Semantics for Hybrid Probabilistic Logic Programs with Function Symbols: Technical Summary

Damiano Azzolini[1], Fabrizio Riguzzi[2] and Evelina Lamma[3]

[1]Dipartimento di Scienze dell'Ambiente e della Prevenzione – University of Ferrara
Via Borsari 46, 44121, Ferrara, Italy
[2]Dipartimento di Matematica e Informatica – University of Ferrara, Via Machiavelli 30, 44121, Ferrara, Italy
[3]Dipartimento di Ingegneria – University of Ferrara, Via Saragat 1, 44122, Ferrara, Italy

Abstract
Hybrid probabilistic logic programs extends probabilistic logic programs by adding the possibility to manage continuous random variables. Despite the maturity of the field, a semantics that unifies discrete and continuous random variables and function symbols was still missing. In this paper, we summarize the main concepts behind a new proposed semantics for hybrid probabilistic logic programs with function symbols.

Keywords
Probabilistic Logic Programming, Hybrid Programs, Semantics

1. Contribution

This paper is a technical summary of [1]. Probabilistic Logic Programming [2] extends Logic Programming with probabilistic facts, i.e., logical atoms with an associated probability. These are usually indicated with the syntax [3]:

$$\Pi :: f$$

where f is an atom and $\Pi \in]0,1]$. Intuitively, f is true with probability Π and false with probability $1 - \Pi$. To illustrate probabilistic logic programs, let us start with a simple example:

Example 1. *Card single round.*

```
1/3 :: spades(X).
1/2 :: clubs(X).
pick(0,spades) :- spades(0).
pick(0,clubs) :- \+ spades(0), clubs(0).
pick(0,hearts) :- \+ spades(0), \+ clubs(0).
```

PLP 2022: The 9th Workshop on Probabilistic Logic Programming
damiano.azzolini@unife.it (D. Azzolini); fabrizio.riguzzi@unife.it (F. Riguzzi); evelina.lamma@unife.it (E. Lamma)
0000-0002-7133-2673 (D. Azzolini); 0000-0003-1654-9703 (F. Riguzzi); 0000-0003-2747-4292 (E. Lamma)

This program describes a game of card with only 1 round (identified with `0`*) and 3 cards. A player draws a card that can be either of spades, clubs, or hearts. These have all the same probability. Note that, to describe three possible cards we only need 2 (probabilistic) facts, since the third can be represented with the negation of the other two (see line 5). The predicate* `pick/2` *describes the three possible outcomes.*

The probability of a query asked on the program of Example 1 is easy to compute. For example, the probability of `pick(0,hearts)` is $(1 - 1/3) \cdot (1 - 1/2) = 1/3$. However, we can make the program more interesting by adding multiple rounds. To do this, we introduce a function symbol `s/1` to the program of Example 1.

Example 2. *Cards with multiple rounds. We extend Example 1 by considering multiple rounds. We introduce the function symbol* `s/1` *to indicate a round and with* `s(X)` *we indicate the round after the round* `X`*. So, starting from* `0` *(first round), we have* `s(0)` *for the second round,* `s(s(0))` *for the third round and so on. We introduce an additional rule: the game stops when the player picks a card of hearts. The program thus became:*

```
1/3 :: spades(X).
1/2 :: clubs(X).
pick(0,spades) :- spades(0).
pick(0,clubs) :- \+ spades(0), clubs(0).
pick(0,hearts) :- \+ spades(0), \+ clubs(0).
pick(s(X),spades):- \+ pick(X,hearts), spades(s(X)).
pick(s(X),clubs):- \+ pick(X,hearts), \+ spades(s(X)), clubs(s(X)).
pick(s(X),hearts):- \+ pick(X,hearts), \+ spades(s(X)),
    \+ clubs(s(X)).

at_least_once_spades :- pick(_,spades).
never_spades :- \+ at_least_once_spades.
```

A possible question could be: "what is the probability that the player picks at least one time spades?" This probability can be computed by asking the query $at\_least\_once\_spades$.

To compute the probability of the query $at\_least\_once\_spades$ from Example 2, we need to consider *pairwise incompatible covering set of explanations* [4]. If we replace `spades` with f_1 and `clubs` with f_2 and we use 0 and 1 to indicate respectively not selected and selected, we get a pairwise incompatible covering set of explanations $K = \{\kappa_0, \kappa_1, \ldots\}$ with

$$
\begin{aligned}
\kappa_0 &= \{(f_1, \{X/0\}, 1)\} \\
\kappa_1 &= \{(f_1, \{X/0\}, 0), (f_2, \{X/0\}, 1), (f_1, \{X/s(0)\}, 1)\} \\
&\ldots \\
\kappa_i &= \{(f_1, \{X/0\}, 0), (f_2, \{X/0\}, 1), \ldots, (f_1, \{X/s^{i-1}(0)\}, 0), \\
&\quad (f_2, \{X/s^{i-1}(0)\}, 1), (f_1, \{X/s^i(0)\}, 1)\} \\
&\ldots
\end{aligned}
$$

From here, we can compute the probability of the query $q = at\_least\_once\_spades$ as

$$
\begin{aligned}
P(q) =& \frac{1}{3} + \frac{1}{3} \cdot \left(\frac{2}{3} \cdot \frac{1}{2}\right) + \frac{1}{3} \cdot \left(\frac{2}{3} \cdot \frac{1}{2}\right)^2 + \dots \\
=& \frac{1}{3} + \frac{1}{3} \cdot \left(\frac{1}{3}\right) + \frac{1}{3} \cdot \left(\frac{1}{3}\right)^2 + \dots \\
=& \frac{1}{3} \cdot \frac{1}{1 - \frac{1}{3}} = \frac{1}{3} \cdot \frac{3}{2} = \frac{1}{2}
\end{aligned}
$$

since we have a sum of a geometric series.

We can even further extend the previous example by also considering continuous random variables. To represent these, we use the syntax

$$a : density$$

where a is an atom and $density$ is a special atom that denotes its probability density.

Example 3. *Cards multiple rounds and continuous random variables. We extend Example 2 by adding another rule: the player still draws a card but, in addition, he/she need to spin a wheel. If the axis of the wheel is between 0 and 180 degrees the game stops. This scenario can be encoded with:*

```
1  angle(_,X) : uniform_dens(X,0,360).
2  1/3 :: spades(X).
3  1/2 :: clubs(X).
4  pick(0,spades) :- spades(0), angle(0,V), V > 180.
5  pick(0,clubs) :- \+ spades(0), clubs(0), angle(0,V), V > 180.
6  pick(0,hearts) :- \+ spades(0), \+ clubs(0), angle(0,V), V > 180.
7  pick(s(X),spades):- \+ pick(X,hearts), spades(s(X)), angle(s(X),V),
       V > 180.
8  pick(s(X),clubs):- \+ pick(X,hearts), \+ spades(s(X)), clubs(s(X)),
       angle(s(X),V), V > 180.
9  pick(s(X),hearts):- \+ pick(X,hearts), \+ spades(s(X)),
10      \+ clubs(s(X)), angle(s(X),V), V > 180.
11
12 at_least_once_spades :- pick(_,spades).
13 never_spades :- \+ at_least_once_spades.
```

In line 1 we have a continuous probabilistic fact `angle/2` *where its argument* `X` *follows a uniform distribution between 0 and 360.*

We may be still interested in computing the probability of the query $at\_least\_once\_spades$ from Example 3. Differently from Example 2, we now need to consider a *mutually disjoint covering set of worlds* ω. First, we partition the random variables in two sets: a countable set X of continuous random variables (identified by 0, $s(0)$, …, where each element has a range

$[0, 360]$) and a countable set Y of discrete random variables (where each element can be true or false). The set $\omega = \omega_0 \cup \omega_1 \ldots$ is such that:

$$\begin{aligned}
\omega_0 = \{(w_\mathrm{X}, w_\mathrm{Y}) \mid\ & w_\mathrm{X} = (x_0, x_1, \ldots), w_\mathrm{Y} = (y_0^c, y_0^s, y_1^c, y_1^s, \ldots), \\
& x_0 \in]180, 360], y_0^s = 1\} \\
\omega_1 = \{(w_\mathrm{X}, w_\mathrm{Y}) \mid\ & w_\mathrm{X} = (x_0, x_1, \ldots), w_\mathrm{Y} = (y_0^c, y_0^s, y_1^c, y_1^s, \ldots), \\
& x_0 \in]180, 360], y_0^s = 0, y_0^c = 1, x_1 \in]180, 360], y_1^s = 1\} \\
\ldots
\end{aligned}$$

In other words: for ω_0, spades was selected at round 0 ($y_0^s = 1$) and the wheel (x_0) in the same round was in the range $]180, 360]$; for ω_1, spades was not selected at round 0 ($y_0^s = 0$), clubs was selected at round 0 ($y_0^c = 1$), the wheel (x_0) was in the range $]180, 360]$ at round 0, spades was selected at round $s(0)$ ($y_1^s = 1$) and the wheel (x_1) was in the range $]180, 360]$ at round $s(0)$, and so on.

The probability for each ω_i can be computed by multiplying the discrete and continuous components. For ω_0 (the process is similar for all the $\omega_i \in \omega$) we have:

$$\begin{aligned}
\mu(\omega_0) &= \int_{180}^{360} \mu_\mathrm{Y}(\{(y_0^c, y_0^s, y_1^c, y_1^s, \ldots) \mid y_0^c = 1\})\, d\mu_\mathrm{X} \\
&= \int_{180}^{360} \frac{1}{3} \cdot \frac{1}{360} dx_0 = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}.
\end{aligned}$$

where $\frac{1}{3}$ is the contribution of the discrete random variable (spades) and $\frac{1}{360}$ is the contribution of the continuous one (angle). By considering all the values obtained for all the ω_i, we get $\frac{1}{3} \cdot \frac{1}{2} \cdot \sum_{i=0}^{\infty} (\frac{2}{3} \cdot \frac{1}{2} \cdot \frac{1}{2})^i = \frac{1}{6} \cdot \sum_{i=0}^{\infty} (\frac{1}{6})^i = \frac{1}{6} \cdot \frac{6}{5} = \frac{1}{5}$ as probability for the query `at_least_once_spades`.

In [1], we prove that this semantics is well defined, i.e., it assigns a probability value to queries for a large class of programs. However, as discussed in [5], these programs must met some requirements, mainly needed to ensure the existence of the sets of discrete and continuous random variables. These requirements are: 1) the set of random variables must be countable; 2) clauses with the same head but different bodies must be mutually exclusive; 3) the value of a continuous random variable must be used only as a parameter for another distribution, and not as a variable for another term; 4) clauses must be range restricted (every variable in the head also appears in a positive literal in the body): this ensures that answers to queries are ground instantiations of it. For a more in-depth discussion see [5].

References

[1] D. Azzolini, F. Riguzzi, E. Lamma, A semantics for hybrid probabilistic logic programs with function symbols, Artificial Intelligence 294 (2021) 103452. doi:10.1016/j.artint.2021.103452.

[2] F. Riguzzi, Foundations of Probabilistic Logic Programming: Languages, semantics, inference and learning, River Publishers, Gistrup, Denmark, 2018.

[3] L. De Raedt, A. Kimmig, H. Toivonen, Problog: A probabilistic prolog and its application in link discovery, in: M. M. Veloso (Ed.), IJCAI, 2007, pp. 2462–2467.

[4] F. Riguzzi, The distribution semantics for normal programs with function symbols, International Journal of Approzimate Reasoning 77 (2016) 1–19. doi:10.1016/j.ijar.2016.05.005.

[5] D. Azzolini, F. Riguzzi, Syntactic requirements for well-defined hybrid probabilistic logic programs, in: A. Formisano, Y. A. Liu, B. Bogaerts, A. Brik, V. Dahl, C. Dodaro, P. Fodor, G. L. Pozzato, J. Vennekens, N.-F. Zhou (Eds.), Proceedings 37th International Conference on Logic Programming (Technical Communications), Open Publishing Association, Waterloo, Australia, 2021, pp. 14–26. doi:10.4204/EPTCS.345.