Learning and Planning with Probabilistic Relational Rules

Hanna Pasula, Luke Zettlemoyer, and Leslie Pack Kaelbling

The Problem: Our research involves learning models of world action dynamics, which can then be used to construct plans to reach a wide range of goals. The work is applied to simulated worlds, such as the blocks-world environment shown in Figure 1.

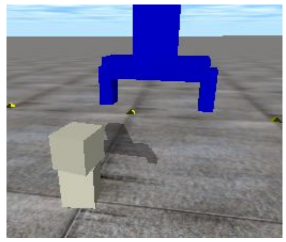

Figure 1: A three-dimensional blocks-world simulation built with the OpenDynamics toolkit [7]. The world consists of a table, blocks of roughly uniform size and mass, and a robotic hand that is moved by simulated motors.

Motivation: Robust robotic control in complex worlds is a challenging problem. Hand-engineering a solution is difficult and time-consuming. Developing techniques that will allow robots to gather knowledge about the world and use it to design their own control strategies seems like a reasonable alternative.

Previous Work: We represent world action dynamics using probabilistic planning rules. Figure refrel-rules-fig shows two rules that model actions that can be performed by the robotic arm in the blocks world of Figure 1. Such rules enable us to take advantage of the inherent structure found in many uncertain, complex environments by making the following assumptions about the world:

- **Frame Assumption:** When an agent takes an action in a world, anything not explicitly changed by that action stays the same.
- **Object Abstraction Assumption:** The world is made up of objects, and the effects of actions on these objects generally depend on their attributes rather than their identities.
- **Action Outcomes Assumption:** Each action can only affect the world in a small number of distinct ways. Each possible effect causes a set of changes to the world that happen together as a single *outcome*.

The first two assumptions have been captured in almost all planning representations, such as STRIPS operators [3]. The third assumption has been made by several probabilistic planning representations, such as the probabilistic rules of Blum and Langford [2].

The problem of learning deterministic action models is well-studied [9, 1, 4], that of learning probabilistic action models less so: we know of only one method, that of Oates [6], however, their representation is propositional and allows each rule to contain only a single outcome. We have developed an algorithm capable of learning rules such as those in Figure 2 [8]. There exist several algorithms for acting with planning representations that scale to large problems [2, 5].

$$\mathit{pickup}(X,Y) : \mathit{on}(X,Y), \mathit{clear}(X), \mathit{inhand\text{-}nil} \rightarrow \left\{ \begin{array}{ll} .7: & \mathit{clear}(Y), \neg\mathit{clear}(X), \neg\mathit{inhand\text{-}nil}, \mathit{inhand}(X), \neg\mathit{on}(X,Y) \\ .2: & \mathit{clear}(Y), \neg\mathit{on}(X,Y), \mathit{on}(X,\textsc{table}) \\ .1: & \text{no change} \end{array} \right.$$

$$\mathit{puton}(X,\textsc{table}) : \mathit{inhand}(X) \rightarrow \left\{ \begin{array}{ll} .66: & \mathit{inhand\text{-}nil}, \neg\mathit{inhand}(X), \mathit{on}(X,\textsc{table}), \mathit{clear}(X) \\ .34: & \text{no change} \end{array} \right.$$

Figure 2: Two rules for a simple blocks world. The top line of each rule gives the action and the context. The bracketed lines describe the outcomes and their associated distribution.

Approach: Our research extends probabilistic relational rules in ways that enable planning and learning in environments that do not fully meet the planning assumptions from previous work. We are exploring the following representational extensions:

- **Noise:** Traditional rules rely on the existence of a relatively small set of simple, deterministic outcomes. In many environments, such as our simulated blocks world, this assumption is broken by the existence of additional complicated, low-probability transitions. We are extending traditional rules to include a noise process that models these transitions.

- **Deixis:** Deictic references refer to objects in the world via relative statements, such as the-block-that-I-am-holding, rather than constants, like block-12. A deictic reference names the unique object in a world that fills the appropriate role. As originally noticed by Benson [1], all variables in planning operators are, in essence, deictic references, since they must also map to a unique object whenever the rule is used. For example, in the action $pickup(X)$, X could easily be called the-block-that-I-am-trying-to-pickup, a more traditional deictic name. We are exploring *deictic rules*: rules that cannot include constants.

For both of these representational extensions, we will develop algorithms for planning and learning. These algorithms will be based our previous learning algorithm [8] and the planning algorithm of Kearns, et. al. [5].

Impact: By designing robots that learn their own control strategies, we will enable them to explore and act in complex and novel domains.

Future Work: We plan to extend our approach in several ways. Since it seems unreasonable to assume that an appropriate language, one capturing all the important properties of the world, will be available in every environment, we are also investigating the possibility of constructing useful predicate-concepts out of a given set of perceptual primitives. Also, in more realistic domains, robots do not have perfect global perception. Expanding our approach to allow for partial observability will greatly improve its applicability to real robotic systems. Finally, we believe that a robot should be able to learn from the world as it interacts with it. It is not always possible to obtain a set of training examples that contains a reasonable sampling of those worlds that are likely to be relevant to the robot. Developing incremental, online algorithms that learn as the robot explores will be essential for scaling this approach to large domains.

References:

[1] S. Benson. *Learning Action Models for Reactive Autonomous Agents*. PhD thesis, Stanford University, 1996.

[2] A. Blum and J. Langford. Probabilistic planning in the graphplan framework. In *Proceedings of the Fifth European Conference on Planning*, 1999.

[3] Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(2), 1971.

[4] Y. Gil. Learning by experimentation: Incremental refinement of incomplete planning domains. In *Proceedings of the Eleventh International Conference on Machine Learning*, 1994.

[5] M. Kearns, Y. Mansour, and A. Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine Learning*, 49(2), 2002.

[6] Tim Oates and Paul R Cohen. Searching for planning operators with context-dependent and probabilistic effects. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 1996.

[7] ODE. Open dynamics engine toolkit, 2004. http://opende.sourceforge.net.

[8] H. Pasula, L. Zettlemoyer, and L. Kaelbling. Learning probabilistic relational planning rules. In *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling*, 2004.

[9] Wei-Min Shen and Herbert A. Simon. Rule creation and rule learning through environmental exploration. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1989.