

Nonmonotonicity of Probabilistic Reasoning

Michael Gelfond and J. Nelson Rushton

Computer Science Department

Texas Tech University

Lubbock, TX 79409 USA

michael.gelfond@ttu.edu, nelson.rushton@ttu.edu

In the last thirty years a substantial amount of work in Knowledge Representation and Reasoning concentrated on the development and understanding of logical languages with non-monotonic entailment relations [5]. Recall that an entailment relation is called *non-monotonic* if $T \models F$ does not imply that $T \cup A \models F$, i.e. the addition of new information can force a reasoner to withdraw his previous conclusion. A logical language with a non-monotonic entailment relation is often referred to as a *non-monotonic logic*. The work on non-monotonic logics helped us to better understand and model various forms of non-mathematical reasoning, including reasoning with defaults, common sense reasoning about actions and change, etc. In addition there is compelling evidence that the use of non-monotonic logics for knowledge representation increases the degree of elaboration tolerance of our knowledge bases and provides a useful tool for declarative programming. Non-monotonic logics normally allow reasoning about possible beliefs of agents associated with the corresponding theories. The agent may believe that p is true, or that p is false, or can remain undecided about the truth value of p . In some situations, however, one may want to represent and reason with a more finely graded degrees of reasoner's beliefs. This requires probabilistic reasoning. A substantial amount of work has been devoted to modeling reasoning combining both logic and probability. Most of this work, however, concentrated on the syntax and semantics of the corresponding languages, and on the development of efficient reasoning algorithms. To the best of our knowledge much less was done on understanding the non-monotonic nature of common sense reasoning which involves probabilities. At first glance that may seem natural. In classical probability theory, changes in probability caused by obtaining new knowledge are normally handled by *conditional probability*, $P(A | B)$. This guarantees some degree of non-monotonicity – after all addition of a new information, B , may force the reasoner to change its degree of belief in A . However the conditional probability approach limits the possible updates to knowledge expressible by propositional formulas of the original language. But it does not tell us what should happen if the agent learned a new default, or some rules expanding his vocabulary by new terms, or an occurrence of a deliberate action in the sense of Pearl, etc. Moreover, modeling updates by conditional probability does

not allow the reasoner to *react to new knowledge by changing probabilistic models¹ of the domain and creating new possible worlds*. We will argue that this is an important ability which corresponds to true non-monotonicity of logics allowing probabilistic reasoning. We illustrate our point by a simple example written in the knowledge representation language P-log [2], which can be viewed as “probabilistic” expansion of Answer Set Prolog (see, for instance, [1]). A P-log program Π defines a probabilistic model representing an agent’s beliefs about some domain. Answer sets of Π are identified with *possible worlds* of the model. The probability function is given by the program’s probabilistic part, and by the *Principle of Indifference* — a commonsense rule which says that *any two possible values of a random variable are assumed to be equally likely if we have no reason to prefer one of them to the other*. The P-log program below describes the effects of a robot’s movement from location r_0 to location r_1 . P-log requires declarations of sorts and function symbols. The sorts are declared by statements:

$time = \{0, 1\}$ $locations = \{r_0, r_1\}$

We will need three functions:

$in : time \rightarrow locations$

which returns the location of the robot at a given time, and

$faulty : boolean$

which returns *true* if the robot is faulty, and

$move : locations$

which denotes the location the robot attempts to move to at moment 0.

To illustrate our point we also need the initial situation

$in(0) = r_0$ $move = r_1$

a causal law

$in(1) = r_1 \leftarrow move = r_1, not\ faulty$

The program, Π_0 , described above can be viewed as a program of Answer Set Prolog (written in the slightly expanded syntax). Clearly the program entails $in(1) = r_1$.

Suppose now we learn that: *When a faulty robot attempts to move, there is only a 0.8 probability that the move is successful; otherwise it stays where it is*. To represent this information in P-log we first declare the location of the faulty robot after its attempt to move to location r_1 to be random.

$random(in(1)) \leftarrow faulty, move = r_1$.

Roughly speaking this statement has the same meaning as an Answer Set Prolog rule

$in(1) = r_0$ or $in(1) = r_1 \leftarrow faulty, move = r_1$.

The numerical probabilistic information from the story can be written in P-log as follows:

¹Probabilistic model is a set of possible worlds and probability function defined on sets of such worlds.

$$pr(in(1) = r_1 \mid \textit{faulty}, \textit{move} = r_1) = .8$$

The new program, say, Π_1 still entails $in(1) = 1$. Π_1 has exactly one answer set and hence the probabilistic model defined by Π_1 has exactly one possible world, $\{\textit{move} = r_1, in(0) = r_0, in(1) = r_1\}$. Hence, Π_1 entails that the probability of the robot remaining into location r_0 after the move is 0. Suppose now the reasoner learns that *the robot is faulty*. This fact is simply added to his knowledge base. It is not difficult to see that the resulting program, Π_2 , has two possible worlds $\{\textit{faulty}, \textit{move} = r_1, in(0) = r_0, in(1) = r_0\}$ and $\{\textit{faulty}, \textit{move} = r_1, in(0) = r_0, in(1) = r_1\}$. The reasoner changed his probabilistic model. The old possible world is replaced by the two new ones. The previous probabilistic conclusion is withdrawn. The new program entails that the robot can stay in r_0 after the move with probability .2.

This update cannot be represented using standard conditional probability, because it is a theorem of classical probability that if $P(A) = 0$ then $P(A \mid B) = 0$.

The ability to introduce new possible worlds as a result of conditioning reflects the common sense semantics of utterances such as “the robot might be faulty.” In practice, such a sentence does not eliminate existing possible beliefs, and so there is no classical (i.e., monotonic) semantics in which the statement would be informative. If it is informative, as common sense suggests, then its content seems to introduce new possibilities into the listener’s thought process. It can also improve elaboration tolerance of our knowledge bases as well as the efficiency of reasoning.

There are, of course, a large number of unanswered questions relating to P-log. This includes the development of efficient inference engines, better understanding the methodology of P-log’s use, and P-log’s mathematical properties. Comparison with other reasoning tools, including graphical models, will also be very useful. Further reading on the subject may include [4, 2, 3].

References

- [1] Chitta Baral. *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press, Jan 2003.
- [2] Chitta Baral, Michael Gelfond, and Nelson Rushton. Probabilistic reasoning with answer sets. *Journal of Theory and Practice of Logic Programming (TPLP)*, 9(1):57–144, 2009.
- [3] Chitta Baral and Matt Hunsaker. Using the probabilistic logic programming language p-log for causal and counterfactual reasoning and non-naive conditioning. In *Proceedings of IJCAI-2007*, pages 243–249, 2007.
- [4] M. Gelfond and N. Rushton. Causal and probabilistic reasoning in p-log. In R. Dechter, H. Geffner, and J. Halpern., editors, *Heuristics, Probabilities and Causality. A tribute to Judea Pearl*, pages 337–359. College Publications, 2010.
- [5] Victor W. Marek and Miroslaw Truszczyński. *Nonmonotonic logics; context dependent reasoning*. Springer Verlag, Berlin, 1993.