Reinforcement Learning and Optimal Control

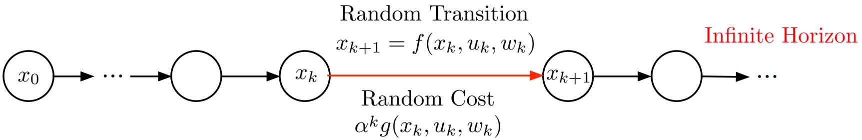
ASU, CSE 691, Winter 2019

Dimitri P. Bertsekas
dimitrib@mit.edu

Lecture 8

Outline

Stochastic DP 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.
- Policies $\pi = \{\mu_0, \mu_1, \ldots\}$ with $\mu_k(x) \in U(x)$ for all x and k.
- Special scalar α with $0 < \alpha \leq 1$. If $\alpha < 1$ the problem is called discounted.
- Cost of stage k: $\alpha^k g\big(x_k, \mu_k(x_k), w_k\big)$.
- Cost of a policy

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

- Optimal cost function $J^*(x_0) = \min_\pi J_\pi(x_0)$
- If $\alpha = 1$ we assume a special cost-free termination state t. The objective is to reach t at minimum expected cost. The problem is called stochastic shortest path (SSP) problem.

Convergence of VI

Given any initial conditions $J_0(1), \ldots, J_0(n)$, the sequence $\{J_k(i)\}$ generated by VI

$$J_{k+1}(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha J_k(j)\big), \qquad i = 1, \ldots, n,$$

converges to $J^*(i)$ for each i.

Bellman's equation

The optimal cost function $J^* = \big(J^*(1), \ldots, J^*(n)\big)$ satisfies the equation

$$J^*(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha J^*(j)\big), \qquad i = 1, \ldots, n,$$

and is the unique solution of this equation.

Optimality condition

A stationary policy μ is optimal if and only if for every state i, $\mu(i)$ attains the minimum in the Bellman equation.

Additional Results: Bellman Equation and Value Iteration for Policies

Fix a policy μ with cost function J_μ. Change the problem so the only control available at i is just $\mu(i)$ [not the set $U(i)$].

Apply our Bellman equation and VI convergence results:

- The VI algorithm (for policy μ),

$$J_{k+1}(i) = \sum_{j=1}^{n} p_{ij}\big(\mu(i)\big)\Big(g\big(i, \mu(i), j\big) + \alpha J_k(j)\Big), \qquad i = 1, \ldots, n,$$

  converges to the cost $J_\mu(i)$ for each i, for any initial conditions $J_0(1), \ldots, J_0(n)$.
- J_μ is the unique solution of the Bellman equation (of policy μ)

$$J_\mu(i) = \sum_{j=1}^{n} p_{ij}\big(\mu(i)\big)\Big(g\big(i, \mu(i), j\big) + \alpha J_\mu(j)\Big), \qquad i = 1, \ldots, n$$

- Solving this linear system of n equations with n unknowns, the costs $J_\mu(i)$, is called evaluation of policy μ.
- Evaluation of μ can be done by exact solution of the Bellman equation (e.g., Gaussian elimination), or iteratively with the VI algorithm (most likely for large n).
- Similar results hold for SSP problems.

Shorthand Notation

We introduce the DP operators

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

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

- They provide convenience of notation AND a vehicle for unification.
- T_μ and T form the "mathematical signature" of a DP problem, and serve to unify the DP theory (extensions to minimax, games, infinite spaces problems, etc).
- Their critical property is monotonicity (as J increases so does $T_\mu J$ and TJ); see the "Abstract DP" book (DPB, 2018).

All the DP results/algorithms can be written in math shorthand using T and T_μ

- VI algorithm: $J_{k+1} = TJ_k$, $J_{k+1} = T_\mu J_k$, $k = 0, 1, \dots$
- Bellman equation: $J^* = TJ^*$, $J_\mu = T_\mu J_\mu$.
- μ is optimal if and only if $TJ^* = T_\mu J^*$.

Contraction Property of T and T_μ

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

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

In our discounted and SSP problems, T and T_μ are contractions

- Introduce a (weighted max) norm for the vectors $J = \big(J(1),\ldots,J(n)\big)$:

$$\|J\| = \max_{i=1,\ldots,n} \frac{\big|J(i)\big|}{v(i)},$$

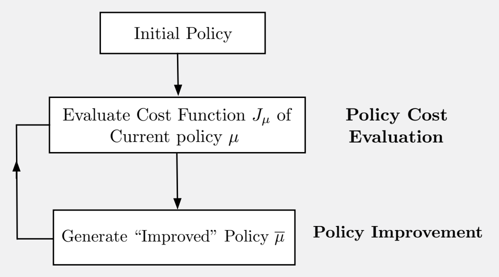
  where $v(1),\ldots,v(n)$ are some positive scalars.

- Definition: A mapping H that maps $J = \big(J(1),\ldots,J(n)\big)$ to the vector $HJ = \big((HJ)(1),\ldots,(HJ)(n)\big)$ is a contraction if for some ρ with $0 < \rho < 1$

$$\|HJ - HJ'\| \le \rho\|J - J'\|, \qquad \text{for all } J, J'$$

- For our discounted and SSP problems, under our assumptions, T and T_μ are contractions (in addition to being monotone).
- For the discounted problem, $\rho = \alpha$ and $v(i) \equiv 1$.
- This is the mathematical reason why our problems are so nice!

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 $J_{\mu^k}(i)$, $i = 1, \ldots, n$, as the solution of the (linear) Bellman equation system

$$J_{\mu^k}(i) = \sum_{j=1}^{n} p_{ij}\big(\mu^k(i)\big)\Big(g\big(i, \mu^k(i), j\big) + \alpha J_{\mu^k}(j)\Big), \quad i = 1, \ldots, n$$

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

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

- Compactly (in shorthand): PI is written as $T_{\mu^{k+1}} J_{\mu^k} = T J_{\mu^k}$.

Proof of Policy Improvement Property

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

We will show that $J_{\overline{\mu}} \leq J_\mu$, where $\overline{\mu}$ is obtained from μ by PI

- Denote by J_N the cost function of a policy that applies $\overline{\mu}$ for the first N stages and applies μ thereafter.
- We have the Bellman equation $J_\mu(i) = \sum_{j=1}^n p_{ij}\big(\mu(i)\big)\Big(g\big(i,\mu(i),j\big) + \alpha J_\mu(j)\Big)$, so

$$J_1(i) = \sum_{j=1}^n p_{ij}\big(\overline{\mu}(i)\big)\Big(g\big(i,\overline{\mu}(i),j\big) + \alpha J_\mu(j)\Big) \leq J_\mu(i) \qquad \text{(by policy improvement eq.)}$$

- From the definition of J_2 and J_1, monotonicity, and the preceding relation, we have

$$J_2(i) = \sum_{j=1}^n p_{ij}\big(\overline{\mu}(i)\big)\Big(g\big(i,\overline{\mu}(i),j\big)+\alpha J_1(j)\Big) \leq \sum_{j=1}^n p_{ij}\big(\overline{\mu}(i)\big)\Big(g\big(i,\overline{\mu}(i),j\big)+\alpha J_\mu(j)\Big) = J_1(i)$$

  so $J_2(i) \leq J_1(i) \leq J_\mu(i)$ for all i.
- Continuing similarly, we obtain $J_{N+1}(i) \leq J_N(i) \leq J_\mu(i)$ for all i and N. Since $J_N \rightarrow J_{\overline{\mu}}$ (VI for $\overline{\mu}$ converges), it follows that $J_{\overline{\mu}} \leq J_\mu$.

Optimistic PI: Like Standard PI, but Policy Evaluation is Approximate, and Based on a Finite Number of VI

Generates sequences of cost function approximations $\{J_k\}$ and policies $\{\mu^k\}$

Given the typical function J_k:

- Policy improvement computes a policy μ^k such that

$$\mu^k(i) \in \arg\min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha J_k(j)\big), \quad i = 1, \ldots, n$$

- Optimistic policy evaluation starts with $\hat{J}_{k,0} = J_k$, and uses m_k VI iterations for policy μ^k to compute $\hat{J}_{k,1}, \ldots, \hat{J}_{k,m_k}$ according to

$$\hat{J}_{k,m+1}(i) = \sum_{j=1}^{n} p_{ij}(\mu^k(i))\Big(g\big(i,\mu^k(i),j\big) + \alpha \hat{J}_{k,m}(j)\Big)$$

for all $i = 1, \ldots, n$, $m = 0, \ldots, m_k - 1$, and sets $J_{k+1} = \hat{J}_{k,m_k}$.

Convergence (using a cost improvement argument similar to standard PI)

For the optimistic PI algorithm, we have $J_k \to J^*$ and $J_{\mu^k} \to J^*$.

Multistep Policy Iteration: Policy Improvement with Multistep Lookahead

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.

Given the typical policy μ^k:

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

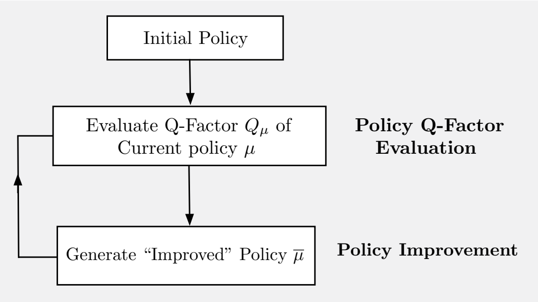
$$J_{\mu^k}(i) = \sum_{j=1}^{n} p_{ij}\big(\mu^k(i)\big)\Big(g\big(i, \mu^k(i), j\big) + \alpha J_{\mu^k}(j)\Big), \quad i = 1, \ldots, n$$

- Policy improvement with ℓ-step lookahead then 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 $\hat{\mu}_0$.

Convergence (using similar argument to standard PI)

Exact multistep PI has the same solid convergence properties as its one-step lookahead counterpart.

Policy Iteration for Q-Factors (Can be Used in Model-Free/Monte Carlo Contexts)



Given the typical policy μ^k:

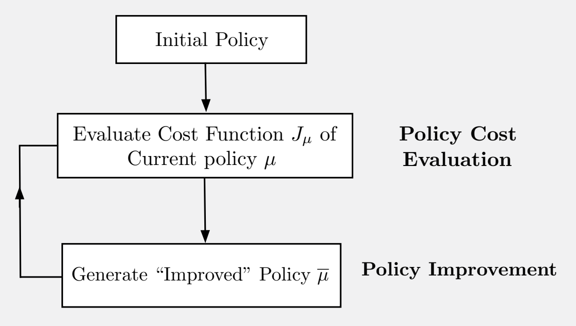
- Policy evaluation computes $Q_{\mu^k}(i,u)$, for all $i = 1, \dots, n$, and $u \in U(i)$, as the solution of the (linear) system of equations

$$Q_{\mu^k}(i,u) = \sum_{j=1}^{n} p_{ij}(u)\Big(g(i,u,j) + \alpha Q_{\mu^k}\big(j, \mu^k(j)\big)\Big)$$

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

$$\mu^{k+1}(i) \in \arg\min_{u \in U(i)} Q_{\mu^k}(i,u), \qquad i = 1, \dots, n$$

A Working Break: Think About Approximate PI



How would you introduce approximations into PI?

What would make sense for:

- Approximation in policy evaluation?
- Approximation in policy improvement?

Give examples (problem approximation, rollout, MPC, neural nets ...)

Approximation in Value Space for Infinite Horizon Problems

Approximate minimization

First Step "Future"

$$\min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \tilde{J}(j)\big)$$

Approximations:

Replace $E\{\cdot\}$ with nominal values (certainty equivalence)

Adaptive simulation

Monte Carlo tree search

Computation of $\tilde{J}$:

Problem approximation

Rollout

Approximate PI

Parametric approximation

Aggregation

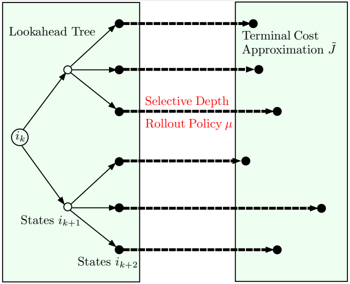
We will focus on rollout, and particularly on approximate PI schemes, which operate as follows:

- Several policies $\mu^0, \mu^1, \ldots, \mu^m$ are generated, starting with an initial policy μ^0.
- Each policy μ^k is evaluated approximately, with a cost function $\tilde{J}_{\mu^k}$, often with the use of a parametric approximation/neural network approach.
- The next policy μ^{k+1} is generated by policy improvement based on $\tilde{J}_{\mu^k}$.
- The approximate evaluation $\tilde{J}_{\mu^m}$ of the last policy in the sequence is used as the lookahead approximation $\tilde{J}$ in a one-step or multistep lookahead minimization.

Rollout

The pure form of rollout : Approximation in value space with $\tilde{J} = J_\mu$

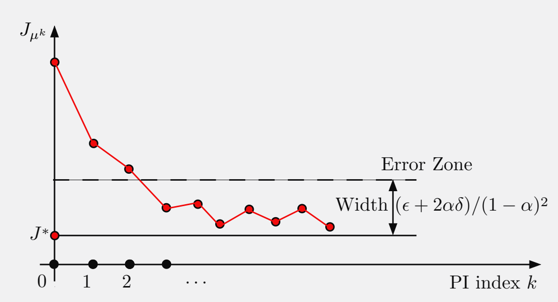
- μ is called the base policy, and is usually evaluated by Monte-Carlo.
- The rollout policy is the result of a single policy improvement using μ.
- So the rollout policy improves over the base policy.



Variants of rollout (ℓ-step lookahead, truncated rollout, terminal cost approx)

- ℓ-step lookahead, then rollout with policy μ for a limited number of steps, and finally a terminal cost approximation.
- This is a single optimistic policy iteration combined with multistep lookahead.

Approximate (Nonoptimistic) Policy Iteration - Error Bound - Stability



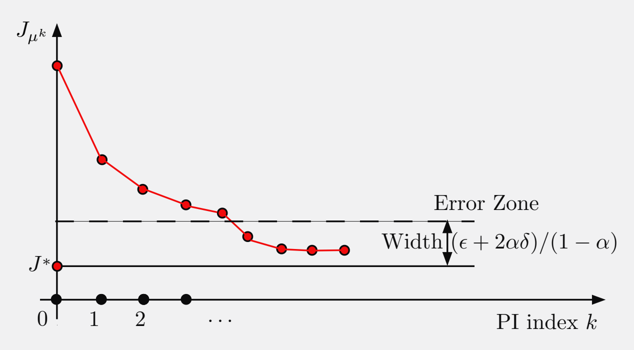
Assume an approximate policy evaluation error satisfying

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

and an approximate policy improvement error satisfying

$$\max_{i=1,\ldots,n} \Bigg| \sum_{j=1}^{n} p_{ij}\big(\mu^{k+1}(i)\big)\big(g(i,\mu^{k+1}(i),j) + \alpha \tilde{J}_{\mu^k}(j)\big) - \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha \tilde{J}_{\mu^k}(j)\big) \Bigg| \leq \epsilon$$

Error Bound for the Case Where Policies Converge



- 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.

About the Next Lecture

We will cover:

- PI with parametric approximation methods
- Linear programming approach
- Q-learning
- Additional methods; temporal differences

PLEASE READ AS MUCH OF SECTIONS 4.7-4.10 AS YOU CAN

PLEASE DOWNLOAD THE LATEST VERSIONS FROM MY WEBSITE