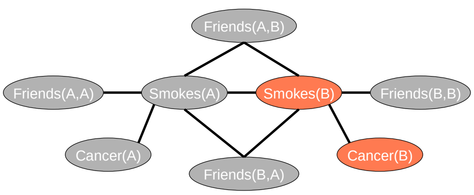
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$



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

Example Grounding

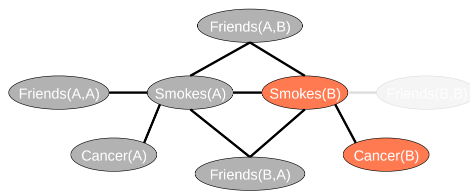
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$



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

Example Grounding

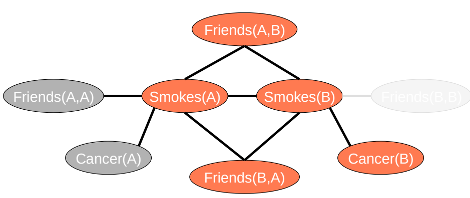
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$



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

Example Grounding

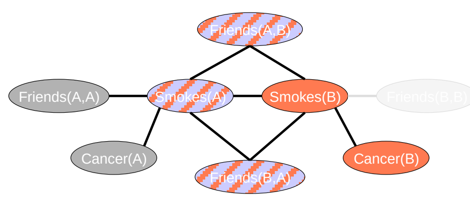
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$



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

Example Grounding

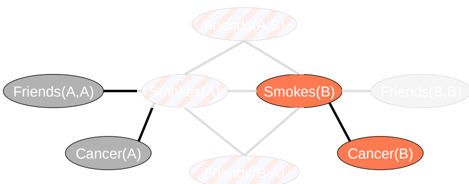
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$



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

Example Grounding

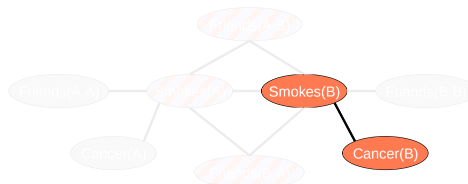
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x,y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$



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

Example Grounding

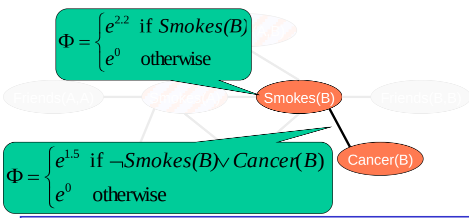
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x,y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$



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

Example Grounding

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



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