Topics in Reinforcement Learning: Rollout and Approximate Policy Iteration

ASU, CSE 691, Spring 2021

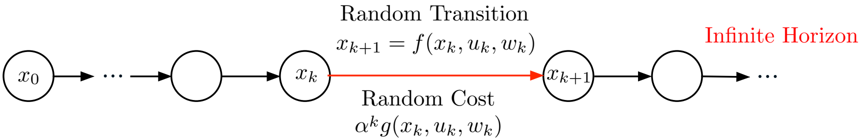
Links to Class Notes, Videolectures, and Slides at
http://web.mit.edu/dimitrib/www/RLbook.html

Dimitri P. Bertsekas
dbertsek@asu.edu

Lecture 9
Infinite Horizon Problems: Theory and Algorithms

Outline

Infinite Horizon Problems



Infinite number of stages, and stationary system and cost

- System $x_{k+1} = f(x_k, u_k, w_k)$ with state, control, and random disturbance
- Stationary policies μ with $\mu(x) \in U(x)$ for all x
- Cost of stage k: $\alpha^k g\big(x_k, \mu(x_k), w_k\big)$
- Cost of a policy μ: The limit as $N \to \infty$ of the N-stage costs

$$J_\mu(x_0) = \lim_{N\to\infty} E_{w_k}\left\{\sum_{k=0}^{N-1} \alpha^k g\big(x_k, \mu(x_k), w_k\big)\right\}$$

- Optimal cost function $J^*(x_0) = \min_\mu J_\mu(x_0)$
- $0 < \alpha \leq 1$ is the discount factor. If $\alpha < 1$ the problem is called Discounted
- Problems with $\alpha = 1$ typically include a special cost-free termination state t and are called Stochastic Shortest Path (SSP) problems.

Transition Probability Notation for Finite-State Problems

- States: $x = 1, \ldots, n$. Successor states: y. (For SSP there is also the extra termination state t.)
- Probability of $x \to y$ transition under control u: $p_{xy}(u)$
- Cost of $x \to y$ transition under control u: $g(x, u, y)$

Going from one notation system to the other (discounted case):

- Replace $x_{k+1} = f(x_k, u_k, w_k)$ with $x_{k+1} = w_k$ (a simpler system)
- Replace $P(w \mid x, u)$ with $p_{xy}(u)$ (a 3-dimensional matrix)
- Replace cost per stage $E\{g(x, u, w)\}$ with $\sum_{y=1}^{n} p_{xy}(u) g(x, u, y)$
- Replace cost-to-go $E\Big\{J\big(f(x, u, w)\big)\Big\}$ with $\sum_{y=1}^{n} p_{xy}(u) J(y)$

Example: Bellman equation (translated to the new notation)

$$J^*(x) = \min_{u \in U(x)} \sum_{y=1}^{n} p_{xy}(u)\big(g(x, u, y) + \alpha J^*(y)\big) \qquad \text{(for Discounted)}$$

$$J^*(x) = \min_{u \in U(x)} \left[p_{xt}(u) g(x, u, t) + \sum_{y=1}^{n} p_{xy}(u)\big(g(x, u, y) + J^*(y)\big) \right] \qquad \text{(for SSP)}$$

The Three Theorems for Discounted Problems: If $g(x, u, y)$ is Bounded the Entire Exact Theory Goes Through with No Exceptions

1) VI convergence: $J_k(x) \to J^*(x)$ for all J_0, where:

$$J_{k+1}(x) = \min_{u \in U(x)} \left[\sum_{y=1}^{n} p_{xy}(u) \big(g(x, u, y) + \alpha J_k(y) \big) \right]$$

2) J^* satisfies uniquely Bellman's equation

$$J^*(x) = \min_{u \in U(x)} \left[\sum_{y=1}^{n} p_{xy}(u) \big(g(x, u, y) + \alpha J^*(y) \big) \right], \qquad x = 1, \ldots, n$$

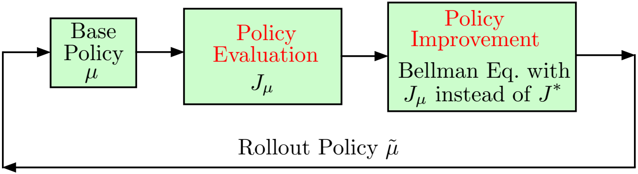
3) Optimality condition

A stationary policy μ is optimal if and only if $\mu(x)$ attains the minimum for every state x.

Also J_μ is the unique solution of the Bellman equation (for policy μ)

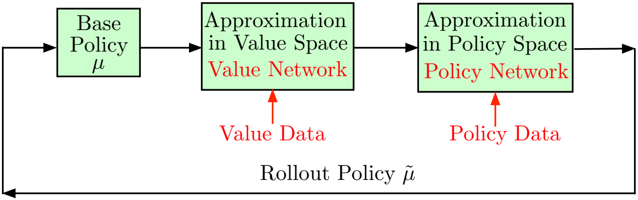
$$J_\mu(x) = \sum_{y=1}^{n} p_{xy}\big(\mu(x)\big) \Big(g\big(x, \mu(x), y\big) + \alpha J_\mu(y) \Big), \qquad x = 1, \ldots, n$$

Exact and Approximate Policy Iteration



Important facts:

- Exact PI yields in the limit an optimal policy
- Exact PI is much faster than VI; it is Newton's method for solving Bellman's Eq.
- Policy evaluation can be implemented by a variety of simulation-based methods. Lots of RL theory (e.g., temporal difference methods)
- PI can be implemented approximately, with a value and/or a policy network



Finite Spaces SSP Problems - Statement of Main Results

Most favorable Assumption (Termination Inevitable Under all Policies)

There exists $m > 0$ such that for every policy and initial state, there is positive probability that t will be reached within m stages

Intuitively: This is really a finite horizon problem, but with random horizon. Easy analysis.

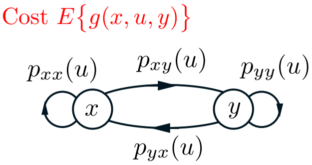
VI Convergence: $J_k \to J^*$ for all initial conditions J_0, where

$$J_{k+1}(x) = \min_{u \in U(x)} \left[p_{xt}(u) g(x, u, t) + \sum_{y=1}^{n} p_{xy}(u) \big(g(x, u, y) + J_k(y) \big) \right], \qquad x = 1, \ldots, n$$

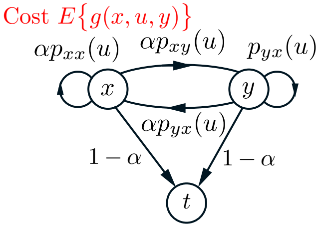
Bellman's equation: J^* satisfies

$$J^*(x) = \min_{u \in U(i)} \left[p_{xt}(u) g(x, u, t) + \sum_{y=1}^{n} p_{xy}(u) \big(g(x, u, y) + J^*(y) \big) \right], \qquad x = 1, \ldots, n,$$

and is the unique solution of this equation.
Optimality condition: μ is optimal if and only if for every x, $\mu(x)$ attains the minimum in the Bellman equation.



Discounted Problem



SSP Equivalent

A discounted problem can be converted to an SSP problem (with termination inevitable)

- Reason: The stage k cost $[\alpha^k E\{g(x,u,y)\}]$ is identical in both problems, under the same policy.
- Proofs for discounted case: Start with SSP analysis, get discounted analysis as special case.
- This line of proof applies to finite-state problems. For infinite-state discounted problems a different line is needed (based on contraction mapping ideas).

SSP Extensions

SSP problems often do not satisfy the “termination inevitable for all policies" assumption (e.g., deterministic SP problems with cycles)

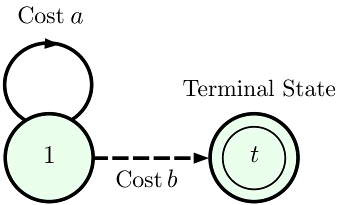
A more general assumption for SSP results: Nonterminating policies are “bad"

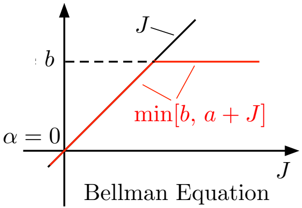
- Every policy that does not terminate with > 0 probability, has ∞ cost for some initial states.
- There exists at least one policy under which termination is inevitable.
- Major results are salvaged under this assumption.

SSP further extensions can be very challenging

- Bellman’s Eq. can have many solutions
- Bellman’s Eq. may have a unique solution that is not equal to J^* (even for finite-state, but stochastic, problems)!!
- VI and PI may fail (even for finite-state problems)
- Infinite-state problems can exhibit “strange" behavior (even with bounded cost per stage)
- See the on-line Abstract DP book (DPB, 2018) for detailed discussion

Working Break: Challenge Questions About a Tricky SSP Problem; see the Abstract DP Book, Section 3.1.1, for More Analysis



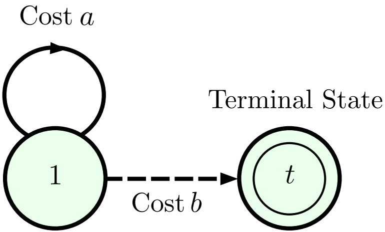


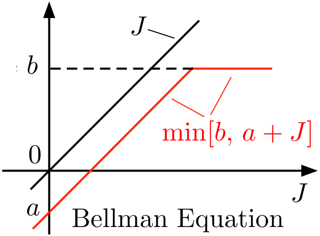
This example violates the “nonterminating policies are bad" assumption for $a = 0$. Then:

- Bellman equation, $J(1) = \min\big[b,\ a + J(1)\big]$, has multiple solutions
- VI converges to J^* from some initial conditions but not from others

Challenge questions: Consider the cases $a > 0$, $a = 0$, and $a < 0$

- What is $J^*(1)$?
- What is the solution set of Bellman's equation?
- What is the limit of the VI algorithm $J_{k+1}(1) = \min\big[b,\ a + J_k(1)\big]$?





Bellman Eq: $J(1) = \min\big[b,\ a + J(1)\big]$; **VI:** $J_{k+1}(1) = \min\big[b,\ a + J_k(1)\big]$

- If $a > 0$ (positive cycle): $J^*(1) = b$ is the unique solution, and VI converges to $J^*(1)$. Here the "nonterminating policies are bad" assumption is satisfied.
- If $a = 0$ (zero cycle):
  - $J^*(1) = \min[0, b]$.
  - Bellman Eq. is $J(1) = \min\big[b,\ J(1)\big]$; its solution set is $= (-\infty, b]$.
  - The VI algorithm, $J_{k+1}(1) = \min\big[b,\ J_k(1)\big]$, converges to b starting from $J_0(1) \geq b$, and does not move from a starting value $J_0(1) \leq b$.
- If $a < 0$ (negative cycle): The Bellman Eq. has no solution, and VI diverges to $J^*(1) = -\infty$.

Approximations to the VI algorithm: Fitted VI

Consider (discounted problem) VI with sequential approximation

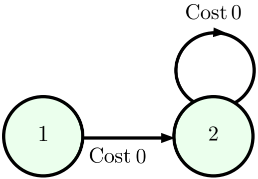
$$J_{k+1}(x) = \min_{u \in U(x)} \sum_{y=1}^{n} p_{xy}(u)\big(g(x,u,y) + \alpha J_k(y)\big) \qquad \text{(VI algorithm)}$$

Approximate version: Assume that for some $\delta > 0$

$$\max_{x=1,\ldots,n} \left| \tilde{J}_{k+1}(x) - \min_{u \in U(x)} \sum_{y=1}^{n} p_{xy}(u)\big(g(x,u,y) + \alpha \tilde{J}_k(y)\big) \right| \leq \delta \tag{1}$$

- Under condition (1), the cost function error $\max_{x=1,\ldots,n} \big|\tilde{J}_k(x) - J^*(x)\big|$ can be shown to be $\leq \delta/(1-\alpha)$ (asymptotically, as $k \to \infty$).
- ... but this result may not be meaningful for some natural methods: It may be difficult to maintain Eq. (1) over an infinite horizon, because $\{\tilde{J}_k\}$ may become unbounded.
- Illustration: Start with $\tilde{J}_0$, and let $\tilde{J}_k$ be obtained using a parametric architecture:
  - Given parametric approximation $\tilde{J}_k$, obtain a parametric approximation $\tilde{J}_{k+1}$ using a least squares fit.
  - We will give an example where the cost function error accumulates to ∞.

Instability of Fitted VI (Tsitsiklis and VanRoy, 1996)

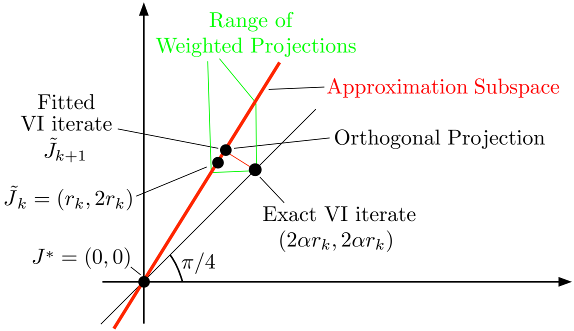


Single policy

Bellman Eq.: $J(1) = \alpha J(2), \; J(2) = \alpha J(2)$

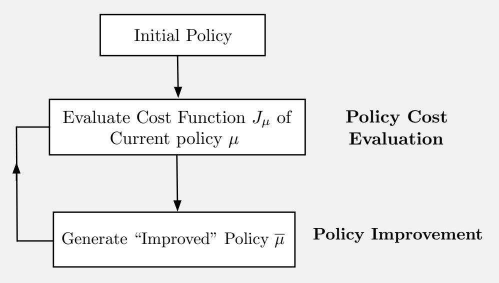
$$J^*(1) = J^*(2) = 0$$

Exact VI : $J_{k+1}(1) = \alpha J_k(2), \; J_{k+1}(2) = \alpha J_k(2)$



By using a weighted projection we may correct the problem. What is the right projection?

Policy Iteration (PI) Algorithm: Generates a Sequence of Policies $\{\mu^k\}$



Given the current policy μ^k, a PI consists of two phases:

- **Policy evaluation** computes $J_{\mu^k}(x)$, $x = 1, \ldots, n$, as the solution of the (linear) Bellman equation system

$$J_{\mu^k}(x) = \sum_{y=1}^{n} p_{xy}\big(\mu^k(x)\big)\Big(g\big(x, \mu^k(x), y\big) + \alpha J_{\mu^k}(y)\Big), \quad x = 1, \ldots, n$$

- **Policy improvement** then computes a new policy μ^{k+1} as

$$\mu^{k+1}(x) \in \arg\min_{u \in U(x)} \sum_{y=1}^{n} p_{xy}(u)\big(g(x, u, y) + \alpha J_{\mu^k}(y)\big), \quad x = 1, \ldots, n$$

Proof of Policy Improvement (Standard Rollout/PI Proof Line)

PI finite convergence: PI generates an improving sequence of policies, i.e., $J_{\mu^{k+1}}(x) \leq J_{\mu^k}(x)$ for all x and k, and terminates with an optimal policy.

Let $\tilde{\mu}$ be the rollout policy obtained from base policy μ: Will show that $J_{\tilde{\mu}} \leq J_\mu$

- Denote by J_N the cost function of a policy that applies $\tilde{\mu}$ for the first N stages and applies μ thereafter.
- We have the Bellman equation $J_\mu(x) = \sum_{y=1}^n p_{xy}\big(\mu(x)\big)\Big(g\big(x,\mu(x),y\big) + \alpha J_\mu(y)\Big)$, so
$$J_1(x) = \sum_{y=1}^n p_{xy}\big(\tilde{\mu}(x)\big)\Big(g\big(x,\tilde{\mu}(x),y\big)+\alpha J_\mu(y)\Big) \leq J_\mu(x) \quad \text{(by policy improvement eq.)}$$
- From the definition of J_2 and J_1, and the preceding relation, we have
$$J_2(x) = \sum_{y=1}^n p_{xy}\big(\tilde{\mu}(x)\big)\Big(g\big(x,\tilde{\mu}(x),y\big)+\alpha J_1(y)\Big) \leq \sum_{y=1}^n p_{xy}\big(\tilde{\mu}(x)\big)\Big(g\big(x,\tilde{\mu}(x),y\big)+\alpha J_\mu(y)\Big)$$
  so $J_2(x) \leq J_1(x) \leq J_\mu(x)$ for all x.
- Continuing similarly, we obtain $J_{N+1}(x) \leq J_N(x) \leq J_\mu(x)$ for all x and N. Since $J_N \to J_{\tilde{\mu}}$ (VI for $\tilde{\mu}$ converges to $J_{\tilde{\mu}}$), it follows that $J_{\tilde{\mu}} \leq J_\mu$.

Optimistic PI - This is Just Repeated Truncated Rollout

Generates sequence of policy-cost function approximation pairs $\{(\mu^k, J_k)\}$

Given the typical pair (μ^k, J_k), do truncated rollout with base policy μ^k and cost approximation J_k:

- Policy evaluation (m_k steps of rollout using μ^k): Starting with $\hat{J}_{k,0} = J_k$, compute $\hat{J}_{k,1}, \ldots, \hat{J}_{k,m_k}$ according to

$$\hat{J}_{k,m+1}(x) = \sum_{y=1}^{n} p_{xy}(\mu^k(x))\Big(g(x, \mu^k(x), y) + \alpha \hat{J}_{k,m}(y)\Big), \qquad x = 1, \ldots, n$$

- Policy improvement (standard): Set

$$\mu^{k+1}(x) \in \arg \min_{u \in U(x)} \sum_{y=1}^{n} p_{xy}(u)\big(g(x, u, y) + \alpha \hat{J}_{k,m_k}(y)\big), \qquad x = 1, \ldots, n,$$

$$J_{k+1}(x) = \min_{u \in U(x)} \sum_{y=1}^{n} p_{xy}(u)\big(g(x, u, y) + \alpha \hat{J}_{k,m_k}(y)\big), \qquad x = 1, \ldots, n.$$

Convergence (using similar argument to standard PI)

Multistep PI: Uses Multistep Lookahead

Given the typical policy μ^k:

- Policy evaluation (standard): Computes $J_{\mu^k}(x)$, $x = 1, \ldots, n$, as the solution of the (linear) Bellman equation

$$J_{\mu^k}(x) = \sum_{y=1}^{n} p_{xy}\big(\mu^k(x)\big)\Big(g\big(x, \mu^k(x), y\big) + \alpha J_{\mu^k}(y)\Big), \quad x = 1, \ldots, n$$

- Policy improvement with ℓ-step lookahead: Solves the ℓ-stage problem with terminal cost function J_{μ^k}. If $\{\hat{\mu}_0, \ldots, \hat{\mu}_{\ell-1}\}$ is the optimal policy of this problem, then the new policy μ^{k+1} is $\hat{\mu}_0$.

Motivation: It may yield a better policy μ^{k+1} than with one-step lookahead, at the expense of a more complex policy improvement operation.

Convergence (using similar argument to standard PI)

Simplified Minimization
Multiagent policy improvement

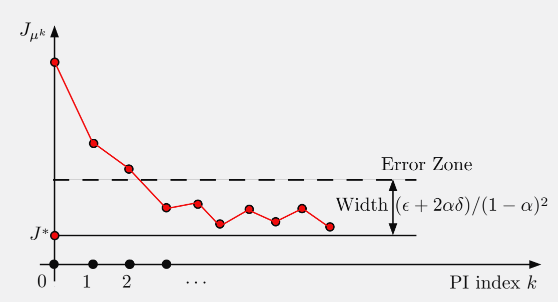
First Step "Future"

$$\min_{u\in U(x)} \sum_{y=1}^{n} p_{xy}(u)\big(g(x,u,y) + \alpha \tilde{J}_{\mu}(y)\big)$$

Approximation of $E\{\cdot\}$
Adaptive simulation
Monte Carlo tree search
Certainty equivalence

Approximation of J_μ
Rollout by (possibly inexact) simulation
Truncated rollout (optimistic PI)
Parallel rollout (multiple policies)
Problem approximation (aggregation)

- Multistep lookahead may be used
- Multiple policies variant uses $\tilde{J}(y) = \min\left\{J_{\mu^1}(x), \ldots, J_{\mu^m}(x)\right\}$
- Corresponding PI variants
- Approximate PI: Repeated approximate rollout; generates a sequence of policies $\{\mu^k\}$
- Approximate PI needs off-line training of policies and/or terminal cost function approximations



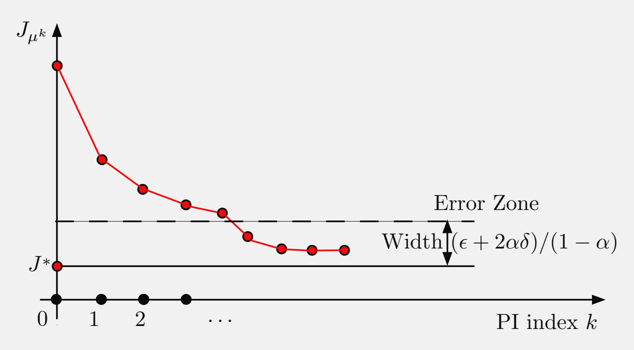
Assuming an approximate policy evaluation error satisfying

$$\max_{x=1,\dots,n} \left| \tilde{J}_{\mu^k}(x) - J_{\mu^k}(x) \right| \leq \delta$$

and an approximate policy improvement error satisfying

$$\max_{x=1,\dots,n} \left| \sum_{y=1}^{n} p_{xy}\big(\mu^{k+1}(x)\big)\big(g(x,\mu^{k+1}(x),y) + \alpha \tilde{J}_{\mu^k}(y)\big) - \min_{u\in U(x)} \sum_{y=1}^{n} p_{xy}(u)\big(g(x,u,y) + \alpha \tilde{J}_{\mu^k}(y)\big) \right| \leq \epsilon$$

Error Bound for the Case Where Policies Converge (NDP, 1996)



- A better error bound (by a factor $1-\alpha$) holds if the generated policy sequence $\{\mu^k\}$ converges to some policy.
- Convergence of policies is guaranteed in some cases; approximate PI using aggregation is one of them.

Truncated Rollout with Multistep Lookahead - Error Bound

Consider truncated rollout with

- ℓ-step lookahead
- Followed by rollout with a policy μ for m steps
- Followed by terminal cost function approximation $\tilde{J}$

For the rollout policy $\tilde{\mu}$, we have:

- The error bound

$$\|J_{\tilde{\mu}} - J^*\| \leq \frac{2\alpha^\ell}{1-\alpha}\left(\alpha^m \|\tilde{J} - J_\mu\| + \|J_\mu - J^*\|\right),$$

where $\|J\| = \max_{x=1,\ldots,n} \big|J(x)\big|$ is the max-norm.

- The cost improvement bound

$$J_{\tilde{\mu}}(x) \leq J_\mu(x) + \frac{2\alpha^{m-1}}{1-\alpha}\|\tilde{J} - J_\mu\|, \qquad x = 1, \ldots, n$$

Note that it helps to have:

ℓ and m: large, $\|\tilde{J} - J_\mu\|$ and $\|J_\mu - J^*\|$: small

About the Next Lecture

We will cover distributed and multiagent RL:

- Multiagent rollout and policy iteration
- State space partitioning and use of parallel computation
- Case studies