Closing the Learning-Planning Loop with Predictive State Representations

(Extended Abstract)

Byron Boots
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
beb@cs.cmu.edu

Sajid M. Siddiqi*
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
siddiqi@google.com

Geoffrey J. Gordon
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
ggordon@cs.cmu.edu

ABSTRACT

A central problem in artificial intelligence is to plan to maximize future reward under uncertainty in a partially observable environment. Models of such environments include *Partially Observable Markov Decision Processes* (POMDPs) [4] as well as their generalizations, *Predictive State Representations (PSRs)* [9] and *Observable Operator Models (OOMs)* [7]. POMDPs model the state of the world as a latent variable; in contrast, PSRs and OOMs represent state by tracking occurrence probabilities of a set of future events (called tests or characteristic events) conditioned on past events (called histories or indicative events). Unfortunately, exact planning algorithms such as *value iteration* [14] are intractable for most realistic POMDPs due to the *curse of history* and the *curse of dimensionality* [11]. However, PSRs and OOMs hold the promise of mitigating both of these curses: first, many successful approximate planning techniques designed to address these problems in POMDPs can easily be adapted to PSRs and OOMs [8, 6]. Second, PSRs and OOMs are often more compact than their corresponding POMDPs (i.e., need fewer state dimensions), mitigating the curse of dimensionality. Finally, since tests and histories are observable quantities, it has been suggested that PSRs and OOMs should be easier to learn than POMDPs; with a successful learning algorithm, we can look for a model which ignores all but the most important components of state, reducing dimensionality still further.

In this paper we take an important step toward realizing the above hopes. In particular, we propose and demonstrate a fast and statistically consistent spectral algorithm which learns the parameters of a PSR directly from sequences of action-observation pairs. We then *close the loop* from observations to actions by planning in the learned model and recovering a policy which is near-optimal in the *original* environment. Closing the loop is a much more stringent test than simply checking short-term prediction accuracy, since the quality of an optimized policy depends strongly on the accuracy of the model: inaccurate models typically lead to useless plans.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence

*now at Google Pittsburgh

Cite as: Closing the Learning-Planning Loop with Predictive State Representations (Extended Abstract), Byron Boots, Sajid M. Siddiqi and Geoffrey J. Gordon, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. XXX-XXX.

General Terms

Algorithms, Theory

Keywords

Machine Learning, Predictive State Representations, Planning

Closing the Learning-Planning Loop with PSRs

We propose a novel algorithm for learning a variant of PSRs [12] directly from execution traces. Our algorithm is closely related to subspace identification for linear dynamical systems (LDSs) [15] and spectral algorithms for Hidden Markov Models (HMMs) [5] and reduced-rank HMMs [13]. We then use the algorithm to learn a model of a simulated high-dimensional, vision-based mobile robot planning task, and compute a policy by approximate point-based planning in the learned model [6]. Finally, we show that the learned state space compactly captures the essential features of the environment, allows accurate prediction, and enables successful and efficient planning.

By comparison, previous POMDP learners such as Expectation-Maximization (EM) [1] do not avoid local minima or scale to large state spaces; recent extensions of approximate planning techniques for PSRs have only been applied to models constructed by hand [8, 6]; and, although many learning algorithms have been proposed for PSRs (e.g. [16, 3]) and OOMs (e.g. [10]), none have been shown to learn models that are accurate enough for lookahead planning. As a result, there have been few successful attempts at closing the loop.

Our learning algorithm starts from $P_{\mathcal{H}}$, $P_{\mathcal{T},\mathcal{H}}$, and $P_{\mathcal{T},ao,\mathcal{H}}$, matrices of probabilities of one-, two-, and three-tuples of observations conditioned on present and future actions. (For additional details see [2].) We show that, for a PSR with true parameters m_1, m_∞, and M_{ao} (the initial state, the normalization vector, and a transition matrix for each action-observation pair), the matrices $P_{\mathcal{T},\mathcal{H}}$ and $P_{\mathcal{T},ao,\mathcal{H}}$ are low-rank, and can be factored using smaller matrices of test predictions R and S:

$$P_{\mathcal{T},\mathcal{H}} = RS\mathrm{diag}(P_{\mathcal{H}}) \tag{1a}$$

$$P_{\mathcal{T},ao,\mathcal{H}} = RM_{ao}S\mathrm{diag}(P_{\mathcal{H}}) \tag{1b}$$

Next we prove that the true PSR parameters may be recovered, up to a linear transform, from the above matrices and an additional matrix U that obeys the condition that $U^\mathsf{T}R$ is invertible:

$$b_1 \equiv U^\mathsf{T}P_{\mathcal{T},\mathcal{H}}1_k = (U^\mathsf{T}R)m_1 \tag{2a}$$

$$b_\infty^\mathsf{T} \equiv P_{\mathcal{H}}^\mathsf{T}(U^\mathsf{T}P_{\mathcal{T},\mathcal{H}})^\dagger = m_\infty^\mathsf{T}(U^\mathsf{T}R)^{-1} \tag{2b}$$

$$B_{ao} \equiv U^\mathsf{T}P_{\mathcal{T},ao,\mathcal{H}}(U^\mathsf{T}P_{\mathcal{T},\mathcal{H}})^\dagger = (U^\mathsf{T}R)M_{ao}(U^\mathsf{T}R)^{-1} \tag{2c}$$

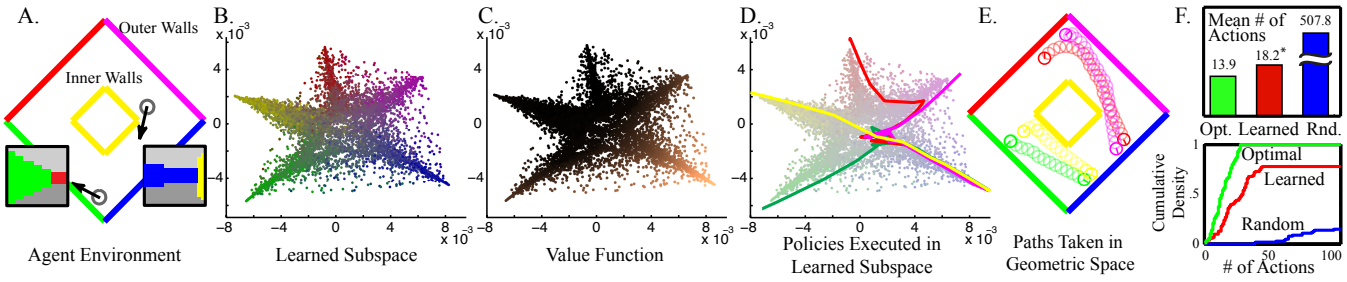

Figure 1: Experimental results. (A) Simulated robot domain and sample images from two positions. (B) Training histories embedded into learned subspace. (C) Value function at training histories (lighter indicates higher value). (D) Paths executed in learned subspace. (E) Corresponding paths in original environment. (F) Performance analysis.

Our learning algorithm works by building empirical estimates $\widehat{P}_{\mathcal{H}}$, $\widehat{P}_{\mathcal{T},\mathcal{H}}$, and $\widehat{P}_{\mathcal{T},ao,\mathcal{H}}$ of $P_{\mathcal{H}}$, $P_{\mathcal{T},\mathcal{H}}$, and $P_{\mathcal{T},ao,\mathcal{H}}$ by repeatedly sampling execution traces of an agent interacting with an environment. We then pick $\widehat{U}$ by singular value decomposition of $\widehat{P}_{\mathcal{T},\mathcal{H}}$, and learn the transformed PSR parameters by plugging $\widehat{U}$, $\widehat{P}_{\mathcal{H}}$, $\widehat{P}_{\mathcal{T},\mathcal{H}}$, and $\widehat{P}_{\mathcal{T},ao,\mathcal{H}}$ into Eq. 2. As we include more data in our estimates $\widehat{P}_{\mathcal{H}}$, $\widehat{P}_{\mathcal{T},\mathcal{H}}$, and $\widehat{P}_{\mathcal{T},ao,\mathcal{H}}$, the law of large numbers guarantees that they converge to their true expectations. So, if our system is truly a PSR of finite rank, the resulting parameters $\widehat{b}_1$, $\widehat{b}_\infty$, and $\widehat{B}_{ao}$ converge to the true parameters of the PSR up to a linear transform—that is, our learning algorithm is *consistent*.

Fig. 1 shows our experimental domain and results. A simulated robot uses visual sensing to traverse a square domain with multi-colored walls and a central obstacle (1A). We collect data by running short trajectories from random starting points, and then learn a PSR. We visualize the learned state space by plotting a projection of the learned state for each history in our training data (1B), with color equal to the average RGB color in the first image in the highest probability test. We give the robot high reward for observing a particular image (facing the blue wall), and plan using point-based value iteration; (1C) shows the resulting value function. To demonstrate the corresponding greedy policy, we started the robot at four positions (facing the red, green, magenta, and yellow walls); (1D) and (1E) show the resulting paths in the state space and in the original environment (in red, green, magenta, and yellow, respectively). Note that the robot *cannot observe* its position in the original environment, yet the paths in E still appear near-optimal. To support this intuition, we sampled 100 random start positions and recorded statistics of the resulting greedy trajectories (1F): the bar graph compares the mean number of actions taken by the optimal solution found by A* search in configuration space (left) to the greedy policy (center; the asterisk indicates that this mean was only computed over the 78 *successful* paths) and to a random policy (right). The line graph illustrates the cumulative density of the number of actions given the optimal, learned, and random policies.

To our knowledge this is the first research to combine several benefits which have not previously appeared together: our learner is computationally efficient and statistically consistent; it handles high-dimensional observations and long time horizons by working from real-valued features of observation sequences; and finally, our close-the-loop experiments provide an end-to-end practical test. See the long version [2] for further details.

Acknowledgements

SMS was supported by the NSF under grant number 0000164, by the USAF under grant number FA8650-05-C-7264, by the USDA under grant number 4400161514, and by a project with MobileFusion/TTC. BB was supported by the NSF under grant number EEEC-0540865. BB and GJG were supported by ONR MURI grant number N00014-09-1-1052.

References

[1] J. Bilmes. A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report, ICSI-TR-97-021, 1997.

[2] B. Boots, S. Siddiqi, and G. Gordon. Closing the learning-planning loop with predictive state representations. *http://arxiv.org/abs/0912.2385*, 2009.

[3] M. Bowling, P. McCracken, M. James, J. Neufeld, and D. Wilkinson. Learning predictive state representations using non-blind policies. In *Proc. ICML*, 2006.

[4] A. R. Cassandra, L. P. Kaelbling, and M. R. Littman. Acting optimally in partially observable stochastic domains. In *Proc. AAAI*, 1994.

[5] D. Hsu, S. Kakade, and T. Zhang. A spectral algorithm for learning hidden Markov models. In *COLT*, 2009.

[6] M. T. Izadi and D. Precup. Point-based planning for predictive state representations. In *Proc. Canadian AI*, 2008.

[7] H. Jaeger. Observable operator models for discrete stochastic time series. *Neural Computation*, 12:1371–1398, 2000.

[8] M. R. James, T. Wessling, and N. A. Vlassis. Improving approximate value iteration using memories and predictive state representations. In *AAAI*, 2006.

[9] M. Littman, R. Sutton, and S. Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[10] M. Zhao and H. Jaeger and M. Thon. A bound on modeling error in observable operator models and an associated learning algorithm. *Neural Computation*, 2009.

[11] J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: an anytime algorithm for POMDPs. In *Proc. IJCAI*, 2003.

[12] M. Rosencrantz, G. J. Gordon, and S. Thrun. Learning low dimensional predictive representations. In *Proc. ICML*, 2004.

[13] S. M. Siddiqi, B. Boots, and G. J. Gordon. Reduced-rank hidden Markov models. *http://arxiv.org/abs/0910.0902*, 2009.

[14] E. J. Sondik. The optimal control of partially observable Markov processes. PhD. Thesis, Stanford University, 1971.

[15] P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer, 1996.

[16] E. Wiewiora. Learning predictive representations from a history. In *Proc. ICML*, 2005.