Pathologies of Approximate Policy Iteration in Dynamic Programming

Dimitri P. Bertsekas

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

March 2011

Summary

- We consider policy iteration with cost function approximation
- Used widely but exhibits very complex behavior and a variety of potential pathologies
- Case of the tetris test problem
- Two types of pathologies
  - Deterministic: Due to cost function approximation
  - Stochastic: Due to simulation errors/noise
- We survey the pathologies in
  - Policy evaluation: Due to errors in approximate evaluation of policies
  - Policy improvement: Due to policy improvement mechanism
- Special focus: Policy oscillations and local attractors
- Causes of the problem in TD/projected equation methods:
  - The projection operator may not be monotone
  - The projection norm may depend on the policy evaluated
- We discuss methods that address the difficulty

References

- D. P. Bertsekas, “Pathologies of Temporal Differences Methods in Approximate Dynamic Programming," Proc. 2010 IEEE Conference on Decision and Control, Proc. 2010 IEEE Conference on Decision and Control, Atlanta, GA.
- D. P. Bertsekas, Dynamic Programming and Optimal Control, Vol. II, 2007, Supplementary Chapter on Approximate DP: On-line; a “living chapter."

MDP: Brief Review

- $J^*(i)$ = Optimal cost starting from state i
- $J_\mu(i)$ = Cost starting from state i using policy μ
- Denote by T and T_μ the DP mappings that transform $J \in \Re^n$ to the vectors TJ and $T_\mu J$ with components

$$(TJ)(i) \stackrel{\text{def}}{=} \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha J(j)\big), \qquad i = 1, \dots, n,$$

$$(T_\mu J)(i) \stackrel{\text{def}}{=} \sum_{j=1}^{n} p_{ij}\big(\mu(i)\big)\big(g(i,\mu(i),j) + \alpha J(j)\big), \qquad i = 1, \dots, n$$

  $\alpha < 1$ for a discounted problem; $\alpha = 1$ and 0-cost termination state for a stochastic shortest path problem

- Bellman's equations have unique solution

$$J^* = TJ^*, \qquad J_\mu = T_\mu J_\mu$$

- μ^* is optimal (i.e., $J^* = J_{\mu^*}$) iff $T_{\mu^*} J^* = TJ^*$

Policy Iteration: Lookup Table Representation

- Policy iteration (exact): Start with any μ
  - Evaluation of policy μ: Find J_μ

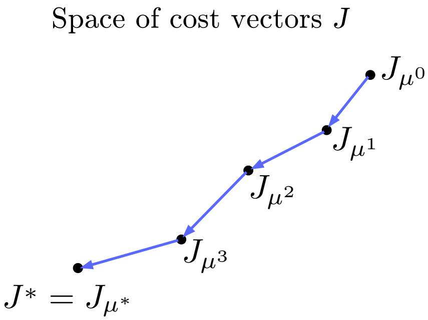
    $$J_\mu = T_\mu J_\mu$$

    A linear equation
  - Improvement of policy μ: Find $\overline{\mu}$ that attains the min in TJ_μ, i.e.,

    $$T_{\overline{\mu}} J_\mu = T J_\mu$$

- Policy iteration converges finitely (if exact)

Illustration of Convergence

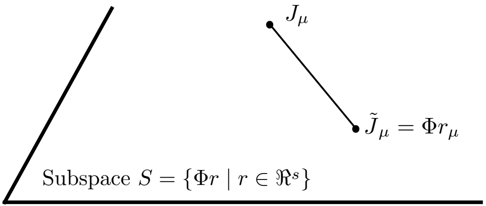


With exact policy evaluation, convergence is finite and monotonic

Policy Iteration: Cost Function Approximation

- An old, time-tested approach for solving large-scale equation problems
- Approximation within subspace $S = \{\Phi r \mid r \in \Re^s\}$

$$J \approx \Phi r, \qquad \Phi \text{ is a matrix with basis functions/features as columns}$$



- Instead of J_μ, find $\tilde{J}_\mu = \Phi r \in S$ by some form of “projection" onto S

$$\tilde{J}_\mu = WT_\mu(\tilde{J}_\mu) \qquad \text{or equivalently} \qquad \Phi r_\mu = WT_\mu(\Phi r_\mu)$$

- Example: A projected equation/Galerkin method: $W = \Pi$ (a Euclidean projection)
- Example: An aggregation method: $W = \Phi D$, where Φ (aggregation matrix) and D (disaggregation matrix) have prob. distributions as rows

Approximate Policy Iteration

- Start with any μ
  - Evaluation of policy μ: Solve for $\tilde{J}_\mu$ the linear equation

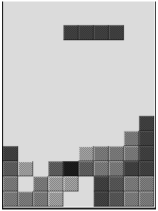
$$\tilde{J}_\mu = W T_\mu(\tilde{J}_\mu)$$

  - Improvement of policy μ: Find $\overline{\mu}$ that attains the min in TJ_μ, i.e.,

$$T_{\overline{\mu}}\tilde{J}_\mu = T\tilde{J}_\mu$$

- Special twists that originated in Reinforcement Learning/ADP:
  - Policy evaluation can be done by simulation, with low-dimensional linear algebra
  - Matrix inversion method LSTD(λ), or iterative methods such as LSPE(λ), TD(λ), λ-policy iteration, etc
  - Similar aggregation methods

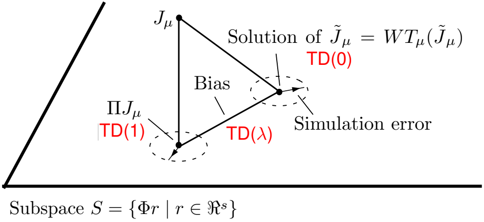
Tetris Case Study



- Classical and challenging test problem with huge number of states
- Initial policy iteration work (VanRoy MS Thesis, under J. Tsitsiklis, 1993) - a 10x20 board, 3 basis functions, average score of $\approx$ 40 points
- Most studies have used a 10x20 board, and a set of “standard" 22 basis functions introduced by Bertsekas and Ioffe (1996)
- Approximate policy iteration [B+I (1996), Lagoudakis and Parr (2003)]
- Policy gradient method [Kakade (2002)]
- Approximate LP [Farias+VanRoy (2006), Desai+Farias+Moallemi (2009)]
- All of the above achieved average scores in the range 3,000-6,000
- BUT with a random search method Szita and Lorenz (2006), and Thierry and Sherrer (2009) achieved scores 600,000-900,000

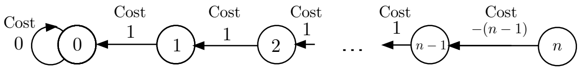
Potential Pathologies

- General issue:
  - Good cost approximation $\Longrightarrow$ good performance of generated policies??
  - Bad cost approximation $\Longrightarrow$ bad performance of generated policies??
    (Can add a constant to the cost of all states without affecting the next generated policy)
- Policy evaluation issues (both can be quantified to some extent)
  - Bias
  - Simulation error/noise



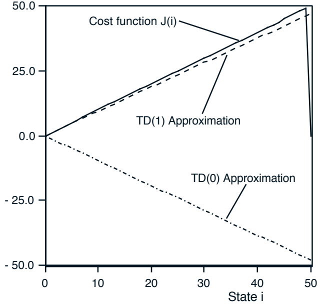
- Policy iteration issues (hard to quantify and understand)
  - Oscillations of policies (local attractors; like local minima)
  - Exploration (simulation must ensure that all parts of the state space are adequately sampled/explored)

Policy Evaluation - Bias Issues - An Example

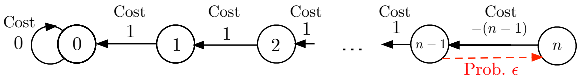


- Stochastic shortest path problem with 0: termination state (from Bertsekas 1995; Neural Computation, Vol. 7)
- Consider a linear approximation of the form

$$\tilde{J}_\mu(i) = i\, r$$

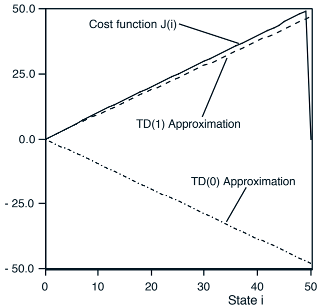


Policy Evaluation - Bias Issues - An Example



- Consider a linear approximation of the form

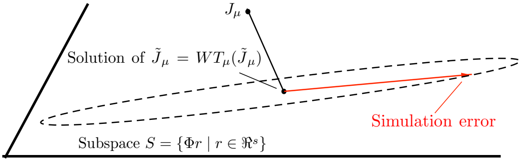
$$\tilde{J}_\mu(i) = i\, r$$



- A strange twist: Introduce an ϵ-probability reverse decision at state $n-1$
  - Policy iteration/TD(0) yields the optimal policy
  - Policy iteration/TD(1) does not

Policy Evaluation - Sensitivity to Simulation Noise

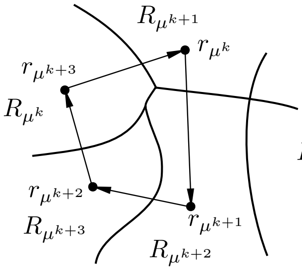
- Consider the evaluation equation $\Phi r = WT_\mu(\Phi r)$
- It is equivalent to a linear equation $Cr = d$ with C a positive definite (nonsymmetric) matrix
- In popular approaches, we compute by simulation $\tilde{C} \approx C$ and $\tilde{d} \approx d$
- The solution $\Phi\tilde{r} = \Phi\tilde{C}^{-1}\tilde{d}$ may be highly sensitive to simulation error



- This necessitates lots of sampling ... confidence interval/convergence rate analysis needed (Konda Ph.D. Thesis 2002)
- Can happen even without subspace approximation/lookup table representation ($S = \Re^n$)
- Regularization methods may be used, but they introduce additional bias ... need to quantify

Policy Improvement - Oscillations

- Consider the space of weights r (policy μ is evaluated as $\tilde{J}_\mu = \Phi r_\mu$)
- R_μ = set of r for which μ is greedy: $T_\mu(\Phi r) = T(\Phi r)$ (Greedy Partition)
- μ improves to $\overline{\mu}$ iff $r_\mu \in R_{\overline{\mu}}$



Space of weights r

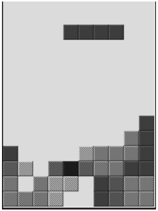
$$R_\mu = \{ r \mid T_\mu(\Phi r) = T(\Phi r) \}$$

- The algorithm ends up repeating a cycle of policies $\mu^k, \mu^{k+1}, \ldots, \mu^{k+m}$:

$$r_{\mu^k} \in R_{\mu^{k+1}},\ r_{\mu^{k+1}} \in R_{\mu^{k+2}}, \ldots, r_{\mu^{k+m-1}} \in R_{\mu^{k+m}},\ r_{\mu^{k+m}} \in R_{\mu^k}$$

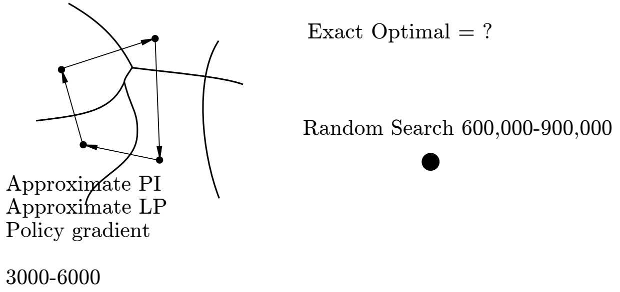
- The greedy partition depends only on Φ - is independent of the policy evaluation method used

Back to Tetris



- 10x20 board, set of “standard" 22 basis functions
- Approximate policy iteration [Bertsekas and Ioffe (1996), Lagoudakis and Parr (2003)]
- Approximate LP [Farias+VanRoy (2006), Desai+Farias+Moallemi (2009)]
- Policy gradient method [Kakade (2002)]
- All of the above achieved average scores in the range 3,000-6,000
- BUT with a random search method Szita and Lorenz (2006), and Thierry and Sherrer (2009) achieved scores 600,000-900,000

What's Going on in Tetris?



- Based on tests with a smaller board: Oscillations occur often in "bad parts of the weight space". Not clear if oscillations are the problem
- Random search and well-designed aggregation methods achieve a score very close to the exact optimal
- The basis functions are very powerful (approx. optimal $\approx$ exact optimal)
- Starting from an excellent weight vector, approximate policy iteration drifts off to cycle around a significantly inferior weight vector
- Starting from a bad weight vector, approximate policy iteration drifts off to cycle around a better but not good weight vector

Search for Remedies

- Consider again approximation within subspace $S = \{\Phi r \mid r \in \Re^s\}$
- Problem with oscillations: Projection is not monotone (also depends on μ)
- Remedy: Replace projection by a constant monotone operator W with range S
- Policy evaluation using an approximate Bellman equation: Find $\tilde{J}_\mu$ with

$$\tilde{J}_\mu = WT_\mu(\tilde{J}_\mu) \qquad \text{instead of} \qquad \tilde{J}_\mu = \Pi T_\mu(\tilde{J}_\mu)$$

- Policy iteration (approximate): Start with any μ
  - Evaluation of policy μ: Solve for $\tilde{J}_\mu$ the equation

$$\tilde{J}_\mu = WT_\mu(\tilde{J}_\mu)$$

  - Improvement of policy μ: Find $\overline{\mu}$ that attains the min in TJ_μ, i.e.,

$$T_{\overline{\mu}}\tilde{J}_\mu = T\tilde{J}_\mu$$

Conditions for Convergence

- Convergence Result: Assume the following:

  (a) W is monotone: $WJ \leq WJ'$ for any two $J, J' \in \Re^n$ with $J \leq J'$

  (b) For each μ, WT_μ is a contraction

  (c) Termination when $\overline{\mu}$ is obtained such that $T_{\overline{\mu}}\tilde{J}_{\overline{\mu}} = T\tilde{J}_{\overline{\mu}}$

  Then the method terminates in a finite number of iterations, and the cost vector obtained upon termination is a fixed point of WT.

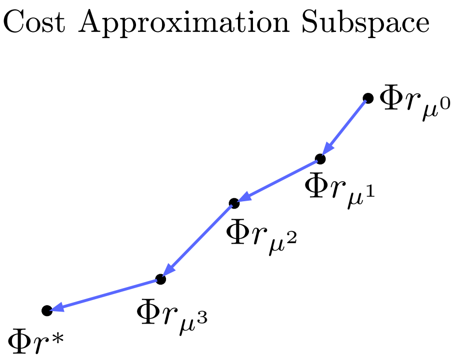
- Proof is similar to classical proof of convergence of exact policy iteration

- Contraction assumption can be weakened: For all J such that $(WT_\mu)(J) \leq J$, we must have

$$\tilde{J}_\mu = \lim_{k\to\infty} (WT_\mu)^k(J)$$

  More general DP models can be accommodated.

Convergence within the Approximation Subspace



Convergence is finite and monotonic ... but how good is the limit?

Methods for Selecting W

- Aggregation: $W = \Phi D$ with rows of Φ and D being probability distributions (this is a serious restriction)
- Hard aggregation is an interesting special case: Then W is also a projection
- Another approach: No restriction on Φ (advantage when we have a desirable Φ)
  - "Double" the number of columns so that $\Phi \geq 0$ (separate $+$ and $-$ parts of the columns)
  - Let $W = \Phi D$. Choose W by some optimization criterion subject to $D \geq 0$ and W (sup-norm) nonexpansive, i.e.,

    $$\phi(i)'\zeta \leq 1, \qquad \forall \text{ states } i,$$

    where $\phi(i)'$ is the ith row of Φ, and ζ is the vector of row sums of D.
- A special possibility: Start with $\Phi \geq 0$, and use

  $$W = \gamma\, \Phi M^{-1} \Phi' \Xi,$$

  where $\gamma \approx 1$ and M is a (constant) positive definite diagonal replacement of $\Phi' \Xi \Phi$ in the projection formula

  $$\Pi = \Phi(\Phi' \Xi \Phi)^{-1} \Phi' \Xi$$

Some Perspective

- There are several pathologies in approximate PI ... How bad is that?
- Other methods have pathologies, e.g., gradient methods that may be attracted to local minima.
- This does not mean that they are not useful ...
- ... BUT in approximate PI the pathologies are many and diverse
- ... makes it hard to know what went wrong
- Other approximate DP methods also have their own pathologies
- Need better understanding of the pathologies, how to fix them and how to detect them
- What's going on in tetris?