Probabilistic Logic Programming with cplint

Week 1, lecture 1: syntax, semantics and exact inference

Fabrizio Riguzzi

Outline

- Probabilistic logic programming
- Applications
- Exact inference

Combining Logic and Probability

- Logic does not handle well uncertainty
- Graphical models do not handle well relationships among entities
- Solution: combine the two
- Many approaches proposed in the areas of Logic Programming, Uncertainty in AI, Machine Learning, Databases, Knowledge Representation

Probabilistic Logic Programming

- Distribution Semantics [Sato ICLP95]
- A probabilistic logic program defines a probability distribution over normal logic programs (called instances or possible worlds or simply worlds)
- The distribution is extended to a joint distribution over worlds and interpretations (or queries)
- The probability of a query is obtained from this distribution

Probabilistic Logic Programming (PLP) Languages under the Distribution Semantics

- Probabilistic Logic Programs [Dantsin RCLP91]
- Probabilistic Horn Abduction [Poole NGC93], Independent Choice Logic (ICL) [Poole AI97]
- PRISM [Sato ICLP95]
- Logic Programs with Annotated Disjunctions (LPADs) [Vennekens et al. ICLP04]
- ProbLog [De Raedt et al. IJCAI07]
- They differ in the way they define the distribution over logic programs

Logic Programs with Annotated Disjunctions

`http://cplint.eu/e/sneezing_simple.pl`

$$sneezing(X) : 0.7 \ ; \ null : 0.3 \leftarrow flu(X).$$
$$sneezing(X) : 0.8 \ ; \ null : 0.2 \leftarrow hay\_fever(X).$$
$$flu(bob).$$
$$hay\_fever(bob).$$

- Distributions over the head of rules
- *null* does not appear in the body of any rule
- Worlds obtained by selecting one atom from the head of every grounding of each clause

Distribution Semantics

- Case of no function symbols: finite Herbrand universe, finite set of groundings of each disjoint statement/switch/clause
- Atomic choice: selection of the i-th atom for grounding $C\theta$ of switch/clause C
  - represented with the triple (C, θ, i)
- Example $C_1 = sneezing(X) : 0.7 \; ; \; null : 0.3 \leftarrow flu(X).$, $(C_1, \{X/bob\}, 1)$

Distribution Semantics

- Selection σ: a total set of atomic choices (one atomic choice for every grounding of each clause)
- A selection σ identifies a logic program w_σ called world
- The probability of w_σ is $P(w_\sigma) = \prod_{(C,\theta,i)\in\sigma} P_0(C,i)$
- Finite set of worlds: $W_T = \{w_1, \ldots, w_m\}$
- $P(w)$ distribution over worlds: $\sum_{w\in W_T} P(w) = 1$

Distribution Semantics

- Ground query Q
- $P(Q|w) = 1$ if Q is true in w and 0 otherwise
- $P(Q) = \sum_w P(Q, w) = \sum_w P(Q|w)P(w) = \sum_{w \models Q} P(w)$

Example Program (LPAD) Worlds

`http://cplint.eu/e/sneezing_simple.pl`

$sneezing(bob) \leftarrow flu(bob).$
$sneezing(bob) \leftarrow hay\_fever(bob).$
$flu(bob).$
$hay\_fever(bob).$
$P(w_1) = 0.7 \times 0.8$

$null \leftarrow flu(bob).$
$sneezing(bob) \leftarrow hay\_fever(bob).$
$flu(bob).$
$hay\_fever(bob).$
$P(w_2) = 0.3 \times 0.8$

$sneezing(bob) \leftarrow flu(bob).$
$null \leftarrow hay\_fever(bob).$
$flu(bob).$
$hay\_fever(bob).$
$P(w_3) = 0.7 \times 0.2$

$null \leftarrow flu(bob).$
$null \leftarrow hay\_fever(bob).$
$flu(bob).$
$hay\_fever(bob).$
$P(w_4) = 0.3 \times 0.2$

$$P(Q) = \sum_{w \in W_{\mathcal{T}}} P(Q, w) = \sum_{w \in W_{\mathcal{T}}} P(Q|w)P(w) = \sum_{w \in W_{\mathcal{T}}: w \models Q} P(w)$$

- $sneezing(bob)$ is true in 3 worlds
- $P(sneezing(bob)) = 0.7 \times 0.8 + 0.3 \times 0.8 + 0.7 \times 0.2 = 0.94$

Logic Programs with Annotated Disjunctions

`http://cplint.eu/e/sneezing.pl`

$strong\_sneezing(X) : 0.3 \; ; \; moderate\_sneezing(X) : 0.5 \leftarrow flu(X).$
$strong\_sneezing(X) : 0.2 \; ; \; moderate\_sneezing(X) : 0.6 \leftarrow hay\_fever(X).$
$flu(bob).$
$hay\_fever(bob).$

- 9 worlds
- $strong\_sneezing(bob))$ is true in 5
- $P(strong\_sneezing(bob)) =$
  $0.3 \cdot 0.2 + 0.3 \cdot 0.6 + 0.3 \cdot 0.2 + 0.5 \cdot 0.2 + 0.2 \cdot 0.2 = 0.44$

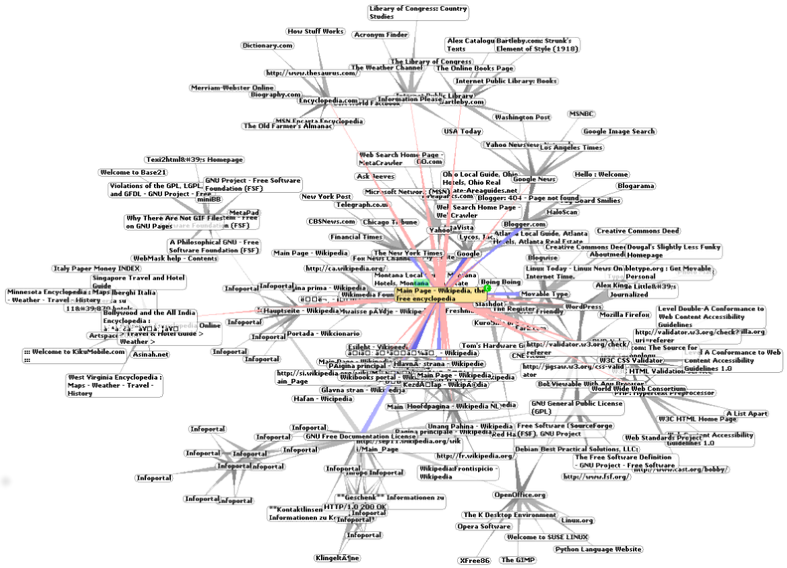
Applications

- Link prediction: given a (social) network, compute the probability of the existence of a link between two entities (UWCSE)

```
advisedby(X, Y) :0.7 :-
  publication(P, X),
  publication(P, Y),
  student(X).
```

Applications

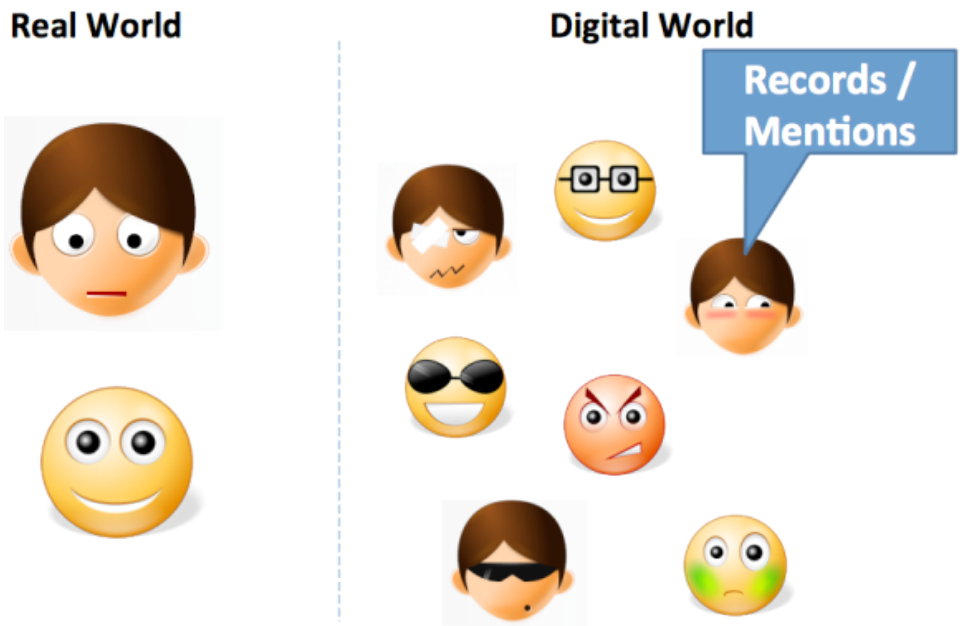
- Classify web pages on the basis of the link structure (WebKB)



```
coursePage(Page1): 0.3 :- linkTo(Page2,Page1),coursePage(Page2).
coursePage(Page1): 0.6 :- linkTo(Page2,Page1),facultyPage(Page2).
...
coursePage(Page): 0.9 :- has('syllabus',Page).
...
```

Applications

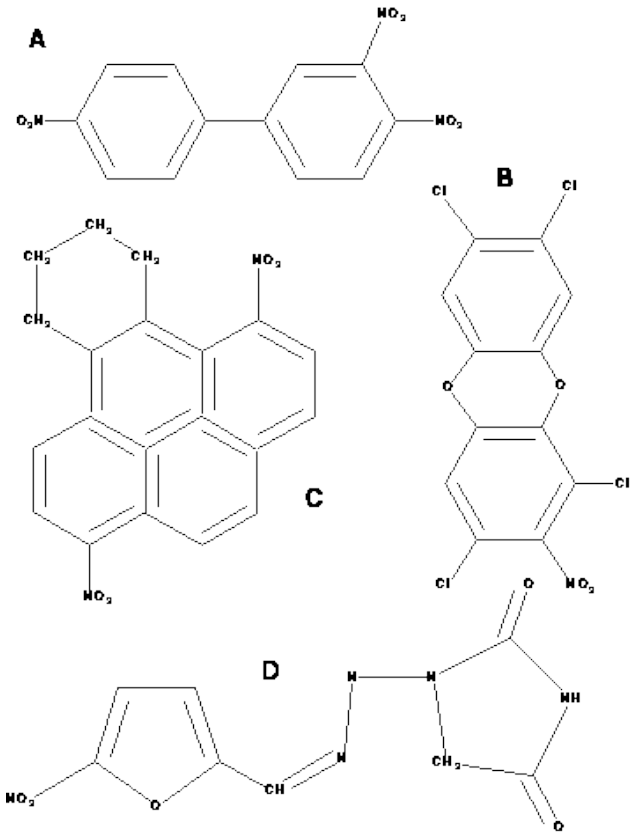
- Entity resolution: identify identical entities in text or databases




```
samebib(A,B):0.9 :-
samebib(A,C), samebib(C,B).
sameauthor(A,B):0.6 :-
  sameauthor(A,C), sameauthor(C,B).
sametitle(A,B):0.7 :-
  sametitle(A,C), sametitle(C,B).
samevenue(A,B):0.65 :-
  samevenue(A,C), samevenue(C,B).
samebib(B,C):0.5 :-
  author(B,D),author(C,E),sameauthor(D,E).
samebib(B,C):0.7 :-
  title(B,D),title(C,E),sametitle(D,E).
samebib(B,C):0.6 :-
  venue(B,D),venue(C,E),samevenue(D,E).
samevenue(B,C):0.3 :-
  haswordvenue(B,logic),
  haswordvenue(C,logic).
...
```

Applications

- Chemistry: given the chemical composition of a substance, predict its mutagenicity or its carcenogenicity




```
active(A):0.4 :-
   atm(A,B,c,29,C),
   gteq(C,-0.003),
   ring_size_5(A,D).
active(A):0.6:-
   lumo(A,B), lteq(B,-2.072).
active(A):0.3 :-
   bond(A,B,C,2),
   bond(A,C,D,1),
   ring_size_5(A,E).
active(A):0.7 :-
   carbon_6_ring(A,B).
active(A):0.8 :-
   anthracene(A,B).
...
```

Applications

- Medicine: diagnose diseases on the basis of patient information (Hepatitis), influence of genes on HIV, risk of falling of elderly people

Inference for PLP under DS

- Computing the probability of a query (no evidence)
- Knowledge compilation:
  - compile the program to an intermediate representation
    - Binary Decision Diagrams (ProbLog [De Raedt et al. IJCAI07], `cplint` [Riguzzi AIIA07,Riguzzi LJIGPL09], PITA [Riguzzi & Swift ICLP10])
    - deterministic, Decomposable Negation Normal Form circuit (d-DNNF) (ProbLog2 [Fierens et al. TPLP15])
    - Sentential Decision Diagrams
  - compute the probability by weighted model counting

Inference for PLP under DS

- Bayesian Network based:
  - Convert to BN
  - Use BN inference algorithms (CVE [Meert et al. ILP09])
- Lifted inference

Examples

Throwing coins `http://cplint.eu/e/coin.swinb`

```
heads(Coin):1/2 ; tails(Coin):1/2 :-
  toss(Coin),\+biased(Coin).
heads(Coin):0.6 ; tails(Coin):0.4 :-
  toss(Coin),biased(Coin).
fair(Coin):0.9 ; biased(Coin):0.1.
toss(coin).
```

Russian roulette with two guns `http://cplint.eu/e/trigger.pl`

```
death:1/6 :- pull_trigger(left_gun).
death:1/6 :- pull_trigger(right_gun).
pull_trigger(left_gun).
pull_trigger(right_gun).
```

Examples

Mendel's inheritance rules for pea plants
`http://cplint.eu/e/mendel.pl`

```
color(X,purple):-cg(X,_A,p).
color(X,white):-cg(X,1,w),cg(X,2,w).
cg(X,1,A):0.5 ; cg(X,1,B):0.5 :-
  mother(Y,X),cg(Y,1,A),cg(Y,2,B).
cg(X,2,A):0.5 ; cg(X,2,B):0.5 :-
  father(Y,X),cg(Y,1,A),cg(Y,2,B).
```

Probability of paths `http://cplint.eu/e/path.swinb`

```
path(X,X).
path(X,Y):-path(X,Z),edge(Z,Y).
edge(a,b):0.3.
edge(b,c):0.2.
edge(a,c):0.6.
```

PLP Online

- `http://cplint.eu`
  - Inference (knwoledge compilation, Monte Carlo)
  - Parameter learning (EMBLEM)
  - Structure learning (SLIPCOVER, LEMUR)

Epidemic Example

If somebody has the flu and the climate is cold, an epidemic arises with 60% probability, a pandemic arises with 30% probability, whereas we have a 10% probability that neither an epidemic nor a pandemic arises. We can write

```
epidemic : 0.6; pandemic : 0.3; null: 0.1 :- flu(_), cold.
```

The null atom can be implicit. Therefore the previous rule, without changing its meaning, can be written

```
epidemic : 0.6; pandemic : 0.3 :- flu(_), cold.
```

Epidemic Example

The weather is cold with a 70% probability. Note that the null atom is implicit here as well.

```
cold : 0.7.
```

David and Robert have the flu for certain:

```
flu(david).
flu(robert).
```

Epidemic Example

The program is almost complete, we need to load the library pita in order to perform exact inference

```
:- use_module(library(pita)).
```

We need to initalize the library with `:-pita` and the program should be enclosed by `:- begin_lpad.` and `:- end_lpad.` (respectively at the begin and at the end of the program).

Epidemic Example

```
:- use_module(library(pita)).
:- pita.
:- begin_lpad.
epidemic : 0.6; pandemic : 0.3 :- flu(_), cold.
cold : 0.7.
flu(david).
flu(robert).
:- end_lpad.
```

How to Execute a Simple Query

To query a program you must use the `prob/2` predicate

```
prob(:Query:atom,-Probability:float).
```

Let us compute the probability that an epidemic arises:

```
prob(epidemic, P).
```

You can ask conditional queries with

```
prob(:Query:atom,:Evidence:atom,-Probability:float).
```

Remember: $P(q|e) = \frac{P(q,e)}{P(e)}$
We can ask for the probability that an epidemic arises given that outside is cold

```
prob(epidemic, cold, P).
```

Monty Hall Puzzle

- A player is given the opportunity to select one of three closed doors, behind one of which there is a prize.
- Behind the other two doors are empty rooms.
- Once the player has made a selection, Monty is obligated to open one of the remaining closed doors which does not contain the prize, showing that the room behind it is empty.
- He then asks the player if he would like to switch his selection to the other unopened door, or stay with his original choice.
- Does it matter if he switches?

Monty Hall Puzzle

```
:- use_module(library(pita)).
:- endif.
:- pita.
:- begin_lpad.
prize(1):1/3; prize(2):1/3; prize(3):1/3.

open_door(2):0.5 ; open_door(3):0.5:- prize(1).
open_door(2):- prize(3).
open_door(3):- prize(2).

win_keep:- prize(1).

win_switch:-
  prize(2),
  open_door(3).

win_switch:-
  prize(3),
  open_door(2).
:- end_lpad.
```

Monty Hall Puzzle

- Queries:

```
prob(win_keep,Prob).
prob(win_switch,Prob).
```

Game of dice

```
on(0,1):1/3 ; on(0,2):1/3 ; on(0,3):1/3.
on(T,1):1/3 ; on(T,2):1/3 ; on(T,3):1/3 :-
  T1 is T-1, T1>=0, on(T1,F), \+ on(T1,3).
```

What is the probability that the die lands on face 1 at time 2?

```
?- prob(on(2,1),Prob).
```

What is the probability that the die lands on face 1 at time 2 given that it landed on face 1 at time 0?

```
?- prob(on(2,1),on(0,1),Prob).
```

Graphics

cplint on SWISH can show the probabilistic results of a query as histograms using predicate `bar/2`

```
bar(+Probability:float,-Chart:dict).
```

A graphical bar chart of the two values will be plotted
Examples:

```
prob(epidemic, P),bar(P,G).
prob(epidemic, cold, P),bar(P,G).
```