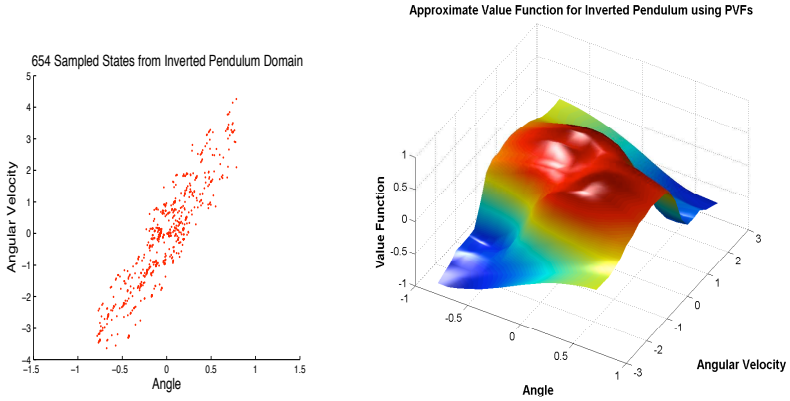
Learning Representation and Control in Markov Decision Processes

Sridhar Mahadevan
Department of Computer Science
University of Massachusetts
140 Governor's Drive
Amherst, MA 01003
(mahadeva)@cs.umass.edu

February 19, 2009

Much past work in control learning, such as approximate dynamic programming [4] or reinforcement learning [5] has assumed that the underlying representation is hand-engineered. In the past several years, we have undertaken a research program exploring a variety of methods for automatically constructing basis functions, applying ideas from manifold learning in machine learning and harmonic analysis. In this paper, we briefly survey a variety of directions that are emerging in the community on frameworks for automatically learning both representation and control in Markov decision processes. The approaches can broadly be divided into two classes of methods: those that construct basis functions that are sensitive to the reward function and the specific policy being followed [3], versus approaches that depend only on the geometry of the state (action) space [2]. These two approaches can both be viewed as constructing representations by analyzing the effect of operators on the space of functions on a state (action) space. Broadly, there are two general principles for constructing representations in this manner: *dilation* methods, which include *Krylov* bases as well as multiscale *wavelet* methods [1], and *diagonalization* methods which work by constructing eigenvectors (or eigenfunctions). Both these approaches essentially can be viewed as finding *invariant* subspaces of the operator.

We will describe an algorithmic framework for simultaneously learning representation and control called *representation policy iteration* (RPI). This framework modifies the traditional policy iteration algorithm by including an outer loop that constructs basis functions that are either reward-sensitive or independent of rewards. A variety of operators can be used to construct basis functions: one approach is to use a graph connecting nearby points on the sampled state (action) space manifold; another approach is to construct an approximation of the Bellman backup operator. Diagonalization methods work with self-adjoint (symmetric) operators. Dilation methods include multiscale diffusion wavelet



h]

Figure 1: Left: Samples from a series of random walks in an inverted pendulum task. Due to physical constraints, the samples are largely confined to a narrow region. One approach to basis construction methods is to diagonalize a random walk operator on a graph connecting nearby samples on such point-sets. Right: An approximation of the value function learned by using PVFs.

methods that construct a hierarchy of basis functions, as well as Krylov methods that dilate the Bellman backup operator. Theoretical properties and experimental results comparing these approaches will be presented. A major challenge for future research is how to scale these approaches to large MDPs (e.g Tetris or backgammon), which have so far only been solved with hand-engineered features.

References

[1] S. Mahadevan and M. Maggioni. Value function approximation with Diffusion Wavelets and Laplacian Eigenfunctions. In *Proceedings of the Neural Information Processing Systems (NIPS)*. MIT Press, 2006.

[2] S. Mahadevan and M. Maggioni. Proto-Value Functions: A Laplacian Framework for Learning Representation and Control in Markov Decision Processes. *Journal of Machine Learning Research*, 8:2169–2231, 2007.

[3] R. Parr, C. Painter-Wakefiled, L. Li, and M. Littman. Analyzing feature generation for value function approximation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 737–744, 2007.

[4] W. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley, 2007.

[5] R. Sutton and A. G. Barto. *An Introduction to Reinforcement Learning*. MIT Press, 1998.