Laplacian “Agent” Learning: Representation Policy Iteration

Sridhar Mahadevan

AUTONOMOUS LEARNING LABORATORY
Example of a Markov Decision Process

What should the agent do?

$V_{a1}("Earth") = f(0,1,1,1,1,...)$

$V_{a2}("Earth") = f(100,-1,-1,-1,-1,...)$
Infinite Horizon Markov Decision Processes

- **Graded myopia:** "discounted sum of rewards"
  - Maximize \( \sum_t \gamma^t r_t \)
  - Hell if \( \gamma < 0.98 \), otherwise Heaven

- **Maximize "average-adjusted sum of rewards"**
  \[
  \lim_{n \to \infty} \frac{\sum_{t=1}^n r_t}{n} = \sum_t \gamma^t r_t
  \]

Always go to heaven!
Sample MDP

\[ V_b(1) = 0.5 \times 1 + 0.5 \times 2 + \gamma (0.5) V_b(1) + \gamma (0.5) V_b(2) \]

\[ V_b(2) = 1 + \gamma V_b(2) \]
Sample MDP

\[ V_b(2) = \frac{1}{1 - \gamma} = 2 \quad \text{if} \quad \gamma = 0.5 \]

\[ V_b(1) = \frac{1.5 + \gamma (0.5) V_b(2)}{1 - \gamma (0.5)} = \frac{8}{3} \]
Sample MDP

\[ V_r(2) = \frac{1}{1 - \gamma} = 2 \quad \text{(if} \quad \gamma = 0.5) \]
\[ V_r(1) = \frac{1 + \gamma (0.5)V_r(2)}{1 - \gamma(0.5)} = 2 \]

Hence, the blue policy is optimal in state 1
Bellman Equation

\[ V^*(x) = \max_{a \in A(x)} \left( r(x, a) + \gamma \sum_y P_{xy}^a V^*(y) \right) \]
Value Function Approximation

Lspi [Lagoudakis and Parr, 2003]
Inverted Pendulum with Radial Basis Functions (10)
VFA by Least-Squares Projection: Minimize Bellman Residual Error

\[ \hat{V}^\pi \approx T^\pi (\hat{V}^\pi) \]

\[ \hat{V}(s) = \sum_i \phi_i(s)w_i \]

Subspace \( \Phi \)

(poly, RBF, neural nets)
VFA by Least-Squares Projection: Fixed Point Projection

\[ T^\pi(\hat{V}(s)) \]

\[ \hat{V}^\pi \approx \Phi (\Phi^T \Phi)^{-1} \Phi^T T^\pi(\hat{V}^\pi) \]

\[ \hat{V}(s) = \sum_i \phi_i(s) w_i \]
Parametric Value Function Approximation Can Fail

Approximator blind to symmetries!

LSPI converges to an incorrect policy

Goal is to get to center of square grid

Approximator blind to bottlenecks!

[Drummond, JAIR 2002]
Divergence of Neural Nets in Mountain Car Task
(Boyan and Moore, NIPS 1995)

Figure 5: The car-on-the-hill domain. When the velocity is below a threshold, the car must reverse up the left hill to gain enough speed to reach the goal, so $J^*$ is discontinuous.
Automating Value Function Approximation

- Many approaches to value function approximation
  - Neural nets, radial basis functions, support vector machines, kernel density estimation, nearest neighbor
- How to automate the design of a function approximator?
- We want to go beyond model selection!
Representation Learning using Harmonic Analysis on Manifolds
(Mahadevan and Maggioni, NIPS 2005)

Fourier

\[ \nabla^2 (F) = 0 \]
\[ \text{Div(Grad(F))} = 0 \]

Wavelets

Dilations of diffusion operator on manifold

Inverted pendulum

Samples from random walk
Proto-Value Functions
(Mahadevan, ICML 2005)

Proto-value functions are the representational building blocks of all value functions on a given environment.

Proto-value functions are based on modeling the state space manifold.

Fourier

Wavelet
How are Proto-Value Functions Learned?

Proto-value functions

Graph Laplacian
Representation Policy Iteration
(Mahadevan, UAI 2005)

Policy improvement

“Greedy” Policy

Representation Learner

Trajectories

“Actor”

Policy evaluation

“Critic”

Proto-value functions
Policy Iteration
(Howard, 1960)

Policy Evaluation: \( V^\pi(x) = r(x, \pi(x)) + \gamma \sum_y P_{xy}^{\pi(x)} V^\pi(y) \)

Policy Improvement: \( \pi'(x) = \arg\max_a \left( r(x, a) + \gamma \sum_y P_{xy}^a V^\pi(y) \right) \)
Representation Policy Iteration

- Learn a set of proto-value functions from a sample of transitions generated from a random walk (or from watching an expert).

- These basis functions can then be used in an approximate policy iteration algorithm, such as Least Squares Policy Iteration [Lagoudakis and Parr, 2003].
Least-Squares Policy Iteration
(Boyan, ICML 1999; Lagoudakis and Parr, JMLR 2003)

Do a random walk generating a set of transitions \( D = (s_t, a_t, r, s'_t) \)

\[
\tilde{A}^{t+1} = \tilde{A}^t + \phi(s_t, a_t) (\phi(s_t, a_t) - \gamma \phi(s'_t, \pi(s'_t)))^T \\
\tilde{b}^{t+1} = \tilde{b}^t + \phi(s_t, a_t)r_t
\]

Solve the equation:

\[
\tilde{A} w^\pi = \tilde{b}
\]

\[
Q^\pi(x, a) \approx \sum_{i=1}^{k} \phi(s, a)w_i^\pi
\]
Learned vs. Handcoded Representations: Chain MDP
Results of RPI on a Grid World

Goal is to get to center of square grid
Results of RPI on Two Room World

Nonlinearity due to bottleneck is nicely captured by RPI!
RPI in Continuous State Spaces
(Mahadevan, Maggioni, Ferguson, Osentoski, 2006)

- To extend RPI to continuous state spaces, we need to use the Nystrom extension to extend eigenfunctions from sample points to new points.
- Many issues are involved here, including how many sampled points to choose from, how to construct the graph to reflect the underlying manifold, etc.
RPI: Inverted Pendulum Task

Sample transitions from random walk (~800-1600)

Normalized Graph

Laplacian matrix

Proto value functions
RPI: Inverted Pendulum

Learned Proto-Value Functions

Handcoded Radial Basis Functions
RPI Results on Inverted Pendulum
(Mahadevan, Maggioni, Ferguson, Osentoski, 2006)

Pendulum: Proto-Value Functions vs Radial Basis Functions

Steps vs Number of training episodes

- 10 RBF
- 50 PVF
- 36 RBF
- 50 RBF
RPI Results for Mountain Car
(Mahadevan, Maggioni, Ferguson, Osentoski, 2006)
Summary: Proto-Value Functions

- A major challenge facing MDP research is to how to automate value function approximation
- Proto-value functions
  - Fourier and wavelet representations on manifolds that capture large-scale geometry of the state (action) space
  - Representation Policy Iteration is a general framework for simultaneously learning representations and policies
- Extensions of proto-value functions
  - "On-policy" proto-value functions [Maggioni and Mahadevan, 2005]
  - Factored Markov decision processes [Mahadevan, 2006]
  - Group-theoretic extensions [Mahadevan, in preparation]