PLP: a brick to build your own starship

Riccardo Zese

Dipartimento di Ingegneria – University of Ferrara
riccardo.zese@unife.it

Università
degli Studi
di Ferrara

Outline

- Logic
- Probabilistic logics
- Probabilistic logic programming
- Probabilistic description logics
- PLP and Probabilistic DL

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

Università degli Studi di Ferrara

Logic Programming

- A subset of First Order Logic
- Closed World Assumption
- Turing complete
- Prolog

$$\begin{array}{l} \mathit{has\_starship}(\mathit{harlock}). \\ \mathit{is\_pirate}(\mathit{harlock}). \\ \mathit{spacePirate}(X) \leftarrow \mathit{has\_starship}(X). \\ \mathit{spacePirate}(X) \leftarrow \mathit{is\_pirate}(X). \end{array}$$

Description Logics

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

$$\begin{aligned}
&\mathit{spock} : \mathit{Officier}\\
&\mathit{kirk} : \mathit{Officier}\\
&\mathit{Officier} \sqsubseteq \mathit{FederationOfficier}\\
&\exists \mathit{hasOfficier}.\mathit{FederationOfficier} \sqsubseteq \mathit{FederationShip}\\
&(\mathit{ussEnterprise}, \mathit{spock}) : \mathit{hasOfficier}\\
&(\mathit{ussEnterprise}, \mathit{kirk}) : \mathit{hasOfficier}
\end{aligned}$$

Combining Logic and Probability

- Logics do 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 (1995)]: underlies many languages
- 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

PLP under the Distribution Semantics

- A PLP language under the distribution semantics with a general syntax is Logic Programs with Annotated Disjunctions (LPADs)
- Heads of clauses are disjunctions in which each atom is annotated with a probability.
- LPAD T with n clauses: $T = \{C_1, \dots, C_n\}$.
- Each clause C_i takes the form:

$$h_{i1} : \pi_{i1} ; \dots ; h_{iv_i} : \pi_{iv_i} :- b_{i1}, \dots, b_{iu_i}$$

Logic Programs with Annotated Disjunctions

$$
\begin{array}{l}
\mathit{spacePirate}(X) : 0.8 \vee \mathit{null} : 0.2 \leftarrow \mathit{has\_starship}(X). \\
\mathit{spacePirate}(X) : 0.7 \vee \mathit{null} : 0.3 \leftarrow \mathit{is\_pirate}(X). \\
\mathit{has\_starship}(\mathit{harlock}). \\
\mathit{is\_pirate}(\mathit{harlock}).
\end{array}
$$

- 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
  - Groundings are independent of each other.

Example Program (LPAD) Worlds

$spacePirate(harlock) \leftarrow has\_starship(harlock).$
$spacePirate(harlock) \leftarrow is\_pirate(harlock).$
$has\_starship(harlock).$
$is\_pirate(harlock).$
$P(w_1) = 0.8 \times 0.7$

$null \leftarrow has\_starship(harlock).$
$spacePirate(harlock) \leftarrow is\_pirate(harlock).$
$has\_starship(harlock).$
$is\_pirate(harlock).$
$P(w_2) = 0.2 \times 0.7$

$spacePirate(harlock) \leftarrow has\_starship(harlock).$
$null \leftarrow is\_pirate(harlock).$
$has\_starship(harlock).$
$is\_pirate(harlock).$
$P(w_3) = 0.8 \times 0.3$

$null \leftarrow has\_starship(harlock).$
$null \leftarrow is\_pirate(harlock).$
$has\_starship(harlock).$
$is\_pirate(harlock).$
$P(w_4) = 0.2 \times 0.3$

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

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

PLP Online

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

Description Logics

- DISPONTE: “DIstribution Semantics for Probabilistic ONTologiEs” [Riguzzi et al. (2015)]
- Probabilistic axioms:

$$p :: E$$

  - $p :: C \sqsubseteq D$ 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 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

Example

$$\begin{aligned}
&0.4 :: \mathit{spock} : \mathit{Officier}\\
&0.3 :: \mathit{kirk} : \mathit{Officier}\\
&0.6 :: \mathit{Officier} \sqsubseteq \mathit{FederationOfficier}\\
&\exists \mathit{hasOfficier}.\mathit{FederationOfficier} \sqsubseteq \mathit{FederationShip}\\
&(\mathit{ussEnterprise}, \mathit{spock}) : \mathit{hasOfficier}\\
&(\mathit{ussEnterprise}, \mathit{kirk}) : \mathit{hasOfficier}
\end{aligned}$$

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

PDL at work

- Online: `http://trill-sw.eu`
  - Inference
    - TRILL
    - TRILL^P
    - TORNADO
- Offline: BUNDLE
- All the systems are available at `http://ml.unife.it/`

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

Coping with Different Closure Assumptions

- When modeling real world domains sometimes it is required to use both Open World Assumption and Closed World Assumption.
  - LP languages adopt Closed World Assumption (CWA);
  - DLs adopt Open World Assumption (OWA).
- Domains where different information requires different closure assumptions, such as for example in legal reasoning, or health-care
- Probabilistic extensions have been proposed for several of the languages that integrate DL and LP.

Possible approaches

- FOProbLog [Bruynooghe et al. (2010)]
  - Knowledge base composed of disjunctive clauses: a first order formula annotated with a probability.
  - Probabilities act as constraints, and a model is any distribution that satisfies the constraints
  - FOProbLog does not support default negation.
- Bayesian Description Logic Programs [Predoiu and Stuckenschmidt (2007)]
  - Based on Description Logic Programs [Grosof et al. (2003)]: intersection of DL and LP.
  - Each rule is annotated with the probability that the head is true (false) when the body is true.
  - Important expressive features are not supported: default negation in LP rules, reasoning about unknown individuals and existential quantification in consequent.

Possible approaches

- Lukasiewicz's Probabilistic DL-Programs [Lukasiewicz (2007)]
  - Integrate DL-Programs [Eiter et al. (2008)] with Independent Choice Logic (ICL) [Poole (1997)]
    - DL ontology and a set of non-disjunctive LP rules
  - The atoms that occur in DL axioms cannot be the head of rules.

- Probabilistic Disjunctive DL-Programs [Lukasiewicz et al. (2011)]
  - Extension of Probabilistic DL-Programs [Lukasiewicz (2007)].
  - The probabilistic semantics defines tight lower and upper bounds for the probability of a conditional query $b|a$ where a and b are ground atoms in terms of the answer sets or well-founded model determined by the worlds.
  - The expressiveness of DL part is restricted to DL-Lite$_{\mathcal{A}}$ ontology (decomposable into a positive and a negative part).

Possible approaches

- Vaishak Belle [Belle (2017)] defines a system able to perform probabilistic inference on open-universe template models using weighted model counting.
    - Universally quantified clauses to model open-universe information.
    - Leveraging existing algorithms for the clausal representation.
- DL languages have higher expressiveness when coping with open-universe information (e.g., they allow creation of infinite chains of anonymous individuals).
- Number restrictions and existential in the head of clauses may in principle modeled with functions but this is still an open branch of study.
- Inference on DL KBs has in many cases lower complexity.

Probabilistic Hybrid Knowledge Bases

- Based on Hybrid Knowledge Bases (HKBs) [Motik and Rosati (2010)], composed of:
  - a logic program and
  - a set of DL axioms
- Follows a semantics based on the logic of Minimal Knowledge with Negation as Failure (MKNF) [Lifschitz (1991)].

Probabilistic Hybrid Knowledge Bases

- This approach is:
  - faithful $\rightarrow$ it preserves the semantics of both formalisms,
  - tight $\rightarrow$ the DL and the LP components can contribute to the consequences of the other component,
  - flexible $\rightarrow$ the same predicate can be seen under both OWA and CWA and
  - decidable
- The other approaches considered in this presentation lack one or more of these properties.

Minimal Knowledge with Negation as Failure

- MKNF HKBs have been given two-valued [Motik and Rosati (2010)], and three-valued and well-founded semantics [Knorr et al. (2011); Liu and You (2017)].
- A proof procedure, called **SLG**($\mathcal{O}$), has been presented in [Alferes et al. (2013)], which is sound and complete for the well-founded semantics
    - A top-down procedure for MKNF-based HKBs that extends **SLG** to deal with DL axioms.
    - Axioms are handled by means of an Oracle, i.e., a DL reasoner which can answer queries by assuming as true information not modeled in the DL part.

Università degli Studi di Ferrara

Probabilistic Hybrid Knowledge Bases

- A Probabilistic Hybrid Knowledge Base (PHKB) is a pair $\mathbb{K} = \langle \mathbb{O}, \mathbb{P} \rangle$ where $\mathbb{O}$ is a DISPONTE knowledge base and $\mathbb{P}$ is an LPAD without function symbols.
- In [Alberti et al. (2016)] a PHKB's semantics is given by first grounding it over all the constants in the PHKB.
- A world is the deterministic ground HKB obtained by selecting, for each clause $h_{i1} : \Pi_{i1}; \ldots; h_{in_i} : \Pi_{in_i} \leftarrow b_{i1}, \ldots, b_{im_i}$, one of the disjuncts in the head and some of the DL axioms.
- The world's probability is the product of the probabilities of the selected head disjuncts and the selected axioms.

Probabilistic Hybrid Knowledge Bases

Definition

Given a world w, the probability of a query q is defined as $P(q|w) = 1$ if $w \models q$ and 0 otherwise.
The probability of the query is its marginal probability:

$$P(q) = \sum_{w} P(w)P(q|w) = \sum_{w:w \models q} P(w) \tag{1}$$

Example 1

- KB $\mathbb{K}$ modeling the possibility to build a starship:

$$
\begin{array}{lll}
\mathbb{P} = & (C_1) & \mathit{buildAStarship} : 0.6 \leftarrow \\
 & & \mathit{hasBricks}(X), \sim\mathit{withoutImagination}(X). \\
 & & \mathit{hasBricks}(\mathit{mirko}). \\
 & (C_2) & \mathit{hasBricks}(\mathit{nadia}) : 0.3. \\
\mathbb{O} = & & \exists \mathit{designed}.\mathit{Project} \sqsubseteq \neg \mathit{withoutImagination} \\
 & & (\mathit{mirko}, \mathit{tsProject}) : \mathit{designed} \\
 & (E_1) & 0.4 \ :: \ \mathit{tsProject} : \mathit{StarshipProject} \\
 & & \mathit{StarshipProject} \sqsubseteq \mathit{Project}
\end{array}
$$

Example 1

- This KB has 16 worlds and the query *buildAStarship* is true in those:
  - that contain $hasBricks(nadia)$ and $buildAStarship \leftarrow hasBricks(nadia), {\sim}withoutImmagination(nadia)$, and
  - one world where $hasBricks(nadia)$ is absent and $tsProject : StarshipProject$ and $buildAStarship \leftarrow hasBricks(mirko), {\sim}withoutImmagination(mirko)$ are present.
- $P(buildAStarship) = 0.6 \times 0.3 \times 0.6 + 0.6 \times 0.3 \times 0.4 + 0.6 \times 0.7 \times 0.4 = 0.108 + 0.072 + 0.168 = 0.438$.

Università degli Studi di Ferrara

Example 2

- Nice, now I have my starship! But... is it a battleship or a simple starship?

- To answer this question one could check if there are laser heavy cannons on my starship, but ...
- ... traveling through the galaxy there are many dangers, so every starship should have cannons.
  - The information “if a starship has at least a laser heavy cannon then it is a battleship” satisfies the first point;
  - to model also the second point we need to assign a probability on the first point.

- However, battleships have many heavy cannons, i.e., the more heavy cannons the more likely my ship is a battleship.
- Not all cannons are heavy cannons, i.e., cannons have a certain probability to be heavy cannons.

Example 2

- KB $\mathbb{K}$ modeling the probability that my starship is a battleship:

$$\begin{array}{rcl} \mathbb{P} & = & equipped(myShip, c1) \\ & & cannon(c1). \\ & & equipped(myShip, c2). \\ & & cannon(c2). \\ \mathbb{O} & = & 0.6 \; :: \; \exists equipped.heavyCannon \sqsubseteq battleship \\ & & 0.1 \; :: \; cannon \sqsubseteq heavyCannon \end{array}$$

- $P(battleship(myShip)) = 0.6 \cdot 0.1 = 0.06$.
- This value is independent by the number of cannons equipped.

Example 2

- If we add to $\mathbb{P}$ the information “a cannon has probability 0.1 to be heavy cannon”:

$$\begin{array}{lll} \mathbb{P} & = & heavyCannon(X) : 0.1 \leftarrow cannon(X). \\ & & equipped(myShip, c1) \\ & & cannon(c1). \\ & & equipeeped(myShip, c2). \\ & & cannon(c2). \\ \mathbb{O} & = & 0.6 \;::\; \exists equipped.heavyCannon \sqsubseteq battleship \\ & & 0.1 \;::\; cannon \sqsubseteq heavyCannon \end{array}$$

- $P(battleship(myShip)) = 0.1626$.
- With 7 cannons, then $P(battleship(myShip)) = 0.3417$.

Example 3

- If we do not consider probabilities, the previous program still can be modeled with only the LP part.
- Let's consider some new information:
  - We know that if a starship can fight a Star Destroyer, it has for sure at least one heavy cannon.
  - My friend *han* ensured me that his starship has fought a Star Destroyer.

Example 3

$$\mathbb{P} = \begin{array}{l} heavyCannon(X) : 0.1 \leftarrow cannon(X). \\ equiped(hanShip, c1) \\ cannon(c1). \\ ... \end{array}$$

$$\mathbb{O} = \begin{array}{lll} 0.6 & :: & \exists equipped.heavyCannon \sqsubseteq battleship \\ 0.1 & :: & cannon \sqsubseteq heavyCannon \\ \multicolumn{3}{l}{\exists hasFought.starDestroyer \sqsubseteq \exists equipped.heavyCannon} \\ \multicolumn{3}{l}{hanShip : \exists hasFought.starDestroyer} \end{array}$$

- Blue axioms cannot be translated into LP clauses/facts
  - I don't know neither which particular Star Destroyer he has fought nor how many
  - I still don't know how many heavy cannons *han*'s ship is equipped with.

- $P(battleship(hanShip)) = 0.6$. ...Rrraugrah[1]

[1] http://starwars.wikia.com/wiki/Shyriiwook

Proof Procedure for Probabilistic Hybrid Knowledge Bases

- **sPHeRE** (for “**P**robabilistic **H**ybrid **RE**asoner”).
- Executes the **SLG**($\mathcal{O}$) proof procedure of [Alferes et al. (2013)], extended to cope with probability by means of:
  - a new operator to combine different answers,
  - an extension of the PITA transformation [Riguzzi and Swift (2011)] and
  - TRILL^{O}, an extension of the TRILL [Zese et al. (2016)] DL reasoner

Proof Procedure for Probabilistic Hybrid Knowledge Bases

- PITA transformation
  - Adds an extra argument to subgoals.
  - The extra argument stores a BDD encoding the explanations for the answers of the subgoal.
  - Exploits some predicates that operates on such BDDs, such as *and*, *or*, and *not* operators.

Proof Procedure for Probabilistic Hybrid Knowledge Bases

- TRILL^O
  - Based on TRILL, a reasoner for DISPONTE KBs that uses the tableau algorithm (a directed graph representing an ABox, where each node corresponds to an individual of the KB labeled with the concepts it belongs to and every edge a connection between individuals labeled with roles that link the two individuals).
  - Can assume atoms as true, leaving the task of controlling their truth value to the LP part.
  - Given a query, TRILL^O returns a BDD representing the explanations of the query and a set of literals assumed as true to be checked via the same proof procedure.

Conclusions

- We have seen approaches to cope with open universes.
- In general, PLP can be combined with other approaches to define powerful systems.

Thank you! Questions?

References I

Alberti, M., Cota, G., Riguzzi, F., and Zese, R. (2016). Probabilistic logical inference on the web. In Adorni, G., Cagnoni, S., Gori, M., and Maratea, M., editors, *AI*IA 2016: Advances in Artificial Intelligence, 21st Congress of the Italian Association for Artificial Intelligence, Pisa*, volume 10037 of *Lecture Notes in Computer Science*, pages 351–363. Springer International Publishing.

Alferes, J. J., Knorr, M., and Swift, T. (2013). Query-driven procedures for hybrid MKNF knowledge bases. *ACM Trans. Comput. Logic*, 14(2):16:1–16:43.

Belle, V. (2017). Open-universe weighted model counting. In Singh, S. P. and Markovitch, S., editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3701–3708. AAAI Press.

Bruynooghe, M., Mantadelis, T., Kimmig, A., Gutmann, B., Vennekens, J., Janssens, G., and De Raedt, L. (2010). Problog technology for inference in a probabilistic first order logic. In *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 719–724. IOS Press.

Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., and Tompits, H. (2008). Combining answer set programming with description logics for the semantic web. *Artif. Intell.*, 172(12-13):1495–1539.

Grosof, B. N., Horrocks, I., Volz, R., and Decker, S. (2003). Description logic programs: Combining logic programs with description logic. In *Proceedings of the12th International Conference on World Wide Web*, WWW '03, pages 48–57. ACM Press.

Knorr, M., Alferes, J. J., and Hitzler, P. (2011). Local closed world reasoning with description logics under the well-founded semantics. *Artif. Intell.*, 175(9-10):1528 – 1554.

Lifschitz, V. (1991). Nonmonotonic databases and epistemic queries. In Mylopoulos, J. and Reiter, R., editors, *12th International Joint Conference on Artificial Intelligence (IJCAI 1991)*, pages 381–386, San Francisco, CA, USA. Morgan Kaufmann.

Liu, F. and You, J.-H. (2017). Three-valued semantics for hybrid mknf knowledge bases revisited. *Artificial Intelligence*, 252(Supplement C):123 – 138.

Lukasiewicz, T. (2007). Probabilistic description logic programs. *Int. J. Approx. Reason.*, 45(2):288–307.

References II

Lukasiewicz, T., Predoiu, L., and Stuckenschmidt, H. (2011). Tightly integrated probabilistic description logic programs for representing ontology mappings. *Ann. Math. Artif. Intell.*, 63(3):385–425.

Motik, B. and Rosati, R. (2010). Reconciling description logics and rules. *J. ACM*, 57(5):30:1–30:62.

Poole, D. (1997). The Independent Choice Logic for modelling multiple agents under uncertainty. *Artif. Intell.*, 94:7–56.

Predoiu, L. and Stuckenschmidt, H. (2007). A probabilistic framework for information integration and retrieval on the semantic web.

Riguzzi, F., Bellodi, E., Lamma, E., and Zese, R. (2015). Probabilistic description logics under the distribution semantics. *Semant. Web*, 6(5):447–501.

Riguzzi, F. and Swift, T. (2011). The PITA system: Tabling and answer subsumption for reasoning under uncertainty. *Theor. Pract. Log. Prog.*, 11(4–5):433–449.

Sato, T. (1995). A statistical learning method for logic programs with distribution semantics. In Sterling, L., editor, *12th International Conference on Logic Programming (ICLP 1995)*, pages 715–729. MIT Press.

Zese, R., Bellodi, E., Riguzzi, F., Cota, G., and Lamma, E. (2016). Tableau reasoning for description logics and its extension to probabilities. *Ann. Math. Artif. Intell.*, pages 1–30.