Probabilistic Logics

Fabrizio Riguzzi

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

Dipartimento
di Matematica
e Informatica

Outline

- Logic
- Logic programming
- Description Logics
- Probabilistic logic programming
- Reasoning with PLP
- Probabilistic Description Logics
- Reasoning with PDL

Logic

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

- Very expressive
- Open World Assumption
- Undecidable

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

- Closed World Assumption
- Turing complete
- Prolog

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

Description Logics

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

$$
\begin{aligned}
&\mathit{fluffy} : \mathit{Cat} \\
&\mathit{tom} : \mathit{Cat} \\
&\mathit{Cat} \sqsubseteq \mathit{Pet} \\
&\exists \mathit{hasAnimal}.\mathit{Pet} \sqsubseteq \mathit{NatureLover} \\
&(\mathit{kevin}, \mathit{fluffy}) : \mathit{hasAnimal} \\
&(\mathit{kevin}, \mathit{tom}) : \mathit{hasAnimal}
\end{aligned}
$$

Dipartimento
di Matematica
e Informatica

Logic Programming

- A subset of FOL + inductive definitions [Denecker, Vennekens KR14]
- A subset of FOL + Closed World Assumption (CWA)
- Transitive closure:

$$\begin{array}{l} path(X, X). \\ path(X, Y) \leftarrow edge(X, Z), path(Z, Y). \\ edge(a, b). \\ edge(b, c). \\ edge(a, c). \end{array}$$

Dipartimento di Matematica e Informatica

Semantics of Logic Programming

- Positive programs: Least Herbrand Model
- Normal programs (including negation)

$$
\begin{array}{l}
\mathit{ends}(X, Y) \leftarrow \mathit{path}(X, Y), \sim\mathit{source}(Y). \\
\mathit{source}(X) \leftarrow \mathit{edge}(X, Y).
\end{array}
$$

$\mathit{ends}(X, Y)$ is true if there is a path from X to Y and Y is a terminal node, i.e., it has no outgoing edges.

- Clark's completion (Prolog)
- Stable models (Answer Set Programming)
- Well founded (Prolog+Tabling)

- Special syntax using concepts (unary predicates) and roles (binary predicates)
- Translation into FOL

$$
\begin{array}{rcl}
\pi_x(A) & = & A(x) \\
\pi_x(\neg C) & = & \neg \pi_x(C) \\
\pi_x(\{a\}) & = & (x = a) \\
\pi_x(C \sqcap D) & = & \pi_x(C) \wedge \pi_x(D) \\
\pi_x(C \sqcup D) & = & \pi_x(C) \wedge \pi_x(D) \\
\pi_x(\exists R.C) & = & \exists y \; R(x,y) \wedge \pi_y(C) \\
\pi_x(\forall R.C) & = & \forall y \; R(x,y) \rightarrow \pi_y(C)
\end{array}
$$

and π_y is obtained from π_x by replacing x with y and vice-versa.

Axiom	Translation
$C \sqsubseteq D$	$\forall x \; \pi_x(C) \rightarrow \pi_x(D)$
$a : C$	$C(a)$
$(a, b) : R$	$R(a, b)$

$fluffy : Cat$
$tom : Cat$
$Cat \sqsubseteq Pet$
$\exists hasAnimal.Pet \sqsubseteq NatureLover$
$(kevin, fluffy) : hasAnimal$
$(kevin, tom) : hasAnimal$

$cat(fluffy).$
$cat(tom).$
$\forall X \; pet(X) \leftarrow cat(X).$
$\forall X, Y \; natureLover(X) \leftarrow hasAnimal(X, Y), pet(Y).$
$hasAnimal(kevin, fluffy).$
$hasAnimal(kevin, tom).$

Dipartimento di Matematica e Informatica

- $KB = \langle ABox, TBox, RBox \rangle$

$$princeJohn : Tyrant$$
$$nottinghamSheriff : Tyrant$$
$$Tyrant \sqsubseteq RichPerson$$
$$\exists hasStolenFrom.RichPerson \sqsubseteq GoodThief$$
$$GoodThief \sqsubseteq \exists hasGivenTo.Needy \quad \forall X \; GoodThief(X) \rightarrow \exists Y \; hasGivenTo(X, Y), Needy(Y)$$
$$(robinHood, princeJohn) : hasStolenFrom$$
$$(robinHood, nottinghamSheriff) : hasStolenFrom$$
$$hasGifted \sqsubseteq hasGivenTo$$

existential quantification in consequent

Combining Logic and Probability

- Logic does not handle well uncertainty
- Probability Theory/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

- 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

- 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

- 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

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

PLP Online

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

Logic Programs with Annotated Disjunctions

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

- Distributions over the head of rules
- Worlds obtained by selecting one atom from the head of every grounding of each clause

ProbLog

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

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

Distribution Semantics

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

$$P(\kappa) = \prod_{(C_i,\theta,k)\in\kappa} \Pi_{i,k}$$

Dipartimento
di Matematica
e Informatica

Distribution Semantics

- Selection σ: a total composite choice (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 $P(w_\sigma) = P(\sigma) = \prod_{(C_i,\theta,k)\in\sigma} \Pi_{i,k}$
- 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
- $P(Q) = \sum_w P(Q, w) = \sum_w P(Q|w)P(w) = \sum_{w \models Q} P(w)$

Dipartimento di Matematica e Informatica

Example Program (LPAD) Worlds

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

$$
\begin{array}{ll}
sneezing(bob) \leftarrow flu(bob). & null \leftarrow flu(bob). \\
sneezing(bob) \leftarrow hay\_fever(bob). & sneezing(bob) \leftarrow hay\_fever(bob). \\
flu(bob). & flu(bob). \\
hay\_fever(bob). & hay\_fever(bob). \\
P(w_1) = 0.7 \times 0.8 & P(w_2) = 0.3 \times 0.8 \\
\\
sneezing(bob) \leftarrow flu(bob). & null \leftarrow flu(bob). \\
null \leftarrow hay\_fever(bob). & null \leftarrow hay\_fever(bob). \\
flu(bob). & flu(bob). \\
hay\_fever(bob). & hay\_fever(bob). \\
P(w_3) = 0.7 \times 0.2 & P(w_4) = 0.3 \times 0.2
\end{array}
$$

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

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

Dipartimento di Matematica e Informatica

Example Program (ProbLog) Worlds

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

- 4 worlds

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

$$
\begin{array}{ll}
flu\_sneezing(bob). & \\
hay\_fever\_sneezing(bob). & hay\_fever\_sneezing(bob).\\
P(w_1) = 0.7 \times 0.8 & P(w_2) = 0.3 \times 0.8\\
flu\_sneezing(bob). & \\
P(w_3) = 0.7 \times 0.2 & P(w_4) = 0.3 \times 0.2
\end{array}
$$

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

Dipartimento di Matematica e Informatica

Logic Programs with Annotated Disjunctions

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

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

- 9 worlds
- $P(strong\_sneezing(bob)) = 0.3 \times 0.2 + 0.3 \times 0.6 + 0.3 \times 0.2 + 0.5 \times 0.2 + 0.2 \times 0.2 = 0.44$

Dipartimento di Matematica e Informatica

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

Dipartimento
di Matematica
e Informatica

Reasoning Tasks

- Inference: we want to compute the probability of a query given the model and, possibly, some evidence
- 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

- Computing the probability of a query (no evidence)
- Knowledge compilation:
  - compile the program to an intermediate representation
    - Binary Decision Diagrams (BDD) (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 (ProbLog2 [Fierens et al. TPLP15])
  - compute the probability by weighted model counting

Dipartimento di Matematica e Informatica

- Assign Boolean random variables to the probabilistic rules
- Given a query Q, compute its explanations, assignments to the random variables that are sufficient for entailing the query
- Let K be the set of all possible explanations
- Build a Boolean formula $F(Q)$
- Transform it into an intermediate representation: BDD, d-DNNF, SDD
- Perform Weighted Model Counting (WMC)

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).
```

Dipartimento di Matematica e Informatica

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.
```

Applications

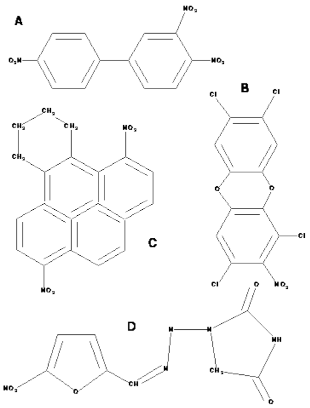
- Link prediction: given a (social) network, compute the probability of the existence of a link between two entities (UWCSE)

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

Dipartimento di Matematica e Informatica

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).
...
```

DISPONTE: DIstribution Semantics for Probabilistic ONTologiEs

- Idea: annotate axioms of an ontology with a probability, under the assumption that the axioms are independent of each other

$$0.6 :: Cat \sqsubseteq Pet$$

  - The probability value specifies a degree of belief in the truth of the corresponding axiom.

- DISPONTE applies the distribution semantics of probabilistic logic programming to description logics
- A probabilistic ontology defines thus a distribution over theories (worlds) obtained by including an axiom in a world with the probability given by the annotation

DISPONTE

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

Dipartimento di Matematica e Informatica

Example

- *fluffy* and *Tom* are *Cats* and *Cats* are *Pets*. Everyone who has a pet animal ($\exists hasAnimal.Pet$) is a *NatureLover*; *donVito* has two animals, *fluffy* and *tom* with probability 0.4 and 0.9 respectively. *NatureLovers* are *GoodPersons* with probability 0.2. `http://trill-sw.eu/p/don_vito.pl`

$$\begin{array}{lll} & & fluffy : Cat \\ & & tom : Cat \\ & & Cat \sqsubseteq Pet \\ & & \exists hasAnimal.Pet \sqsubseteq NatureLover \\ 0.4 & :: & (donVito, fluffy) : hasAnimal \quad (1) \\ 0.9 & :: & (donVito, tom) : hasAnimal \quad (2) \\ 0.2 & :: & NatureLover \sqsubseteq GoodPerson \quad (3) \end{array}$$

- $Q = donVito : GoodPerson$, 8 worlds (Q true in 3 of them):

$$\{ (1), (3) \}, \{ (2), (3) \}, \{ (1), (2), (3) \}$$

and $P(Q) = 0.4 \times (1 - 0.9) \times 0.2 + (1 - 0.4) \times 0.9 \times 0.2$
$+ 0.4 \times 0.9 \times 0.2 = 0.188$

Dipartimento di Matematica e Informatica

Inference in DISPONTE

- The probability of a query Q can be computed first finding the explanation or Q in the knowledge base
  - An explanation of query Q is a subset of logical axioms of a KB sufficient to entail Q.
- **Problem: In general, justifications are not mutually exclusive!**
  - Encode the justifications into a DNF Boolean formula and use a Binary Decision Diagram (BDD) to make the disjuncts mutually exclusive.

Example of DISPONTE Explanations

The Godfather KB

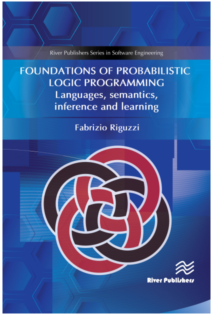
$$
\begin{array}{rcl}
C_1 & = & fluffy : Cat \\
C_2 & = & tom : Cat \\
C_3 & = & Cat \sqsubseteq Pet \\
C_4 & = & \exists hasAnimal.Pet \sqsubseteq NatureLover \\
E_1 & = & 0.4 :: (donVito, fluffy) : hasAnimal \\
E_2 & = & 0.9 :: (donVito, tom) : hasAnimal \\
E_3 & = & 0.2 :: NatureLover \sqsubseteq GoodPerson
\end{array}
$$

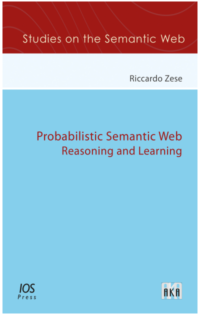
If we forget about the probabilities, the query $Q = donVito : GoodPerson$ has 2 justifications

$$\mathcal{J} = \{\ \{E_1, C_1, C_3, C_4, E_3\},\ \{E_2, C_2, C_3, C_4, E_3\}\ \}$$

Conclusions

- Logics
- Probabilistic Logics
- Inference
- Open problems
  - Programs with continuous variables
  - Combining Deep Learning with PILP





THANKS FOR LISTENING

AND

ANY QUESTIONS ?

Dipartimento
di Matematica
e Informatica