Introduction to Probabilistic Logic Programming

Fabrizio Riguzzi

Department of Mathematics and Computer Science – University of Ferrara
fabrizio.riguzzi@unife.it

Dipartimento
di Matematica
e Informatica

Outline

- Logic
- Handling Uncertainty
- Probabilistic logics
- Probabilistic logic programming
- Other fornalisms
- Examples
- Reasoning

Logic

- Useful to model domains with complex relationships among entities
- Various forms:
  - First Order Logic
  - Logic Programming
  - Description Logics

First Order Logic

- Very expressive
- Open World Assumption
- Undecidable

$$\forall x \; Intelligent(x) \rightarrow GoodMarks(x)$$
$$\forall x, y \; Friends(x, y) \rightarrow (Intelligent(x) \leftrightarrow Intelligent(y))$$

Logic Programming

- Closed World Assumption
- Turing complete
- Prolog

$$\begin{array}{l} \mathit{flu}(\mathit{bob}). \\ \mathit{hay\_fever}(\mathit{bob}). \\ \mathit{sneezing}(X) \leftarrow \mathit{flu}(X). \\ \mathit{sneezing}(X) \leftarrow \mathit{hay\_fever}(X). \end{array}$$

Dipartimento
di Matematica
e Informatica

Description Logics

- Subsets of First Order Logic
- Open World Assumption
- Decidable, efficient inference
- Special syntax using concepts (unary predicates) and roles (binary predicates)

$fluffy : Cat$	$cat(fluffy).$
$tom : Cat$	$cat(tom).$
$Cat \sqsubseteq Pet$	$pet(X) \leftarrow cat(X).$
$\exists hasAnimal.Pet \sqsubseteq NatureLover$	$natureLover(X) \leftarrow hasAnimal(X, Y), pet(Y).$
$(kevin, fluffy) : hasAnimal$	$hasAnimal(kevin, fluffy).$
$(kevin, tom) : hasAnimal$	$hasAnimal(kevin, tom).$

Dipartimento di Matematica e Informatica

Uncertainty

- Logic: representing relationships, powerful inference
- but the real world is often uncertain

$$\forall x \; \mathit{Intelligent}(x) \rightarrow \mathit{GoodMarks}(x)$$
$$\forall x, y \; \mathit{Friends}(x, y) \rightarrow (\mathit{Intelligent}(x) \leftrightarrow \mathit{Intelligent}(y))$$

Handling Uncertainty

- Often convenient to describe a domain using a set of random variables.
- Example: home intrusion detection system
  - Earthquake E
  - Burglary B
  - Alarm A
  - Neighbor call N
- Questions:
  - What is the probability of a burglary? (compute $P(\mathrm{B} = \mathrm{t})$, belief computation)
  - What is the probability of a burglary given that the neighbor called? (compute $P(\mathrm{B} = \mathrm{t}|\mathrm{N} = \mathrm{t})$, belief updating)
  - What is the probability of a burglary given that there was an earthquake and the neighbor called? (compute $P(\mathrm{B} = \mathrm{t}|\mathrm{N} = \mathrm{t}, \mathrm{E} = \mathrm{t})$, belief updating)
  - What is the most likely value for burglary given that the neighbor called? ($\arg\max_b P(b|\mathrm{N} = \mathrm{t})$, belief revision)

Dipartimento di Matematica e Informatica

Handling Uncertainty

- When assigning a causal meaning to the variables, the problems are also called
  - Diagnosis: computing $P(cause|symptom)$
  - Prediction: computing $P(symptom|cause)$
  - Classification: computing $\arg\max_{class} P(class|data)$

Handling Uncertainty

- In general, we want to compute

$$P(\mathbf{q}|\mathbf{e})$$

  of a query $\mathbf{q}$ (assignment of values to a set of variables $\mathbf{Q}$) given the evidence $\mathbf{e}$ (assignment of values to a set of variables $\mathbf{E}$).
- Inference.

Dipartimento di Matematica e Informatica

Rules of Probability Theory

- Product rule: $P(\mathrm{a}, \mathrm{b}) = P(\mathrm{a}|\mathrm{b})P(\mathrm{b})$
  - Implies Bayes rule:

$$P(\mathrm{a}|\mathrm{b})P(\mathrm{b}) = P(\mathrm{b}|\mathrm{a})P(\mathrm{a})$$
$$P(\mathrm{a}|\mathrm{b}) = \frac{P(\mathrm{b}|\mathrm{a})P(\mathrm{a})}{P(\mathrm{b})}$$

- Sum rule: $P(\mathrm{a}) = \sum_{\mathrm{b}} P(\mathrm{a}, \mathrm{b})$

Dipartimento di Matematica e Informatica

Handling Uncertainty

- $\mathbf{X}$: set of all variables describing the domain
- Joint probability distribution $P(\mathbf{X})$: $P(\mathbf{x})$ for all $\mathbf{x}$
- We can answer all types of queries using the definition of conditional probability and the sum rule:

$$
\begin{aligned}
P(\mathbf{q}|\mathbf{e}) &= \frac{P(\mathbf{q},\mathbf{e})}{P(\mathbf{e})} = \\
&\frac{\sum_{\mathbf{y},\mathbf{Y}=\mathbf{X}\setminus\mathbf{Q}\setminus\mathbf{E}} P(\mathbf{y},\mathbf{q},\mathbf{e})}{\sum_{\mathbf{z},\mathbf{Z}=\mathbf{X}\setminus\mathbf{E}} P(\mathbf{z},\mathbf{e})}
\end{aligned}
$$

Handling Uncertainty

- If we have n binary variables ($|\mathbf{X}| = n$), knowing the joint probability distribution requires storing $O(2^n)$ different values.
- Even if we had the space, computing $P(\mathbf{q}|\mathbf{e})$ would require $O(2^n)$ operations.

Dipartimento di Matematica e Informatica

Handling Uncertainty

- A value of $\mathbf{X}$ is $(\mathrm{x}_1, \ldots, \mathrm{x}_n)$:

$$
\begin{aligned}
P(\mathbf{x}) \quad = \quad & P(\mathrm{x}_1, \ldots, \mathrm{x}_n) = \\
& P(\mathrm{x}_n|\mathrm{x}_{n-1}, \ldots, \mathrm{x}_1) P(\mathrm{x}_{n-1}, \ldots, \mathrm{x}_1) = \\
& \ldots \\
& P(\mathrm{x}_n|\mathrm{x}_{n-1}, \ldots, \mathrm{x}_1) \ldots P(\mathrm{x}_2|\mathrm{x}_1) P(\mathrm{x}_1) = \\
& \prod_{i=1}^{n} P(\mathrm{x}_i|\mathrm{x}_{i-1}, \ldots, \mathrm{x}_1)
\end{aligned} \tag{1}
$$

  by repeated application of the product rule.
- Chain rule.

Handling Uncertainty

- If, for each variable X_i, $\mathbf{Pa}_i$ is a subset of $\{\mathrm{X}_{i-1}, \ldots, \mathrm{X}_1\}$ such that X_i is conditionally independent of $\{\mathrm{X}_{i-1}, \ldots, \mathrm{X}_1\} \setminus \mathbf{Pa}_i$ given $\mathbf{Pa}_i$, i.e,

$$P(\mathrm{x}_i|\mathrm{x}_{i-1}, \ldots, \mathrm{x}_1) = P(\mathrm{x}_i|\mathbf{pa}_i) \text{ whenever } P(\mathrm{x}_{i-1}, \ldots, \mathrm{x}_1) > 0,$$

then we could write

$$\begin{aligned} P(\mathbf{x}) &= P(\mathrm{x}_1, \ldots, \mathrm{x}_n) = \\ & P(\mathrm{x}_n|\mathrm{x}_{n-1}, \ldots, \mathrm{x}_1) \ldots P(\mathrm{x}_2|\mathrm{x}_1)P(\mathrm{x}_1) = \\ & P(\mathrm{x}_n|\mathbf{pa}_n) \ldots P(\mathrm{x}_2|\mathbf{pa}_1)P(\mathrm{x}_1|\mathbf{pa}_1) = \\ & \prod_{i=1}^{n} P(\mathrm{x}_i|\mathbf{pa}_i) \end{aligned}$$

Dipartimento di Matematica e Informatica

Handling Uncertainty

- $P(\mathrm{x}_i|\mathbf{pa}_i)$: conditional probability table, much smaller than $\{\mathrm{X}_{i-1}, \ldots, \mathrm{X}_1\}$,
- If k is the maximum size of $\mathbf{Pa}_i$, then the storage requirements are $O(n2^k)$ instead of $O(2^n)$.

Dipartimento di Matematica e Informatica

Probabilistic Graphical Models

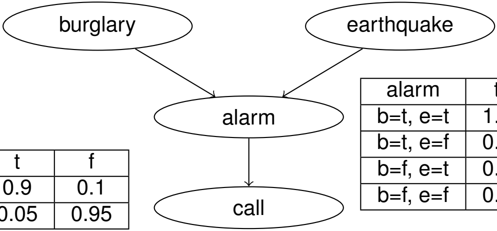
- Taking into account independencies among the variables enables faster inference.
- Graphical models: graph structures that represent independencies.
- Bayesian network [Pearl 88]: directed acyclic graph with a node per variable and an edge from X_j to X_i only if $\mathrm{X}_j \in \mathbf{Pa}_i$.
- A BN together with the set of CPTs $P(\mathrm{x}_i|\mathbf{pa}_i)$ defines a joint probability distribution.

Dipartimento di Matematica e Informatica

Example - Alarm

burg	t	f
	0.1	0.9

earthquake	t	f
	0.2	0.8



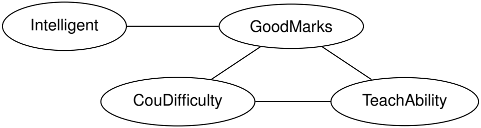
alarm	t	f
b=t, e=t	1.0	0.0
b=t, e=f	0.8	0.2
b=f, e=t	0.8	0.2
b=f, e=f	0.1	0.9

call	t	f
a=t	0.9	0.1
a=f	0.05	0.95

Dipartimento di Matematica e Informatica

Markov Networks

- Undirected graphical models



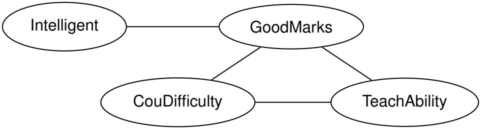
- Each clique in the graph is associated with a potential $\phi_i \geq 0$

$$
\begin{aligned}
P(\mathbf{x}) &= \frac{\prod_i \phi_i(\mathbf{x_i})}{Z} \\
Z &= \sum_{\mathbf{x}} \prod_i \phi_i(\mathbf{x_i})
\end{aligned}
$$

Intelligent	GoodMarks	$\phi_i(I, G)$
false	false	4.5
false	true	4.5
true	false	1.0
true	true	4.5

Dipartimento di Matematica e Informatica

Markov Networks



- If all the potential are strictly positive, we can use a log-linear model (where the f_is are features)

$$P(\mathbf{x}) = \frac{\exp(\sum_i w_i f_i(\mathbf{x_i}))}{Z}$$

$$Z = \sum_{\mathbf{x}} \exp(\sum_i w_i f_i(\mathbf{x_i})))$$

$$f_i(Intelligent, GoodMarks) = \begin{cases} 1 & \text{if } \neg\text{Intelligent}\vee\text{GoodMarks} \\ 0 & \text{otherwise} \end{cases}$$

$$w_i = 1.5$$

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

Dipartimento di Matematica e Informatica

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

Dipartimento di Matematica e Informatica

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

Dipartimento di Matematica e Informatica

PLP Online

- `http://cplint.eu`
  - Inference (knwoledge compilation, Monte Carlo)
  - Parameter learning (EMBLEM)
  - Structure learning (SLIPCOVER)
- `https://dtai.cs.kuleuven.be/problog/`
  - Inference (knwoledge compilation, Monte Carlo)
  - Parameter learning (LFI-ProbLog)

PRISM

$sneezing(X) \leftarrow flu(X), msw(flu\_sneezing(X), 1).$
$sneezing(X) \leftarrow hay\_fever(X), msw(hay\_fever\_sneezing(X), 1).$
$flu(bob).$
$hay\_fever(bob).$

$values(flu\_sneezing(\_X), [1, 0]).$
$values(hay\_fever\_sneezing(\_X), [1, 0]).$
$: -set\_sw(flu\_sneezing(\_X), [0.7, 0.3]).$
$: -set\_sw(hay\_fever\_sneezing(\_X), [0.8, 0.2]).$

- Distributions over *msw* facts (random switches)
- Worlds obtained by selecting one value for every grounding of each *msw* statement

Dipartimento di Matematica e Informatica

Logic Programs with Annotated Disjunctions

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

$$
\begin{array}{l}
sneezing(X) : 0.7 \ ; null : 0.3 \leftarrow flu(X). \\
sneezing(X) : 0.8 \ ; null : 0.2 \leftarrow hay\_fever(X). \\
flu(bob). \\
hay\_fever(bob).
\end{array}
$$

- 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

Dipartimento di Matematica e Informatica

ProbLog

$$
\begin{array}{l}
\mathit{sneezing}(X) \leftarrow \mathit{flu}(X), \mathit{flu\_sneezing}(X).\\
\mathit{sneezing}(X) \leftarrow \mathit{hay\_fever}(X), \mathit{hay\_fever\_sneezing}(X).\\
\mathit{flu}(\mathit{bob}).\\
\mathit{hay\_fever}(\mathit{bob}).\\
0.7 :: \mathit{flu\_sneezing}(X).\\
0.8 :: \mathit{hay\_fever\_sneezing}(X).
\end{array}
$$

- Distributions over facts
- Worlds obtained by selecting or not every grounding of each probabilistic fact

Dipartimento di Matematica e Informatica

Distribution Semantics

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

Dipartimento di Matematica e Informatica

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: $\sum_{w\in W_T} P(w) = 1$

Dipartimento di Matematica e Informatica

Distribution Semantics

- Ground query Q
- $P(Q|w) = 1$ if Q is true in w and 0 otherwise
- $P(Q) = \sum_w P(Q, w) = \sum_w P(Q|w)P(w) = \sum_{w \models Q} P(w)$
- You can see $P(Q)$ as the probability that Q is true in a world sampled at random from $P(w)$
  - for each choice, sample a value to get a world
  - test the query in the world

Dipartimento di Matematica e Informatica

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
- $P(sneezing(bob)) = 0.7 \times 0.8 + 0.3 \times 0.8 + 0.7 \times 0.2 = 0.94$

Example Program (ProbLog) Worlds

- 4 worlds

$sneezing(X) \leftarrow flu(X), flu\_sneezing(X).$
$sneezing(X) \leftarrow hay\_fever(X), hay\_fever\_sneezing(X).$
$flu(bob).$
$hay\_fever(bob).$

$flu\_sneezing(bob).$ $hay\_fever\_sneezing(bob).$ $P(w_1) = 0.7 \times 0.8$	$hay\_fever\_sneezing(bob).$ $P(w_2) = 0.3 \times 0.8$
$flu\_sneezing(bob).$ $P(w_3) = 0.7 \times 0.2$	$P(w_4) = 0.3 \times 0.2$

- $sneezing(bob)$ 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

$$
\begin{array}{l}
strong\_sneezing(X) : 0.3 \; ; \; moderate\_sneezing(X) : 0.5 \leftarrow flu(X). \\
strong\_sneezing(X) : 0.2 \; ; \; moderate\_sneezing(X) : 0.6 \leftarrow hay\_fever(X). \\
flu(bob). \\
hay\_fever(bob).
\end{array}
$$

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

Dipartimento di Matematica e Informatica

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

http://cplint.eu/e/monty.swinb

```
:- 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.
```

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).
```

Examples

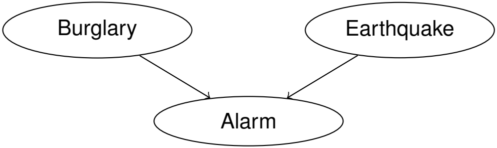
Mendel's inheritance rules for pea plants
http://cplint.eu/e/mendel.pl

```
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.
```

Encoding Bayesian Networks



alarm	t	f
b=t,e=t	1.0	0.0
b=t,e=f	0.8	0.2
b=f,e=t	0.8	0.2
b=f,e=f	0.1	0.9

burg	t	f
	0.1	0.9

earthq	t	f
	0.2	0.8

http://cplint.eu/e/alarm.pl

```
burg(t):0.1 ; burg(f):0.9.
earthq(t):0.2 ; earthq(f):0.8.
alarm(t):-burg(t),earthq(t).
alarm(t):0.8 ; alarm(f):0.2:-burg(t),earthq(f).
alarm(t):0.8 ; alarm(f):0.2:-burg(f),earthq(t).
alarm(t):0.1 ; alarm(f):0.9:-burg(f),earthq(f).
```

Expressive Power

- All languages under the distribution semantics have the same expressive power
- LPADs have the most general syntax
- There are transformations that can convert each one into the others
- PRISM, ProbLog to LPAD: direct mapping

Dipartimento di Matematica e Informatica

LPADs to ProbLog

- Clause C_i with variables $\overline{X}$

$$H_1 : p_1 \vee \ldots \vee H_n : p_n \leftarrow B.$$

is translated into

$$\begin{array}{l}
H_1 \leftarrow B, f_{i,1}(\overline{X}). \\
H_2 \leftarrow B, not(f_{i,1}(\overline{X})), f_{i,2}(\overline{X}). \\
\vdots \\
H_n \leftarrow B, not(f_{i,1}(\overline{X})), \ldots, not(f_{i,n-1}(\overline{X})). \\
\pi_1 :: f_{i,1}(\overline{X}). \\
\vdots \\
\pi_{n-1} :: f_{i,n-1}(\overline{X}).
\end{array}$$

where $\pi_1 = p_1$, $\pi_2 = \frac{p_2}{1-\pi_1}$, $\pi_3 = \frac{p_3}{(1-\pi_1)(1-\pi_2)}, \ldots$

- In general $\pi_i = \frac{p_i}{\prod_{j=1}^{i-1}(1-\pi_j)}$

Dipartimento di Matematica e Informatica

Conversion to Bayesian Networks

- PLP can be converted to Bayesian networks
- Conversion for an LPAD T
- For each ground atom A a binary variable A
- For each clause C_i in the grounding of T

$$H_1 : p_1 \vee \ldots \vee H_n : p_n \leftarrow B_1, \ldots B_m, \neg C_1, \ldots, \neg C_l$$

a variable CH_i with $B_1, \ldots, B_m, C_1, \ldots, C_l$ as parents and H_1, ..., H_n and *null* as values

Conversion to Bayesian Networks

$$H_1 : p_1 \vee \ldots \vee H_n : p_n \leftarrow B_1, \ldots B_m, \neg C_1, \ldots, \neg C_l$$

- The CPT of CH_i is

	$\ldots$	$B_1 = 1, \ldots, B_m = 1, C_1 = 0, \ldots, C_l = 0$	$\ldots$
$CH_i = H_1$	0.0	p_1	0.0
$\ldots$			
$CH_i = H_n$	0.0	p_n	0.0
$CH_i = null$	1.0	$1 - \sum_{i=1}^{n} p_i$	1.0

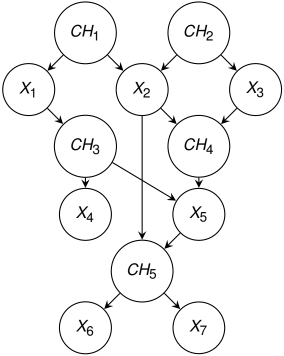
Conversion to Bayesian Networks

- Each variable A corresponding to atom A has as parents all the variables CH_i of clauses C_i that have A in the head.
- The CPT for A is:

	at least one parent = A	remaining cols
$A = 1$	1.0	0.0
$A = 0$	0.0	1.0

Conversion to Bayesian Networks

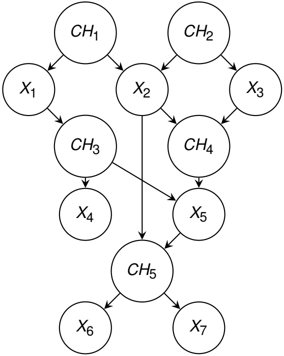
$$
\begin{aligned}
C_1 &= x1 : 0.4 \vee x2 : 0.6. \\
C_2 &= x2 : 0.1 \vee x3 : 0.9. \\
C_3 &= x4 : 0.6 \vee x5 : 0.4 \leftarrow x1. \\
C_4 &= x5 : 0.4 \leftarrow x2, x3. \\
C_5 &= x6 : 0.3 \vee x7 : 0.2 \leftarrow x2, x5.
\end{aligned}
$$



Conversion to Bayesian Networks

CH_1, CH_2	$x1, x2$	$x1, x3$	$x2, x2$	$x2, x3$
$x2 = 1$	1.0	0.0	1.0	1.0
$x2 = 0$	0.0	1.0	0.0	0.0

$x2, x5$	1,1	1,0	0,1	0,0
$CH_5 = x6$	0.3	0.0	0.0	0.0
$CH_5 = x7$	0.2	0.0	0.0	0.0
$CH_5 = null$	0.5	1.0	1.0	1.0



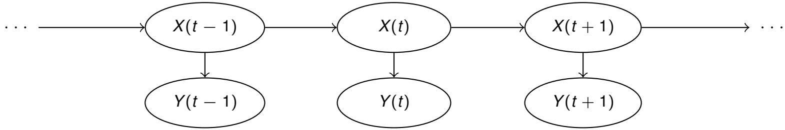
Function Symbols

- What if function symbols are present?
- Infinite, countable Herbrand universe
- Infinite, countable Herbrand base
- Infinite, countable grounding of the program T
- Uncountable W_T
- Each world infinite, countable
- $P(w) = 0$
- Semantics not well-defined

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).
```

Hidden Markov Models




```
hmm(S,O):-hmm(q1,[],S,O).
hmm(end,S,S,[]).
hmm(Q,S0,S,[L|O]):-
  Q\= end,
  next_state(Q,Q1,S0),
  letter(Q,L,S0),
  hmm(Q1,[Q|S0],S,O).
next_state(q1,q1,_S):1/3;next_state(q1,q2_,_S):1/3;
  next_state(q1,end,_S):1/3.
next_state(q2,q1,_S):1/3;next_state(q2,q2,_S):1/3;
  next_state(q2,end,_S):1/3.
letter(q1,a,_S):0.25;letter(q1,c,_S):0.25;
  letter(q1,g,_S):0.25;letter(q1,t,_S):0.25.
letter(q2,a,_S):0.25;letter(q2,c,_S):0.25;
  letter(q2,g,_S):0.25;letter(q2,t,_S):0.25.
```

Dipartimento di Matematica e Informatica

Hybrid Programs

- Up to now only discrete random variables and discrete probability distributions.
- Hybrid Probabilistic Logic Programs: some of the random variables are continuous.
- cplint allows the specification of density functions over arguments of atoms in the head of rules

- A probability density on an argument `Var` of an atom `A` is specified with

  ```
  A : Density :- Body.
  ```

  where `Density` is a special atom
  - `uniform(Var,L,U)`: `Var` is uniformly distributed in $[L, U]$
  - `gaussian(Var,Mean,Variance)`: Gaussian distribution
  - `dirichlet(Var,Par)`: Dirichlet distribution with parameters α specified by the list `Par`
  - `gamma(Var,Shape,Scale)`: gamma distribution
  - `beta(Var,Alpha,Beta)`: beta distribution
  - + others (see the manual)

Dipartimento di Matematica e Informatica

Hybrid Programs

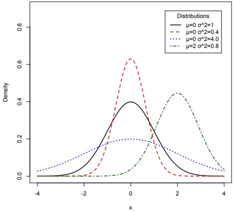
- Also discrete distributions, with either a finite or countably infinite support:
  - `discrete(Var,D)` or `finite(Var,D)`: `D` is a list of couples `Value:Prob` assigning probability `Prob` to `Value`
  - `uniform(Var,D)`: `D` is a list of values each taking the same probability (1 over the length of `D`).
  - `poisson(Var,Lambda)`: Poisson distribution

Semantics

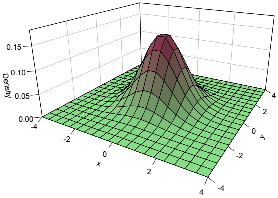
- For each random variable, sample a value, obtaining a world
- Test Q in the world
- $P(Q)$ is the probability that Q is true in the world

Examples

```
g(X) : gaussian(X,0,1).
```




```
g(X) : gaussian(X,[0,0],[ [1,0],[0,1] ]).
```



Gaussian Mixture Example

- `http://cplint.eu/e/gaussian_mixture.pl` defines a mixture of two Gaussians:

```
heads:0.6;tails:0.4.
g(X): gaussian(X,0, 1).
h(X): gaussian(X,5, 2).
mix(X) :- heads, g(X).
mix(X) :- tails, h(X).
```

- The argument `X` of `mix(X)` follows a distribution that is a mixture of two Gaussian, one with mean 0 and variance 1 with probability 0.6 and one with mean 5 and variance 2 with probability 0.4.

Description Logics

- DISPONTE: "DIstribution Semantics for Probabilistic ONTologiEs" [Riguzzi et al. SWJ15]
- Probabilistic axioms:
  - $p :: E$
    e.g., $p :: C \sqsubseteq D$ represents the fact that we believe in the truth of $C \sqsubseteq D$ with probability p.
- DISPONTE applies the distribution semantics of probabilistic logic programming to description logics

DISPONTE

- World w: regular DL KB obtained by selecting or not the probabilistic axioms
- Probability of a query Q given a world w: $P(Q|w) = 1$ if $w \models Q$, 0 otherwise
- Probability of Q
  $P(Q) = \sum_w P(Q, w) = \sum_w P(Q|w)P(w) = \sum_{w:w\models Q} P(w)$

Dipartimento di Matematica e Informatica

Example

$0.4 :: \mathit{fluffy} : \mathit{Cat}$
$0.3 :: \mathit{tom} : \mathit{Cat}$
$0.6 :: \mathit{Cat} \sqsubseteq \mathit{Pet}$
$\exists \mathit{hasAnimal}.\mathit{Pet} \sqsubseteq \mathit{NatureLover}$
$(\mathit{kevin}, \mathit{fluffy}) : \mathit{hasAnimal}$
$(\mathit{kevin}, \mathit{tom}) : \mathit{hasAnimal}$

- $P(\mathit{kevin} : \mathit{NatureLover}) =$
  $0.4 \times 0.3 \times 0.6 + 0.4 \times 0.7 \times 0.6 + 0.6 \times 0.3 \times 0.6 = 0.348$

Knowledge-Based Model Construction

- The probabilistic logic theory is used directly as a template for generating an underlying complex graphical model [Breese et al. TSMC94].
- Languages: CLP(BN), Markov Logic

CLP(BN) [Costa UAI02]

- Variables in a CLP(BN) program can be random
- Their values, parents and CPTs are defined with the program
- To answer a query with uninstantiated random variables, CLP(BN) builds a BN and performs inference
- The answer will be a probability distribution for the variables
- Probabilistic dependencies expressed by means of CLP constraints

```
{ Var = Function with p(Values, Dist) }
{ Var = Function with p(Values, Dist, Parents) }
```

CLP(BN)

```
....
course_difficulty(Key, Dif) :-
{ Dif = difficulty(Key) with p([h,m,l],
[0.25, 0.50, 0.25]) }.
student_intelligence(Key, Int) :-
{ Int = intelligence(Key) with p([h, m, l],
[0.5,0.4,0.1]) }.
....
registration(r0,c16,s0).
registration(r1,c10,s0).
registration(r2,c57,s0).
registration(r3,c22,s1).
```

Dipartimento di Matematica e Informatica

CLP(BN)

```
....
registration_grade(Key, Grade):-
registration(Key, CKey, SKey),
course_difficulty(CKey, Dif),
student_intelligence(SKey, Int),
{ Grade = grade(Key) with
 p([a,b,c,d],
%h h  h m  h l  m h  m m  m l  l h  l m  l l
[0.20,0.70,0.85,0.10,0.20,0.50,0.01,0.05,0.10,
 0.60,0.25,0.12,0.30,0.60,0.35,0.04,0.15,0.40,
 0.15,0.04,0.02,0.40,0.15,0.12,0.50,0.60,0.40,
 0.05,0.01,0.01,0.20,0.05,0.03,0.45,0.20,0.10 ],
 [Int,Dif]))
}.
.....
```

Dipartimento di Matematica e Informatica

CLP(BN)

```
?- [school_32].
   ?- registration_grade(r0,G).
p(G=a)=0.4115,
p(G=b)=0.356,
p(G=c)=0.16575,
p(G=d)=0.06675 ?
?- registration_grade(r0,G),
   student_intelligence(s0,h).
p(G=a)=0.6125,
p(G=b)=0.305,
p(G=c)=0.0625,
p(G=d)=0.02 ?
```

Dipartimento di Matematica e Informatica

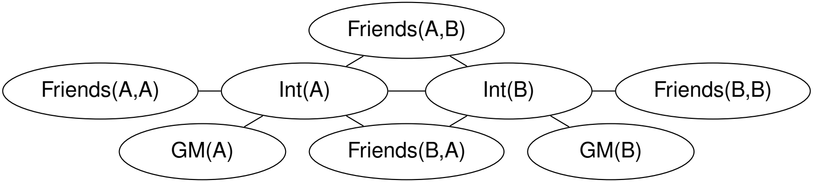
Markov Logic

- A Markov Logic Network (MLN) [Richardson, Domingos ML06] 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 Example

$$1.5 \quad \forall x \; Intelligent(x) \rightarrow GoodMarks(x)$$
$$1.1 \quad \forall x, y \; Friends(x, y) \rightarrow (Intelligent(x) \leftrightarrow Intelligent(y))$$

- Constants Anna (A) and Bob (B)



Dipartimento
di Matematica
e Informatica

Markov Networks

- Probability of an interpretation $\mathbf{x}$

$$P(\mathbf{x}) = \frac{\exp(\sum_i w_i n_i(\mathbf{x_i}))}{Z}$$

- $n_i(\mathbf{x_i}) =$ number of true groundings of formula F_i in $\mathbf{x}$
- Typed variables and constants greatly reduce size of ground Markov net

Reasoning Tasks

- Inference: we want to compute the probability of a query given the model and, possibly, some evidence, or find assignments of the random variables with the highest probability
- Weight learning: we know the structural part of the model (the logic formulas) but not the numeric part (the weights) and we want to infer the weights from data
- Structure learning we want to infer both the structure and the weights of the model from data

Inference for PLP under DS

- EVID: compute an unconditional probability $P(e)$, the probability of evidence (also query in this case).
- COND: compute the conditional probability distribution of the query given the evidence, i.e. compute $P(q|e)$
- MPE or *most probable explanation*: find the most likely value of all non-evidence atoms given the evidence, i.e. solving the optimization problem $\arg\max_q P(q|e)$
- MAP or *maximum a posteriori*: find the most likely value of a set of non-evidence atoms given the evidence, i.e. finding $\arg\max_q P(q|e)$. MPE is a special case of MAP where $Q \cup E = H_T$.
- DISTR: compute the probability distribution or density of the non-ground arguments of a conjunction of literals q, e.g., computing the probability density of X in goal $mix(X)$ of the Gaussian mixture

Weight Learning

- Given
  - model: a probabilistic logic model with unknown parameters
  - data: a set of interpretations
- Find the values of the parameters that maximize the probability of the data given the model
- Discriminative learning: maximize the conditional probability of a set of outputs (e.g. ground instances for a predicate) given a set of inputs
- Alternatively, the data are queries for which we know the probability: minimize the error in the probability of the queries that is returned by the model

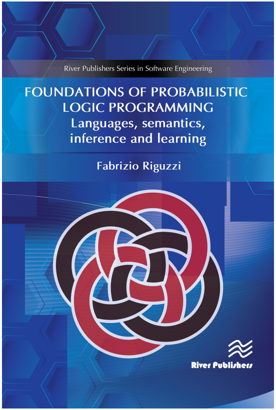
Structure Learning

- Given
  - language bias: a specification of the search space
  - data: a set of interpretations
- Find the formulas and the parameters that maximize the likelihood of the data given the model
- Discriminative learning: again maximize the conditional likelihood of a set of outputs given a set of inputs

Dipartimento
di Matematica
e Informatica

Conclusions

- Handling relationships
- Handling uncertainty
- Open problems
  - Semantics for hybrid programs with function symbols
  - Learning hybrid progams



Dipartimento di Matematica e Informatica

Resources

- Online course on cplint
  - Moodle https://edu.swi-prolog.org/
  - Videos of lectures https://www.youtube.com/playlist?list=PLJPXEH0boeND0UGWJxBRWs7qzzKpC-FkN
- ACAI summer school on Statistical Relational AI http://acai2018.unife.it/
- Videos of lectures https://www.youtube.com/playlist?list=PLJPXEH0boeNDWTNwWTWnVffXi5XwAj1mb
- Videos of lecture Probabilistic Inductive Logic Programming
  - Part 1 https://youtu.be/mLdPGSlgNxU
  - Part 2 https://youtu.be/DRlOft0Y_Ng
- cplint in Playing with Prolog https://www.youtube.com/playlist?list=PLJPXEH0boeNAik6QnfvGlAGRQxFY_LCE3

Dipartimento di Matematica e Informatica

THANKS FOR LISTENING

AND

ANY QUESTIONS ?

References

- Breese, J. S., Goldman, R. P., and Wellman, M. P. (1994). Introduction to the special section on knowledge-based construction of probabilistic and decision models. IEEE Transactions On Systems, Man and Cybernetics, 24(11):1577-1579.
- Dantsin, E. (1991). Probabilistic logic programs and their semantics. In Russian Conference on Logic Programming, volume 592 of LNCS, pages 152-164. Springer.
- De Raedt, L., Kimmig, A., and Toivonen, H. (2007). Problog: A probabilistic prolog and its application in link discovery. In International Joint Conference on Artificial Intelligence, pages 2462-2467.

Dipartimento di Matematica e Informatica

References

- Poole, D. (1993). Logic programming, abduction and probability - a top-down anytime algorithm for estimating prior and posterior probabilities. New Gener. Comput., 11(3):377-400.
- Poole, D. (1997). The Independent Choice Logic for modelling multiple agents under uncertainty. Artif. Intell., 94(1-2):7-56.
- Riguzzi, F. (2009). Extended semantics and inference for the Independent Choice Logic. Logic Journal of the IGPL.

References

- Sato, T. (1995). A statistical learning method for logic programs with distribution semantics. In International Conference on Logic Programming, pages 715-729.
- Vennekens, J., Verbaeten, S., and Bruynooghe, M. (2004). Logic programs with annotated disjunctions. In International Conference on Logic Programming, volume 3131 of LNCS, pages 195-209. Springer.
- Breese, J. S., Goldman, R. P., and Wellman, M. P. (1994). Introduction to the special section on knowledge-based construction of probabilistic and decision models. IEEE Transactions On Systems, Man and Cybernetics, 24(11):1577-1579.

References

- Costa, Vitor Santos, et al. CLP(BN): Constraint logic programming for probabilistic knowledge. Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., 2002.
- Richardson, Matthew, and Pedro Domingos. Markov logic networks. Machine learning 62.1-2 (2006): 107-136.
- Riguzzi, Fabrizio, et al. "Probabilistic description logics under the distribution semantics." Semantic Web 6.5 (2015): 477-501.

Dipartimento di Matematica e Informatica