

# Towards employing PSRs in a continuous domain

Nicholas K. Jong and Peter Stone

Department of Computer Sciences  
University of Texas at Austin  
Austin, Texas 78712  
{nkj,pstone}@cs.utexas.edu

Technical Report UT-AI-TR-04-309

February 24, 2004

## Abstract

Predictive State Representations (PSRs) recently emerged as an alternative framework for reasoning about stochastic environments. However, unlike Markov decision processes, they have not yet been extended to large domains or domains with continuous state variables. This report briefly describes an attempt to scale PSRs to such domains. Our goal was to construct a PSR allowing an agent to track its location on the simulated soccer field used in Robocup. This line of work ended in a negative result.

## 1 Introduction

Predictive State Representations [1, 2] have shown some promise as an alternative way for an agent to model stochastic environments. Current research has been addressing the question of PSR discovery, learning, and planning in these limited domains. Generalizing PSRs to larger, more continuous domains would address another shortcoming of the representation: to date, it has only been applied to small, finite environments. The purpose of this work was simply to try applying the PSR approach to a continuous domain, both to demonstrate the power of the representation and to gain a better understanding of the issues involved. We hoped that working in a spatial domain would give us a better intuition for the task.

## 2 The Soccer Simulator

The domain we chose was localization and navigation on a simulated soccer field. We employed the soccer simulator used by the RoboCup simulation league, defines a relatively sophisticated sensor and action model for each simulated soccer player. Our eventual goal was to have a PSR-based agent navigate around the pitch while localizing, given the distance and heading to the flags that surround the field.

Given the complexity of the domain, we kept simplifying this localization task until we reached a manageable level. For the work described below we started with a very simple domain. The agent sat at the very center of the field and could only rotate left and right at a fixed speed. The agent could see only the six primary flags, with one at each corner and one each at the middle of each sideline. Finally, we disabled the noise that the simulator adds by default to all of the simulated player's movement.

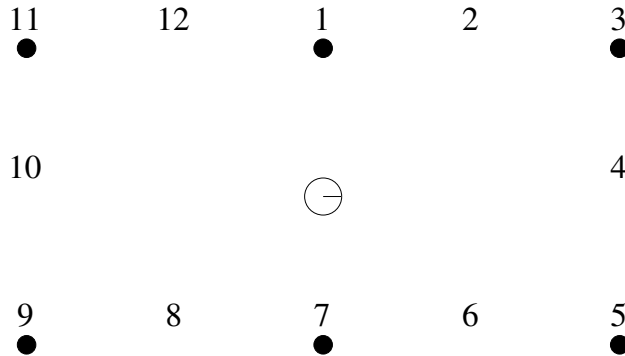


Figure 1: The environment we examined (not to scale). The six filled circles represent the six flags. The circle in the center is the agent. The odd numbers represent percepts in which the agent sees exactly one flag; the even numbers represent percepts in which the agent sees two adjacent flags.

### 3 Creating a PSR

Given the current formulation of PSRs, the task of creating a representation of this domain entailed at least three subtasks. First, we must provide an appropriate action space and an appropriate observation space for the domain. Next, we must determine the core tests, whose probabilities completely describe the state. Finally, we must determine the projection functions for the extension tests, which allow an agent to update the PSR and to make predictions.

#### 3.1 Discretization

Without any clear way of generalizing the PSR notion of a test more directly to the continuous case, we began by discretizing the soccer field domain. One immediate concern is the combinatorial explosion of parameters required as the number of actions and observations increase. In particular, a domain with  $A$  actions,  $O$  observations, and  $Q$  core tests requires a complete PSR model to specify  $AOQ$  projection functions.

As a first step, we decided to make the action and observation discretizations as coarse as possible. We restricted the agent to two actions, allowing it only to turn left or right precisely 40 degrees, since disallowing translational movement should make the overall task far easier. Knowing that the agent would always remain at the center of the field also allowed us to work with a relatively small observation space. We observed that from the center of the field, the agent always perceived either exactly one flag or exactly two adjacent flags. Hence, we could encode the precise subset of the six flags the agent sees at a given time step using only twelve distinct observations, as shown in Figure 1.

#### 3.2 Core test discovery

##### 3.2.1 Initialization

Given the action and observation spaces described above, the environment as perceived by the agent would be trivial, if the underlying soccer player always began at the same starting orientation. Since exactly nine “Turn Right” actions comprised exactly one complete rotation, the agent would simply perceive a constant sequence of nine distinct observations. This domain would simply require nine core tests, each consisting of a single action-observation pair.

To make the task not completely trivial, the agent began each trial at a random starting orientation drawn from the set  $\{-8, -4, 0, 4, 8\}$  (degrees). If the agent always choose to turn right, each starting position yielded a distinct sequence of nine observations, as shown in Figure 2.

The agent remains within one “track” for the duration of a trial, so its task is primarily to figure out in which track it resides. Once the agent has localized, updating the state within one track is simple.

$\theta_0$	Observation sequence								
-8	4	5	6	8	9	10	12	1	2
-4	4	5	6	8	9	11	12	1	3
0	4	5	7	8	9	11	12	1	3
4	4	5	7	8	9	11	12	2	3
8	4	6	7	8	10	11	12	2	3

Figure 2: The observations that result from continually turning right, given five initial starting angles.

$\epsilon$	<i>R9</i>	<i>L9</i>	<i>L5R6</i>	<i>R3RAR5</i>
<i>R1</i>	<i>R10</i>	<i>L10</i>	<i>L8R9</i>	<i>R4R5R6</i>
<i>R2</i>	<i>R11</i>	<i>L11</i>	<i>R9R10</i>	<i>R8R9R10</i>
<i>R3</i>	<i>L1</i>	<i>R12R1</i>	<i>L9R10</i>	<i>L4L3L1</i>
<i>R4</i>	<i>L2</i>	<i>L12R1</i>	<i>L3L1</i>	<i>L9L8L6</i>
<i>R5</i>	<i>L3</i>	<i>L1R1</i>	<i>L4L2</i>	<i>R9R11R12R1</i>
<i>R6</i>	<i>L5</i>	<i>R4R5</i>	<i>L8L6</i>	<i>L5L4L3L1</i>
<i>R7</i>	<i>L6</i>	<i>L4R5</i>	<i>L11L9</i>	<i>L11L9L8L6</i>
<i>R8</i>	<i>L7</i>	<i>R5R6</i>	<i>R11R12R1</i>	<i>R3RAR5R6</i>

Figure 3: The core tests from the domain we investigated.

### 3.2.2 Discovery methods

We had come this far on the presumption that the process of developing the observation and action spaces would develop some intuition for choosing the core tests through some domain specific method or even manually. However, this did not prove to be the case. We did not discover a method of generating core tests that did not resort in the end to translating back and forth from the more familiar geometric (underlying state-based) representation. We did examine the set of core tests that the established discovery algorithm found, relying on the ability to reset the environment to every possible state. These core tests are shown in Figure 3.

In an effort better to understand the domain, we tried computing the projection function mapping these core tests to some arbitrary other core tests. For a domain this small, we can find the projection function by computing the probability of each test as a sum of probabilities of underlying states. For example, the probability that the test *R1R3* would be true is equal to the probability of having started at  $\theta_0 = -4$  or  $\theta_0 = 0$  and having last seen observation 12. (Note in Figure 2 that the subsequence 1,3 only occurs for  $\theta_0 \in \{-4, 0\}$ , and in each case the preceding observation is 12.) Since the 45 tests in Figure 3 are core tests,  $\Pr(R1R3)$  is some linear combination of the probabilities of those tests. Below are some of the projection functions found.

$$\begin{aligned}
\Pr(R1R3) &= \Pr(R1) + \Pr(R12) - \Pr(L10) - \Pr(L9) \\
\Pr(L6L5) &= \Pr(L6) - \Pr(R8) + \Pr(L5) \\
\Pr(R10R12) &= \Pr(R10) - \Pr(L7) + \Pr(R9) - \Pr(L6) + \Pr(R8) - \Pr(L5)
\end{aligned}$$

Note that for at least these very short tests, the prediction is a function of a very small fraction of the available core test predictions. In this case, only one-step core tests are used. Furthermore, the coefficients are all 1 or  $-1$ . Each projection takes the form of the probability of the first step in the test, adjusted by adding and subtracting other predictions. However, this does not imply that these projection functions would be easy to generate. There is no readily apparent scheme for determine what adjustments to apply.

## 4 Conclusion

This report describes a brief attempt to extend PSRs to larger, more convincing domains. Even though we did not set out to tackle the core test discovery problem, the complexity of the domain precludes manual core test generation. We could not conduct any further meaningful work without a reasonable set of core tests, and the only method we currently have for obtaining core tests relies on essentially reducing the domain to a finite state POMDP. While we could have pursued some more complex finite versions of this localization task, we would have had to confine the agent essentially to a gridworld to prevent the unbounded number of states possible with a larger selection of turn angles. We eventually reached the conclusion that it is simply not clear at this time how to scale PSRs in the desired direction.

## References

- [1] Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems 14*, pages 1555–1561, 2002.
- [2] Satinder Singh, Michael L. Littman, Nicholas K. Jong, David Pardoe, and Peter Stone. Learning predictive state representations. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 712–719, 2003.