Closing the Learning-Planning Loop with PSRs

Byron Boots BEB@CS.CMU.EDU
Sajid M. Siddiqi* SIDDIQI@GOOGLE.COM
Geoffrey J. Gordon GGORDON@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh PA 15213

Planning a sequence of actions or a policy to maximize future reward has long been considered a fundamental problem for autonomous agents. *Predictive State Representations (PSRs)* are generalizations of *Partially Observable Markov Decision Processes* (POMDPs) that have attracted interest because they both have greater representational capacity than POMDPs and yield representations that are *at least* as compact. The quality of an optimized policy for a POMDP or PSR depends strongly on the accuracy of the model: inaccurate models usually lead to useless plans. We can specify a model manually or learn one from data, but due to the difficulty of learning, it is far more common to see planning algorithms applied to manually-specified models. Unfortunately, it is usually only possible to hand-specify accurate models for small systems where there is extensive and goal-relevant domain knowledge. For example, recent extensions of approximate planning techniques for PSRs have only been applied to models constructed by hand. For the most part, learning models for planning in partially observable environments has been hampered by the inaccuracy of learning algorithms. For example, Expectation-Maximization (EM) does not avoid local minima or scale to large state spaces; and, although many learning algorithms have been proposed for PSRs that attempt to take advantage of the observability of the state representation, few of these have have strong theoretical guarantees and none have been shown to learn models that are accurate enough for planning. As a result, there have been few successful attempts at learning a model directly from data and then closing the loop by planning in that model.

We propose a principled and provably statistically consistent model-learning algorithm, and demonstrate positive results on a challenging high-dimensional problem with continuous observations. In particular, we propose a novel, consistent spectral algorithm for learning a variant of PSRs called *Transformed PSRs* (TPSRs) directly from execution traces. The algorithm is closely related to subspace identification for learning linear dynamical systems and spectral algorithms for learning Hidden Markov Models. We then demonstrate that this algorithm is able to learn compact models of a difficult, realistic dynamical system without any prior domain knowledge built into the model or algorithm (Figure 1). Finally, we perform point-based approximate value iteration in the learned compact models, and demonstrate that the greedy policy for the resulting value function works well in the original (not the learned) system. To our knowledge this is the first research that combines all of these achievements, closing the loop from observations to actions in an unknown domain with no human intervention beyond collecting the raw transition data.

Acknowledgements: SMS was supported by the NSF under grant number 0000164, the USAF under grant number FA8650-05-C-7264, the USDA under grant number 4400161514, and a project with MobileFusion/TTC. BEB was supported by the NSF under grant number EEEC-0540865. BEB and GJG were both supported by ONR MURI grant number N00014-09-1-1052. *SMS is now at Google Pittsburgh.

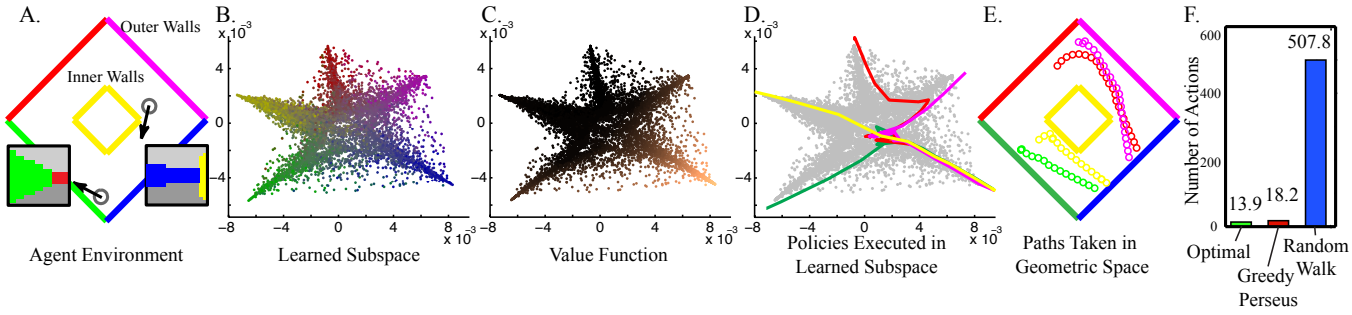

Figure 1. (A) The Simulated Robot Domain. The robot uses visual sensing to traverse a square domain with multi-colored walls and a central obstacle. Examples of images recorded by a robot occupying two different positions in the environment are shown at the bottom. (B) The learned subspace. Each point is the embedding of a single TPSR history, displayed with color equal to the average RGB color in the first image in the highest probability test. (C) The robot was given high reward for observing a particular image (facing blue wall). The value function computed for each embedded point is shown; lighter indicates higher value. (D) Policies executed in the learned subspace. The red, green, magenta, and yellow paths correspond to the policy executed by an agent with starting positions facing the red, green, magenta, and yellow walls respectively. (E) The paths taken by the robot in geometric space while executing the policy. Each of the paths corresponds to the path of the same color in (D). (F) Mean number of actions in path from 100 randomly sampled start position to the target image. The robot was able to reach the goal in 78 trials. In 22 trials the robot got stuck repeatedly taking actions whose effects cancelled. The left bar is the mean number of actions in the optimal solution found by A* search in the robot's configuration space. The center bar is the mean number of actions taken by executing the policy computed by approximate value iteration in the learned model (computed for the 78 *successful* paths). The right bar is the mean number of actions required to find the target with a random policy. The graph indicates that the policy computed from the learned TPSR is close to optimal.