Probabilistic Logic Programming

CS267A - Fall 2018
Guy Van den Broeck

Many slides taken from Luc De Raedt and Angelika Kimmig
Overview

- Datalog
- Prolog
- ProbLog language
- Problog inference 1.0
- Problog learning
Challenges for the Future

Statistical Relational AI:
The study and design of intelligent agents that act in noisy worlds composed of objects and relations among the objects.
Datalog
Non-Recursive Datalog: Grandfather
Non-Recursive Datalog as First-Order Logic
Recursive Datalog
Datalog

\[
\text{path}(X,Y) :\neg \text{edge}(X,Y).
\]
\[
\text{path}(X,Y) :\neg \text{edge}(X,Z), \text{path}(Z,Y).
\]

\[
\text{path}(a,d) = \text{Yes}
\]
Datalog vs. First-Order Logic
Prolog
Prolog: Append
Prolog

member(X, [X|_]).  % member(X, [Head|Tail]) is true if X = Head

member(X, [_|Tail]) :-  % or if X is a member of Tail,
    member(X, Tail).  % ie. if member(X, Tail) is true.

?- member(a, [a, b]).
True.
member(X, [X|_]).  % member(X, [Head|Tail]) is true if X = Head

member(X, [__|Tail]) :-  % or if X is a member of Tail,
  member(X, Tail).  % ie. if member(X, Tail) is true.

?- member(c, [a, b]).
Fail.
Prolog

member(X, [X|_]). % member(X, [Head|Tail]) is true if X = Head

member(X, [_|Tail]) :- % or if X is a member of Tail,
member(X, Tail). % ie. if member(X, Tail) is true.

?- member(X, [a, b]).
X = a ;
X = b ;
fail.
Prolog

member(X, [X|_]).  % member(X, [Head|Tail]) is true if X = Head
member(X, [_|Tail]) :-  % or if X is a member of Tail,
  member(X, Tail).  % ie. if member(X, Tail) is true.

?- member(a, [X, b]).
X = a ;
fail.
Prolog

member(X, [X|_]). % member(X, [Head|Tail]) is true if X = Head
member(X, [_|Tail]) :- % or if X is a member of Tail,
    member(X, Tail). % ie. if member(X, Tail) is true.

?- member(a, X).
X = [a|Y] ;
X = [Y, a|Z] ;
X = [Y, Z, a|W] ;
...

quick_sort([],[]).
quick_sort([H|T],Sorted):-
pivoting(H,T,L1,L2),
quick_sort(L1,Sorted1),
quick_sort(L2,Sorted2),
append(Sorted1,[H|Sorted2]).

pivoting(H,[],[],[]).
pivoting(H,[X|T],[X|L],G):-
  X=<H,
pivoting(H,T,L,G).
pivoting(H,[X|T],L,[X|G]):-
  X>H,
pivoting(H,T,L,G).
Turing-completeness

Reasonable test whether something is a programming language

rule(q0, 1, q0, 1, right).
rule(q0, b, qf, 1, stay).

Turing(Tape0, Tape) :-
    perform(q0, [], Ls, Tape0, Rs),
    reverse(Ls, Ls1),
    append(Ls1, Rs, Tape).

perform(qf, Ls, Ls, Rs, Rs) :- !.
perform(Q0, Ls0, Ls, Rs0, Rs) :-
    symbol(Rs0, Sym, RsRest),
    once(rule(Q0, Sym, Q1, NewSym, Action)),
    action(Action, Ls0, Ls1, [NewSym|RsRest], Rs1),
    perform(Q1, Ls1, Ls, Rs1, Rs).

symbol([], b, []).
symbol([Sym|Rs], Sym, Rs).

action(left, Ls0, Ls, Rs0, Rs) :- left(Ls0, Ls, Rs0, Rs).
action(stay, Ls, Ls, Rs, Rs).
action(right, Ls0, [Sym|Ls0], [Sym|Rs], Rs).

left([], [], Rs0, [b|Rs0]).
left([L|Ls], Ls, Rs, [L|Rs]).
ProbLog Language
ProLog
probabilistic Prolog

Distribution Semantics [Sato, ICLP 95]:
probabilistic choices + logic program → distribution over possible worlds

one world

several possible worlds

Facts as random variables

Parlamentary learning, Relational learning techniques

Prolog / logic programming
Reasoning with relational data / datalog

stress(ann).
influences(ann,bob).
influences(bob,carl).
smokes(X) :- stress(X).
smokes(X) :-
influences(Y,X), smokes(Y).

0.8::stress(ann).
0.6::influences(ann,bob).
0.2::influences(bob,carl).
ProbLog by example:

A bit of gambling

- toss (biased) coin & draw ball from each urn
- win if (heads and a red ball) or (two balls of same color)

0.4 :: heads.  **probabilistic fact**: heads is true with probability 0.4
(and false with 0.6)
ProbLog by example:

A bit of gambling

- toss (biased) coin & draw ball from each urn
- win if (heads and a red ball) or (two balls of same color)

0.4 :: heads.

annotated disjunction: first ball is red with probability 0.3 and blue with 0.7

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
ProbLog by example:

A bit of gambling

• toss (biased) coin & **draw ball from each urn**

• win if (heads and a red ball) or (two balls of same color)

0.4 :: heads.

**annotated disjunction:** first ball is red with probability 0.3 and blue with 0.7

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.

0.2 :: col(2,red); 0.3 :: col(2,green);
    0.5 :: col(2,blue) <- true.

**annotated disjunction:** second ball is red with probability 0.2, green with 0.3, and blue with 0.5
ProbLog by example:

A bit of gambling

- toss (biased) coin & draw ball from each urn
- **win if (heads and a red ball) or (two balls of same color)**

\[ \begin{align*}
0.4 &:: \text{heads}. \\
0.3 &:: \text{col}(1,\text{red});
0.7 &:: \text{col}(1,\text{blue}) \leftarrow \text{true.} \\
0.2 &:: \text{col}(2,\text{red});
0.3 &:: \text{col}(2,\text{green});
0.5 &:: \text{col}(2,\text{blue}) \leftarrow \text{true.}
\end{align*} \]

\text{win} \leftarrow \text{heads, col(\_ ,red).} \quad \text{logical rule encoding background knowledge}
ProbLog by example:

A bit of gambling

• toss (biased) coin & draw ball from each urn
• win if (heads and a red ball) or (two balls of same color)

0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green);
  0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).

logical rule encoding background knowledge
ProbLog by example:

A bit of gambling

• toss (biased) coin & draw ball from each urn
• win if (heads and a red ball) or (two balls of same color)

0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green);
0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).
Probabilistic Queries

0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).

• Probability of \textbf{win} ?
• Probability of \textbf{win} given \textbf{col(2,green)} ?
• Most probable world where \textbf{win} is true?

marginal probability  conditional probability  evidence  MPE inference
Possible Worlds

\[0.4 \times 0.3 \times 0.3\]

\[0.4 :: \text{heads.}\]

\[0.3 :: \text{col}(1, \text{red}) ; 0.7 :: \text{col}(1, \text{blue}) \leftarrow \text{true}\]

\[0.2 :: \text{col}(2, \text{red}) ; 0.3 :: \text{col}(2, \text{green}) ; 0.5 :: \text{col}(2, \text{blue}) \leftarrow \text{true}\]

\[
\text{win} :- \text{heads, col}(\_, \text{red}). \\
\text{win} :- \text{col}(1, \text{C}), \text{col}(2, \text{C}).
\]
Possible Worlds

0.4 :: heads.

0.3 :: col(1, red); 0.7 :: col(1, blue) <- true.

0.2 :: col(2, red); 0.3 :: col(2, green); 0.5 :: col(2, blue) <- true.

win :- heads, col(_, red).
win :- col(1, C), col(2, C).

\[
\begin{align*}
0.4 \times 0.3 \times 0.3 & \quad (1-0.4) \times 0.3 \times 0.2
\end{align*}
\]
Possible Worlds

0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).

0.4 × 0.3 × 0.3  (1−0.4)×0.3 ×0.2  (1−0.4)×0.3 ×0.3
All Possible Worlds

0.024
H R R W

0.036
R R W

0.056
H B R W

0.084
B R

0.036
H R G W

0.054
R G

0.084
H B G

0.126
B G

0.060
H R B W

0.090
R B

0.140
H B B W

0.210
B B W
Most likely world where \textit{win} is true?

\begin{tabular}{cccc}
0.024 & 0.036 & 0.056 & 0.084 \\
\textcolor{red}{H} & \textcolor{red}{R} & \textcolor{red}{R} & \textcolor{blue}{W} \\
0.036 & 0.054 & 0.084 & 0.126 \\
\textcolor{red}{H} & \textcolor{red}{R} & \textcolor{green}{G} & \textcolor{green}{W} \\
0.060 & 0.090 & 0.140 & 0.210 \\
\textcolor{red}{H} & \textcolor{red}{R} & \textcolor{blue}{B} & \textcolor{blue}{W} \\
\end{tabular}
Most likely world where $\text{col}(2, \text{blue})$ is false?

MPE Inference
\[ P(\text{win}) = ? = 0.562 \]
\[ P(\text{win} \mid \text{col}(2, \text{green})) = \frac{?}{\sum} \]

\[ P(\text{win} \land \text{col}(2, \text{green}))/P(\text{col}(2, \text{green})) \]
\[ P(\text{win}|\text{col}(2, \text{green})) = \frac{\sum}{\sum} \]

\[ = 0.036/0.3 = 0.12 \]
Distribution Semantics
(with probabilistic facts)

\[ P(Q) = \sum_{F \cup R \models Q} \prod_{f \in F} p(f) \prod_{f \notin F} 1 - p(f) \]

query
sum over possible worlds
where Q is true
subset of
collaborative
probabilistic
facts
Prolog rules
probability of
possible world
Probabilistic Datalog

path(X,Y) :- edge(X,Y).
path(X,Y) :- edge(X,Z), path(Z,Y).

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>c</td>
<td>0.3</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>0.9</td>
</tr>
<tr>
<td>b</td>
<td>c</td>
<td>0.4</td>
</tr>
<tr>
<td>c</td>
<td>d</td>
<td>0.5</td>
</tr>
</tbody>
</table>

\[ P(path(a,d)) = ?? \]
Probabilistic Datalog

\[ P(path(a,d)) = 0.276 \]

\[
\begin{align*}
0.3 & \cdot \text{edge}(a,c) . \\
0.9 & \cdot \text{edge}(a,b) . \\
0.4 & \cdot \text{edge}(b,c) . \\
0.5 & \cdot \text{edge}(c,d) . \\
\end{align*}
\]

\[
\text{path}(X,Y) : - \text{edge}(X,Y) . \\
\text{path}(X,Y) : - \text{edge}(X,Z), \ \text{path}(Z,Y) .
\]
smokes(X) :- stress(X).
smokes(X) :-
    friend(X,Y), influences(Y,X), smokes(Y).

0.3::stress(1).
0.3::stress(2).
0.3::stress(3).
0.3::stress(4).

0.2::influences(1,1).
0.2::influences(1,2).
0.2::influences(1,3).
0.2::influences(1,4).
0.2::influences(2,1).
0.2::influences(2,2).
0.2::influences(2,3).
0.2::influences(2,4).
0.2::influences(3,1).
0.2::influences(3,2).
0.2::influences(3,3).
0.2::influences(3,4).
0.2::influences(4,1).
0.2::influences(4,2).
0.2::influences(4,3).
0.2::influences(4,4).

person(1).
person(2).
person(3).
person(4).

friend(1,2).
friend(2,1).
friend(2,4).
friend(3,4).
friend(4,2).
smokes(X) :- stress(X).
smokes(X) :-
    friend(X,Y), influences(Y,X), smokes(Y).

0.3::stress(X):- person(X).
0.2::influences(X,Y):-
    person(X), person(Y).
smokes(X) :- stress(X).
smokes(X) :-
    friend(X,Y), influences(Y,X), smokes(Y).

0.3::stress(X):- person(X).
0.2::influences(X,Y):-
    person(X), person(Y).
0.4::asthma(X) :- smokes(X).
Rain or sun?

0.5::weather(sun,0) ; 0.5::weather(rain,0) :- true.
0.6::weather(sun,T) ; 0.4::weather(rain,T)
    :- T>0, Tprev is T-1, weather(sun,Tprev).
0.2::weather(sun,T) ; 0.8::weather(rain,T)
    :- T>0, Tprev is T-1, weather(rain,Tprev).

infinite possible worlds! BUT: finitely many partial worlds suffice to answer any given ground query.
ProbLog by example:

Limited Luggage

weight(skis,6).
weight(boots,4).
weight(helmet,3).
weight(gloves,2).
ProbLog by example:

**Limited Luggage**

weight(skis,6).
weight(boots,4).
weight(helmet,3).
weight(gloves,2).

P::pack(Item) :- weight(Item,Weight), \textbf{P is 1.0/Weight.}

**flexible probability:**
computed from the weight of the item

\begin{align*}
1/6::&\text{pack(skis)}.
1/4::&\text{pack(boots)}.
1/3::&\text{pack(helmet)}.
1/2::&\text{pack(gloves)}.
\end{align*}
ProbLog by example:

Limited Luggage

weight(skis,6).
weight(boots,4).
weight(helmet,3).
weight(gloves,2).

P::pack(Item) :- weight(Item,Weight), P is 1.0/Weight.

excess(Limit) :- excess([skis,boots,helmet,gloves],Limit).

list of all items
ProbLog by example:

Limited Luggage

weight(skis,6).
weight(boots,4).
weight(helmet,3).
weight(gloves,2).

P::pack(Item) :- weight(Item,Weight), P is 1.0/Weight.

excess(Limit) :- excess([[skis,boots,helmet,gloves]],Limit).

\textbf{excess([I,R],Limit)} :- pack(I), weight(I,W), L is Limit-W, excess(R,L).
\textbf{excess([I|R],Limit)} :- \textbf{\&+pack(I)}, excess(R,Limit).

\textbf{do not pack first item, continue with rest of items}

\textbf{no items left: did we add too much?}
ProbLog

- probabilistic choices + their consequences
- probability distribution over possible worlds
- how to efficiently answer questions?
  - most probable world (MPE inference)
  - probability of query (computing marginals)
  - probability of query given evidence
Summary: ProbLog Syntax

- input database: ground facts
  
- probabilistic facts
  
- typed probabilistic facts (body deterministic)
  
- flexible probabilities
  
- annotated disjunctions
  
- Prolog clauses

- person(bob).
- 0.5::stress(bob).
- 0.5::stress(X) :- person(X).
- P::pack(Item) :- weight(Item,W), P is 1.0/W.
- 0.4::asthma(X) :- smokes(X).
- 0.5::weather(sun,0) ; 0.5::weather(rain,0) :- true.
- smokes(X) :- influences(Y,X), smokes(Y).
- excess([I|R],Limit) :- \+pack(I), excess(R,Limit).
Language Extensions and Variants

Causal Probabilistic Time-Logic

Dynamics

Variants

Distributional Clauses

Continuous Distributions

ProbLog

Utilities and Decisions

Decision-Theoretic ProbLog

Extensions

ProbLog with Constraints

Extensions not included in ProbLog system
Sample from a List

member(X, [X|_]). % member(X, [Head|Tail]) is true if X = Head

member(X, [__|Tail]) :- % or if X is a member of Tail,
member(X, Tail). % ie. if member(X, Tail) is true.
Sample from a List

```prolog
c sample([X|L], X) :- sample_now([X|L]).
sample([H|L], X) :- \+ sample_now([H|L]), sample(L,X).

P::sample_now(L) :- length(L, N), P is 1/N.
```

Pr(sample([c,a,c,t,u,s],c)) = 0.33
Sample without Replacement

cample(L,N,S) :-
    permutation(S,T),
    sample_ordered(L,N,T).

cample_ordered(_, 0, []).
cample_ordered([X|L], N, [X|S]) :-
    N > 0, cample_now([X|L],N),
    N2 is N-1, cample_ordered(L,N2,S).
cample_ordered([H|L], N, S) :-
    N > 0, " sample_now([H|L],N),
    sample_ordered(L,N,S).

P := cample_now(L,N) :- length(L, M), M >= N, P is N/M.

Pr(cample([c,a,c,t,u,s],3,[c,a,t])) = 0.1
Unreliable Vending Machine

https://dtai.cs.kuleuven.be/problog/editor.html#task=prob&hash=55534b898e6c4d01db7315599c0face6

% probability of different types of coins being accepted by the machine
0.80 :: accept_first_coin([10|_]).
0.90 :: accept_first_coin([25|_]).
0.75 :: accept_first_coin([50|_]).

% does my purse have enough money accepted by the machine?
accept_coins(_,Price) :- Price <= 0.
accept_coins([H|T],Price) :-
    accept_first_coin([H|T]),
    RemainingPrice is Price - H,
    accept_coins(T,RemainingPrice).
accept_coins([H|T],Price) :-
    + accept_first_coin([H|T]),
    accept_coins(T,Price).

0.05 :: product_gets_stuck.

can_buy(Purse,Price) :- accept_coins(Purse,Price), + product_gets_stuck.
ProbLog Inference
Answering Questions

1. using proofs
2. using models

Given:
- program
- queries
- evidence

Find:
- marginal probabilities
- conditional probabilities
- MPE state

logical reasoning
data structure
probabilistic inference
Logical Reasoning: Proofs in Prolog

```
stress(ann).
influences(ann,bob).
influences(bob,carl).

smokes(X) :- stress(X).
smokes(X) :- influences(Y,X),smokes(Y).
```

```
?- smokes(carl).
?- stress(carl).
?- influences(Y,carl),smokes(Y).
?- smokes(bob).
?- stress(bob).
?- influences(Y1,bob),smokes(Y1).
?- smokes(ann).
?- influences(Y2,ann),smokes(Y2).

proof = facts used in successful derivation:
influences(bob,carl) & influences(ann,bob) & stress(ann)
Proofs in ProbLog

\[
\begin{align*}
0.8 &:: \text{stress(ann)}. \\
0.6 &:: \text{influences(ann,bob)}. \\
0.2 &:: \text{influences(bob,carl)}. \\

\text{smokes(X)} :&= \text{stress(X)}. \\
\text{smokes(X)} :&= \\
&\quad \text{influences(Y,X)}, \quad \text{smokes(Y)}. \\

\text{smokes(carl)}: \quad Y=bob \\
\text{smokes(bob)}: \quad Y1=ann \\
\text{smokes(ann)}: \\

\text{probability of proof} = 0.2 \times 0.6 \times 0.8 = 0.096
\end{align*}
\]
Proofs in ProbLog

0.8::stress(ann).
0.4::stress(bob).
0.6::influences(ann,bob).
0.2::influences(bob,carl).

smokes(X) :- stress(X).
smokes(X) :- influences(Y,X), smokes(Y).

? :- smokes(carl).
? :- stress(carl).
? :- influences(Y,carl),smokes(Y).
? :- smokes(bob).
? :- stress(bob).
? :- influences(Y1,bob),smokes(Y1).
? :- smokes(ann).
? :- influences(Y2,ann),smokes(Y2).

influences(bob,carl) & stress(bob)
0.2×0.4 = 0.08

influences(bob,carl) & stress(bob)
0.2×0.6×0.8 = 0.096

proofs overlap!
cannot sum probabilities (disjoint-sum-sum-problem)
## Disjoint-Sum-Problem

### possible worlds

\[
\text{influences}(\text{bob, carl}) \land \text{influences}(\text{ann, bob}) \land \text{stress}(\text{ann})
\]

| Infl (bob, carl) & Infl (ann, bob) & St (ann) & \+St (bob) | Probability |
|----------------|----------------|------|----------|------------|
| \text{infl}(\text{bob, carl}) \land \text{infl}(\text{ann, bob}) \land \text{st}(\text{ann}) \land \text{\+st}(\text{bob}) | 0.0576 |
| \text{infl}(\text{bob, carl}) \land \text{infl}(\text{ann, bob}) \land \text{st}(\text{ann}) \land \text{st}(\text{bob}) | 0.0384 |
| \text{infl}(\text{bob, carl}) \land \text{\+infl}(\text{ann, bob}) \land \text{st}(\text{ann}) \land \text{st}(\text{bob}) | 0.0256 |
| \text{infl}(\text{bob, carl}) \land \text{infl}(\text{ann, bob}) \land \text{\+st}(\text{ann}) \land \text{st}(\text{bob}) | 0.0096 |
| \text{infl}(\text{bob, carl}) \land \text{\+infl}(\text{ann, bob}) \land \text{\+st}(\text{ann}) \land \text{st}(\text{bob}) | 0.0064 |

... \text{influences}(\text{bob, carl}) \land \text{stress}(\text{bob})

\[
\sum = 0.1376
\]

**sum of proof probabilities:** 0.096 + 0.08 = 0.1760
**Binary Decision Diagrams**

- compact graphical representation of Boolean formula
- popular in many branches of CS

---

\[
\text{smokes}(c) = \neg i(b,c) \land s(b) \lor \neg i(b,c) \land i(a,b) \land s(a)
\]

---

[Bryant 86]
Binary Decision Diagrams

- compact graphical representation of Boolean formula
- popular in many branches of CS
- automatically disjoins proofs

![Binary Decision Diagram Example]

\[
influences(bob, carl) \land stress(bob)
\]

\[
influences(bob, carl) \land influences(ann, bob)
\]

\[
\land stress(ann)
\]

\[
\land \neg stress(bob)
\]
Binary Decision Diagrams

probability of \texttt{smokes}(c) = \texttt{i}(b,c)\&\texttt{s}(b) \lor \texttt{i}(b,c)\&\texttt{i}(a,b)\&\texttt{s}(a)
Initial Approach
(ProbLog1)

Find all proofs of query

Binary Decision Diagram (BDD)

calculate marginal by dynamic programming

\[ P(\text{win}) = \text{probability of reaching 1-leaf} \]

[De Raedt et al, IJCAI 07; Kimmig et al, TPLP 11]
Answering Questions

Given:
- program
- queries
- evidence

Find:
- marginal probabilities
- conditional probabilities
- MPE state
- logical reasoning
- data structure
- probabilistic inference

1. using proofs
2. using models
Answering Questions

1. using proofs
2. using models

Given:
- program
- queries
- evidence

Find:
- marginal probabilities
- conditional probabilities
- MPE state

logical reasoning
data structure
probabilistic inference
propensity model

\[ P(Q) = \sum_{F \cup R \models Q} \prod_{f \in F} p(f) \prod_{f \not\in F} (1 - p(f)) \]

possible worlds

\[ WMC(\phi) = \sum_{I_V \models \phi} \prod_{l \in I} w(l) \]

interpretations (truth value assignments) of propositional variables

weight of literal

for \( p :: f \),
\[ w(f) = p \]
\[ w(\neg f) = 1 - p \]
Current Approach  
(ProbLog2)

Find relevant ground program for queries & evidence

Weighted CNF

use weighted model counting / satisfiability

win :- heads(1).
win :- heads(2), heads(3).

win <-> h(1) V (h(2) & h(3))

(¬win V h(1) V h(2))
& (¬win V h(1) V h(3))
& (win V ¬h(1))
& (win V ¬h(2) V ¬h(3))

h(1) -> 0.4  h(2) -> 0.7  h(3) -> 0.5
¬h(1) -> 0.6  ¬h(2) -> 0.3  ¬h(3) -> 0.5

use standard tool

[Fierens et al, TPLP 14]
Lifted ProbLog Inference?
ProbLog Learning
Parameter Learning

e.g., webpage classification model

for each $CLASS1$, $CLASS2$ and each $WORD$

$\texttt{class(Page,C) :- has_word(Page,W), word_class(W,C)}.$

$\texttt{class(Page,C) :-}$

$\texttt{links_to(OtherPage,Page),}$

$\texttt{class(OtherPage,OtherClass),}$

$\texttt{link_class(OtherPage,Page,OtherClass,C).}$
Parameter Estimation

\[ p(\text{fact}) = \frac{\text{count(\text{fact is true})}}{\text{Number of interpretations}} \]
Learning from partial interpretations

- Not all facts observed
- Soft Expectation-Maximization
- Use **expected count** instead of **count**
- \( P(Q|E) \) -- conditional queries!

[Gutmann et al, ECML 11; Fierens et al, TPLP 14]
Getting started

• http://dtai.cs.kuleuven.be/problog
• interactive tutorial
• online interface for inference and parameter estimation
• offline version for download
End