Extending Probabilistic Datalog

Norbert Fuhr
University of Dortmund, Germany

1 Motivation

In [Fuhr 95], we have proposed probabilistic Datalog (pD) as an inference engine for IR. However, a closer look at pD shows that it handles conditional probabilities only in a rather limited way, since it neither allows for the specification of conditional probabilities in rules nor is it able to compute conditional probabilities. Given that probabilistic IR should be interpreted as computing the probability $P(d \rightarrow q) = P(q|d)$, handling of conditional probabilities is essential. In the following, we will describe how this can be achieved.

Another important issue is retrieval for composite documents. In [Rölleke & Fuhr 96b], we have proposed a 4-valued logic for this purpose, and in the companion paper [Rölleke & Fuhr 96a] we extend this approach to uncertain inference. In this paper, we show how this type of logic can be implemented using a 4-valued logic version of pD (4pD). The basic idea is that augmentation can be formulated by means of probabilistic rules in 4pD.

2 2-valued logic

2.1 Probabilistic rules

Probabilistic rules in pD specify conditional probabilities. For example, we may specify that 50% of all humans are women by means of a rule `0.5 woman(X) :- human(X)`, standing for the conditional probability $P(\texttt{woman}|\texttt{human}) = 0.5$. For the computation of the correct probabilities in the pD system, this rule generates an additional event (with probability 0.5) for each instance of X. For example, given the fact `0.7 human(peter)`, for `woman(peter)` we get the event expression `woman(peter) & human(peter)`.

Problems arise if more than one rule is formulated for the same head predicate: given two rules for a predicate l specifying $P(l|r)$ and $P(l|s)$, we are not able to estimate $P(l|r \wedge s)$ for the case when the bodies of both rules are fulfilled. Thus, probabilistic rules have to be formulated in a way such that at most one rule is true at a time. In the following example, we show how probabilistic links between two documents are specified subject to authorship (`author`) and explicit references (`refer`) between documents:

```
sameauthor(D1,D2) :- author(D1,X) & author(D2,X).
0.7 link(D1,D2) :- refer(D1,D2) & sameauthor(D1,D2).
0.5 link(D1,D2) :- refer(D1,D2) & not(sameauthor(D1,D2)).
0.2 link(D1,D2) :- sameauthor(D1,D2) & not(refer(D1,D2)).
```

In fact, this type of rule formulation reflects the specification of link matrices as in the inference network approach.

2.2 Computing conditional probabilities

As mentioned above, probabilistic IR also requires the computation of conditional probabilities according to the formula $P(a|b) = \frac{P(a \wedge b)}{P(b)}$. In pD, both a and b may be rule bodies (in the following, we also allow for disjunction denoted by | in rule bodies). The following pD program computes the conditional probability that a dice shows an even number, given that it shows more than 3:

```
0.17 dice(d,1).  0.17 dice(d,2).  ...  0.17 dice(d,6)
even(D) :- dice(D,2) | dice(D,4) | dice(D,6).
high(D) :- dice(D,4) | dice(D,5) | dice(D,6).
ev_g_hi(D) :- even(D) / high(D).
```

In terms of event expressions, a new event `ev_g_hi(d)` is created, with the probability computed as

$$\frac{P((\texttt{dice(d,2)}|\texttt{dice(d,4)}|\texttt{dice(d,6)})\&(\texttt{dice(d,4)}|\texttt{dice(d,5)}|\texttt{dice(d,6)}))}{P(\texttt{dice(d,4)}|\texttt{dice(d,5)}|\texttt{dice(d,6)})}$$

As a simple example for an IR application, consider the probabilistic view on the vector space model as described in [Wong & Yao 95], where terms are treated as disjoint concepts. In this case, we have

$$\begin{aligned} P(d \rightarrow q) &= \sum_t P(t|d)P(q|t) \\ P(t|d) &= \frac{P(t,d)}{P(d)} \\ P(d) &= \sum_t P(t,d) \end{aligned}$$

Now assume that we have the following predicates representing the corresponding basic probabilities:

$$\begin{aligned} \texttt{dt(D,T)} &\hat{=} P(d,t) \\ \texttt{qgt(Q,T)} &\hat{=} P(q|t) \\ \texttt{tgd(T,D)} &\hat{=} P(t|d) \\ \texttt{d(D)} &\hat{=} P(d) \\ \texttt{qgd(Q,D)} &\hat{=} P(q|d) \end{aligned}$$

Then the conditional probability $P(t|d)$ can be computed as a result of the rules

```
tgd(T,D) :- dt(D,T) / d(D).
d(D)     :- dt(D,T).
```

and the probability of document `D` implying query `Q` is estimated by the rule

```
qgd(Q,D) :- tgd(T,D) & qgt(Q,T).
```

However, this aproach fails in case we are not given the probabilities of the underlying (disjoint) events for computing the conditional probabilities, as for example with $P(q|t)$. For these cases, we need an alternative method for computing conditional probabilities, namely as the quotient of the probabilities of two event expressions. For example, let $\texttt{qt(Q,T)} \hat{=} P(q,t)$ and $\texttt{t(T)} \hat{=} P(t)$, then we can compute $P(q|t)$ by means of a rule

```
qgt(Q,T) :- qt(Q,T) // t(T).
```

These two types of conditional probabilities have a wide variety of applications in IR; for example, imaging can be expressed this way.

3 4-valued logic

3.1 Basic assumptions

For an atom of a given Herbrand base, assume that a world may contain the positive atom, the negated atom, both the positive and the negated atom or none of both. This corresponds to the four truth values true (T), false (F), inconsistent(I) and unknown (U). For example, let the Herbrand base consist of the four atoms `p(a)`, `p(b)`, `p(c)` and `p(d)`, and let W_1 = {`p(b)`, ¬`p(c)`, `p(d)`,¬ `p(d)`}. Then we have the following truth values: τ(`p(a)`) = U, τ(`p(b)`) = T, τ(`p(c)`) = F and τ(`p(d)`) = I.

For 4-valued probabilistic Datalog, we specify the probabilities of an event as a pair of values, where the first stands for the probability of the positive atom (i.e. T or I) and the second for the negated atom (i.e. F or I). If only one value is given, it stands for the probability of true, and the probability of false is the complement to 1. Here we show an example model:

0.3 W_1	`p(a)`	`p(b)`	¬`p(c)`	`p(d)`,¬ `p(d)`
0.5 W_2		`p(b)`	`p(c)`, ¬`p(c)`	`p(d)`,¬ `p(d)`
0.2 W_3		¬`p(b)`	`p(c)`	`p(d)`,¬ `p(d)`
	0.3/0.0 `p(a)`	0.8/0.2 `p(b)`	0.7/0.8 `p(c)`	1.0/1.0 `p(d)`

For Boolean connectors, we define as shown in the following table. In contrast to [Belnap 77], the negation of unknown remains unknown, and the negation of inconsistent remains inconsistent.

∧	T	F	U	I
T	T	F	U	I
F	F	F	F	F
U	U	F	U	F
I	I	F	F	I

∨	T	F	U	I
T	T	T	T	T
F	T	F	U	I
U	T	U	U	T
I	T	I	T	I

¬	
T	F
F	T
U	U
I	I

Assuming probabilistic independence, the combination of two probabilistic events with probabilities t_1/f_1 and t_2/f_2 yields the following results:

	t	f
∧	$t_1 \cdot t_2$	$f_1 + f_2 - f_1 \cdot f_2$
∨	$t_1 + t_2 - t_1 \cdot t_2$	$f_1 \cdot f_2$
¬	f_1	t_1

The following table shows some examples for the conjunction and disjunction of two events a and b:

a	0.8/0.2	0.8/0.4	0.2/0.3	1.0/1.0	0.0/0.0
b	0.5/0.5	0.7/0.3	0.7/0.3	0.7/0.3	0.7/0.3
$a \wedge b$	0.4/0.6	0.56/0.58	0.14/0.51	0.7/1.0	0.0/0.3
$a \vee b$	0.9/0.1	0.96/0.12	0.76/0.09	1.0/0.3	0.7/0.0

3.2 Entailment

In order to define a reasonable semantics for entailment, we must consider that we want to support IR applications. Thus, we want to restrict the inferential capabilities such that the system is not burdened with inferences which are not essential for IR, e.g. inferring that certain facts are unknown.

First, let us define implication as usually, i.e. $a \rightarrow b = \neg a \vee b$. This gives us the following truth table:

→	T	F	U	I
T	T	F	U	I
F	T	T	T	T
U	T	U	U	T
I	T	I	T	I

Standard Datalog is based on modus ponens, so we now have to decide how modus ponens should be used in four-valued logic. In principle, there are four possibilities, depending on the truth values of antecedent and the precedent, which both may be either true only or true or inconsistent:

1. T $\rightarrow$ T
2. T $\rightarrow$ T,I
3. T,I $\rightarrow$ T,I
4. T,I $\rightarrow$ T

Since we do not want to draw any more inferences from an inconsistent antecedent, cases 3 and 4 drop out. Case 1 is the same as in 2-valued logic, but raises two problems in 4-valued logic: First, we are not able to infer any inconsistent facts, and second, programs that would lead to inconsistent facts are incorrect. Thus, we are left with the second choice, where the antecedent must be true, from which we can derive that the precedent is either true or inconsistent. Looking at the truth table of implication, we see that for the antecedent being true, the implication is true if the precedent is true, and it is inconsistent if the precedent is inconsistent, too. In other words, if W is a world in which the implication $a \rightarrow b$ holds (i.e. has truth value T or I), then W is an interpretation iff whenever $a \in W$ and $\neg a \notin W$, then $b \in W$.

Let us denote the truth value(s) as a subscript of the operator $\models$. Then we can formulate modus ponens in our 4-valued logic as follows:

$$\text{If } M \models_T a \text{ and } M \models_{T,I} a \rightarrow b, \text{ then } M \models_{T,I} b.$$

As a simple example, assume a rule `doc(X) :- book(X)`, from which we can conclude that `doc(X)` is true or inconsistent in case `book(X)` is true, but nothing else. If we also want to infer negative information, this has to be stated explicitly. For example, if we assume that X is a student iff she is a person and enrolled, we can formulate this by two rules:

```
student(X) :- person(X) & enrolled(X).
not(student(X)) :- not(person(X)) | not(enrolled(X)).
```

As a more complex example, assume that we want to formulate retrieval rules for structured documents, as outlined in [Rölleke & Fuhr 96b]. For augmentation, we postulate that a document node is about a term if it has access to a node that is indexed positively with that term, and it is not about a term if it can access a node that is negatively indexed with that term:

```
about(D,T) :- acc(D,D1) & docterm(D1,T).
not(about(D,T)) :- acc(D,D1) & not(docterm(D1,T)).
```

From these rules, we can also derive inconsistent knowledge in case we have access to two different nodes, one positively and one negatively indexed with the same term. On the other hand, if an accessible node has inconsistent knowledge with respect to `docterm`, then no further inferences can be drawn from this fact.

4 Outlook

Here we have sketched several concepts for extending probabilistic Datalog. In 2-valued logic, the precise semantics of conditional probabilities is yet to be defined. For the 4-valued logic case, we are investigating different evaluation algorithms.

References

Belnap, N. (1977). A Useful Four-Valued Logic. In: Dunn; Epstein (eds.): *Modern Uses of Multiple-Valued Logic*. Reidel, Dordrecht.

Fuhr, N. (1995). Probabilistic Datalog - a Logic for Powerful Retrieval Methods. In: Fox, E.; Ingwersen, P.; Fidel, R. (eds.): *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 282–290. ACM, New York.

Rölleke, T.; Fuhr, N. (1996a). Composite Documents and Uncertain Inference. In: *Proceedings 1996 Workshop on Information Retrieval, Uncertainty and Logic.*

Rölleke, T.; Fuhr, N. (1996b). Retrieval of Complex Objects Using a Four-Valued Logic. In: *Proceedings SIGIR'96.* ACM, New York.

Wong, S.; Yao, Y. (1995). On Modeling Information Retrieval with Probabilistic Inference. *ACM Transactions on Information Systems 13(1)*, pages 38–68.