CE 273
Markov Decision Processes

Lecture 9

Policy Iteration

Previously on Markov Decision Processes

Practically, a cost of c units in one future time-step is equivalent to incurring αc now, where $0 \leq \alpha < 1$. More generally, cost c at time step n is equivalent to $\alpha^n c$ now.

One interpretation of α is that it reflects the interest rate. Another grim way to look at is to assume that time is finite, and the future may not happen with probability $(1 - \alpha)$. Define the cost over the infinite horizon as

$$C = \sum_{n=0}^{\infty} \alpha^n c(X_n)$$

C is a random variable and hence let's look at the expected total discounted cost starting from state i,

$$\phi(i) = \mathbb{E}\left[C | X_0 = i\right]$$

Theorem

Suppose c and ϕ represent column vectors of $c(i)$s and $\phi(i)$s. For $0 \leq \alpha < 1$,

$$\phi = (I - \alpha P)^{-1} c$$

Previously on Markov Decision Processes

We will mostly deal with countable state, control, and disturbance spaces. In such cases, we can write the DP equations and the T operators in more compact form.

Suppose the state space is $X = \{1, \ldots, n\}$. The transitions no longer are a function of k and hence we can write

$$p_{ij}(u) = \mathbb{P}[x_{k+1} = j | x_k = i, u_k = u] \, \forall \, i, j \in X, u \in U(i)$$

The two T mappings take the form

$$(TJ)(i) = \min_{u \in U(i)} \left\{ g(i, u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J(j) \right\} \forall \, i \in X$$

$$(T_\mu J)(i) = \left\{ g(i, \mu(i)) + \alpha \sum_{j=1}^{n} p_{ij}(\mu(i)) J(j) \right\} \forall \, i \in X$$

Note that it has been implicitly assumed that g does not depend on the disturbance. How can we relax that?

Previously on Markov Decision Processes

One can also write vector forms of these equations.

$$J = \begin{pmatrix} J(1) \\ \vdots \\ J(n) \end{pmatrix} \qquad TJ = \begin{pmatrix} (TJ)(1) \\ \vdots \\ (TJ)(n) \end{pmatrix} \qquad T_\mu J = \begin{pmatrix} (T_\mu J)(1) \\ \vdots \\ (T_\mu J)(n) \end{pmatrix}$$

For a given policy μ, we can also write the one-step transition probability matrix as

$$P_\mu = \begin{pmatrix} p_{11}(\mu(1)) & \dots & p_{1n}(\mu(1)) \\ \vdots & \ddots & \vdots \\ p_{n1}(\mu(n)) & \dots & p_{nn}(\mu(n)) \end{pmatrix}$$

and the cost vector for a fixed policy μ as

$$g_\mu = \begin{pmatrix} g(1, \mu(1)) \\ \vdots \\ g(n, \mu(n)) \end{pmatrix}$$

Thus, the T-mu operator in matrix form can be written as

$$T_\mu J = g_\mu + \alpha P_\mu J$$

Previously on Markov Decision Processes

Proposition

1. *For any bounded function* $J : X \to \mathbb{R}$, $J^* = \lim_{k\to\infty} T^k J$
2. *(Bellman Equations) The optimal value functions satisfy* $J^* = TJ^*$ *and* J^* *is the unique solution of this equation.*
3. *For any bounded function* $J : X \to \mathbb{R}$, $J_\mu = \lim_{k\to\infty} T_\mu^k J$
4. *The value functions associated with a stationary policy* μ *satisfy* $J_\mu = T_\mu J_\mu$ *and* J_μ *is the unique solution of this equation.*
5. *(Necessary and Sufficient Conditions for Optimality) A stationary policy* μ *is optimal* $\Leftrightarrow$ *it attains the minimum in the Bellman equations, i.e.,*
$$TJ^* = T_\mu J^*$$

Previously on Markov Decision Processes

Lemma (Monotonicity Lemma)

For any $J : X \to \mathbb{R}$ *and* $J' : X \to \mathbb{R}$ *such that* $J \leq J'$ *and a stationary policy* μ,

1. $T^k J \leq T^k J'$
2. $T_\mu^k J \leq T_\mu^k J'$

Lemma (Constant Shift Lemma)

For every k, *and* $J : X \to \mathbb{R}$ *and stationary policy* μ

1. $\big(T^k(J + re)\big)(i) = \big(T^k J)\big)(i) + \alpha^k r$
2. $\big(T_\mu^k(J + re)\big)(i) = \big(T_\mu^k J)\big)(i) + \alpha^k r$

Previously on Markov Decision Processes

VALUE ITERATION

Fix a tolerance level $\epsilon > 0$
Select $J_0 \in B(X)$ and $k \leftarrow 0$
$J_1 \leftarrow TJ_0$
while $\|J_{k+1} - J_k\| > \frac{\epsilon(1-\alpha)}{2\alpha}$ **do**
    $k \leftarrow k + 1$
    $J_{k+1} \leftarrow TJ_k$
end while

Select μ_ϵ that satisfies $T_{\mu_\epsilon} J_{k+1} = TJ_{k+1}$

In other words, the policy constructed at termination can be written as

$$\mu_\epsilon(i) \in \arg\min_{u \in U(i)} \mathbb{E}\left\{ g(i, u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J_{k+1}(j) \right\}$$

Lecture Outline

1. Policy Iteration
2. Modified Policy Iteration

Lecture Outline

Policy Iteration

The value iteration method is one way of finding the optimal values and policies. It operates in the 'value space' by moving from one value function to another using the T operator.

Another standard technique to solve MDPs is called Policy iteration, which operates in the 'policy space'. That is, we move from one policy to another.

A key difference between both the algorithms is that:

- Value iteration converges in the limit. In the example we saw earlier, the policy may remain unchanged over multiple iterations.
- Policy iteration converges after a finite number of iterations since the total number of policies are finite (for problems with finite states and actions).

Proposition (Policy Improvement Property (PIP))

Let μ and μ' be stationary policies such that $T_{\mu'} J_\mu = T J_\mu$. Then,

$$J_{\mu'}(i) \leq J_\mu(i) \, \forall \, i = 1, \ldots, n$$

Furthermore, if μ is not optimal, strict inequality holds for at least one i.

Proof.

Recall that $J_\mu = T_\mu J_\mu$ and by assumption $T_{\mu'} J_\mu = T J_\mu$. Thus for every i,

$$\begin{aligned} J_\mu(i) &= g(i, \mu(i)) + \alpha \sum_{j=1}^{n} p_{ij}(\mu(i)) J_\mu(i) \\ &\geq g(i, \mu'(i)) + \alpha \sum_{j=1}^{n} p_{ij}(\mu'(i)) J_\mu(i) \\ &= (T_{\mu'} J_\mu)(i) \end{aligned}$$

Proof.

Apply $T_{\mu'}$ on both sides and use the monotonicity lemma.

$$J_\mu \geq T_{\mu'} J_\mu \geq T^2_{\mu'} J_\mu \geq \ldots \geq lim_{k \to \infty} T^k_{\mu'} J_\mu = J_{\mu'}$$

We'll prove the second part using contraposition, i.e., if $J_\mu = J_{\mu'}$, then we need to show μ is optimal.

Since $J_\mu = J_{\mu'}$, $J_\mu = T_{\mu'} J_\mu$. Also by hypothesis, $T_{\mu'} J_\mu = TJ_\mu$. Hence, $J_\mu = TJ_\mu$. Therefore, J_μ satisfies the Bellman equations and $J_\mu = J^*$. ∎

Can you design a new algorithm using the PIP result?

Policy Iteration

Algorithm

The main steps of policy iteration are:

- For a given policy μ, construct the cost function J_μ (How?)
- Update the policy to μ' by finding the controls which minimize TJ_μ.

When do we stop?

Policy Iteration

Pick an initial policy μ_0 (say a Greedy policy)
Set μ_1 such that $T_{\mu_1} J_{\mu_0} = T J_{\mu_0}$ and $k \leftarrow 0$
while $\mu_{k+1} \neq \mu_k$ **do**
    $k \leftarrow k + 1$
    Compute J_{μ_k} by solving $J_{\mu_k} = T_{\mu_k} J_{\mu_k}$, i.e., ▷ **Policy Evaluation**

$$J_{\mu_k} = (I - \alpha P_{\mu_k})^{-1} g_{\mu_k}$$

    Compute a new policy μ_{k+1} that satisfies ▷ **Policy Improvement**

$$T_{\mu_{k+1}} J_{\mu_k} = T J_{\mu_k}$$

end while
$\mu^* \leftarrow \mu_k$ and $J^* \leftarrow J_{\mu_k}$

Since the termination criteria in the above algorithm compares policies between consecutive iterations, breaking ties arbitrarily can slow convergence.

Hence, we set $\mu_{k+1}(i) = \mu_k(i)$ whenever possible or stop when $J_{\mu_k} = T J_{\mu_k}$

In the policy evaluation stage, we are solving $J_{\mu_k} = T_{\mu_k} J_{\mu_k}$, or equivalently

$$J_{\mu_k} = (I - \alpha P_{\mu_k})^{-1} g_{\mu_k}$$

We discussed earlier that this system has solutions because T_μ is a contraction mapping.

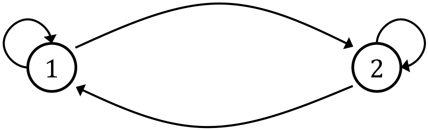
An alternate argument involving the eigenvalues of P_{μ_k} can also be used to show that the system admits a solution.

Recall that a square matrix is invertible iff it does not have a zero eigenvalue. Can an eigenvalue of $I - \alpha P_{\mu_k}$ be zero?

Policy Iteration

Example

Perform three iterations of the PI algorithm for the following example with two states 1 and 2. Assume that the discount factor is 0.9.



- $U(1) = \{u_1, u_2\}$
- $g(1, u_1) = 2, g(1, u_2) = 0.5$
- $p_{1j}(u_1) = [3/4 \;\; 1/4]$
- $p_{1j}(u_2) = [1/4 \;\; 3/4]$

- $U(2) = \{u_1, u_2\}$
- $g(2, u_1) = 1, g(2, u_2) = 3$
- $p_{2j}(u_1) = [3/4 \;\; 1/4]$
- $p_{2j}(u_2) = [1/4 \;\; 3/4]$

Policy Iteration

Example

Iteration 0: Suppose we start with an initial policy μ_0, where $\mu_0(1) = u_1$, and $\mu_0(2) = u_2$.

Policy Evaluation:

$$P_{\mu_0} = \begin{bmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{bmatrix} \qquad g_{\mu_0} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Thus, using $J_{\mu_0} = (I - \alpha P_{\mu_0})^{-1} g_{\mu_0}$,

$$J_{\mu_0} = \begin{bmatrix} J_{\mu_0}(1) \\ J_{\mu_0}(2) \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - 0.9 \begin{bmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{bmatrix} \right)^{-1} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 24.09 \\ 25.91 \end{bmatrix}$$

Policy Iteration

Example

Policy Improvement:

$$(TJ_{\mu_0})(1) = \min\Big\{2 + 0.9(0.75 * 24.09 + 0.25 * 25.91),$$
$$0.5 + 0.9(0.25 * 24.09 + 0.75 * 25.91)\Big\} = \min\Big\{24.09, \mathbf{23.41}\Big\}$$

Hence, set $\mu_1(1) = u_2$

$$(TJ_{\mu_0})(2) = \min\Big\{1 + 0.9(0.75 * 24.09 + 0.25 * 25.91),$$
$$3 + 0.9(0.25 * 24.09 + 0.75 * 25.91)\Big\} = \min\Big\{\mathbf{23.09}, 25.91\Big\}$$

Hence, set $\mu_1(2) = u_1$

Policy Iteration
Example

Iteration 1: $\mu_1(1) = u_2$, and $\mu_1(2) = u_1$.

Policy Evaluation:

$$P_{\mu_1} = \begin{bmatrix} 0.25 & 0.75 \\ 0.75 & 0.25 \end{bmatrix} \qquad g_{\mu_1} = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

Thus, using $J_{\mu_1} = (I - \alpha P_{\mu_1})^{-1} g_{\mu_1}$,

$$J_{\mu_1} = \begin{bmatrix} J_{\mu_1}(1) \\ J_{\mu_1}(2) \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - 0.9 \begin{bmatrix} 0.25 & 0.75 \\ 0.75 & 0.25 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} = \begin{bmatrix} 7.33 \\ 7.67 \end{bmatrix}$$

Policy Improvement:

$$(TJ_{\mu_1})(1) = \min \Big\{ 2 + 0.9(0.75 * 7.33 + 0.25 * 7.67),$$
$$0.5 + 0.9(0.25 * 7.33 + 0.75 * 7.67) \Big\} = \min \Big\{ 8.67, \mathbf{7.33} \Big\}$$

Hence, set $\mu_2(1) = u_2$

$$(TJ_{\mu_1})(2) = \min \Big\{ 1 + 0.9(0.75 * 7.33 + 0.25 * 7.67),$$
$$3 + 0.9(0.25 * 7.33 + 0.75 * 7.67) \Big\} = \min \Big\{ \mathbf{7.67}, 9.83 \Big\}$$

Hence, set $\mu_2(2) = u_1$. Since $\mu_1 = \mu_2$, we have found the optimal policy and value function.

Policy Iteration

Geometric Interpretation

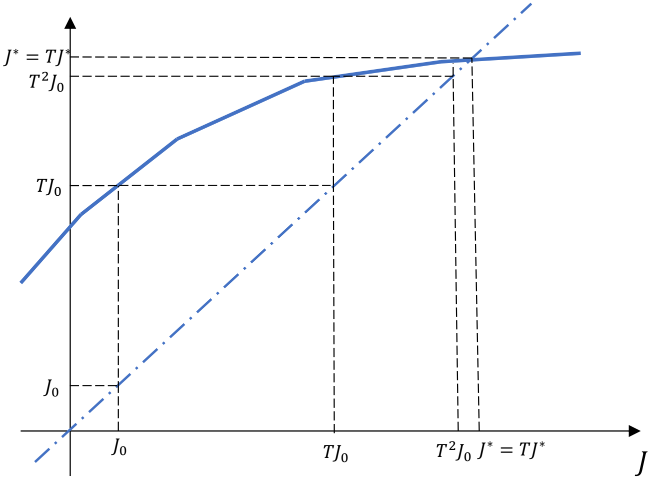


Figure: Value Iteration

For a given stationary policy μ, $T_\mu J = g_\mu + \alpha P_\mu J$ is linear in J

TJ is the piecewise linear function $\min_\mu\{g_\mu + \alpha P_\mu J\}$

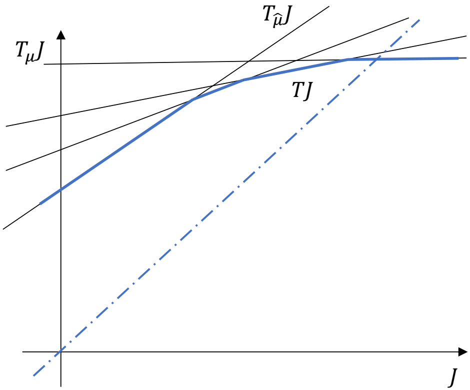


Figure: TJ is 'Piecewise Concave'

Policy Iteration

Geometric Interpretation

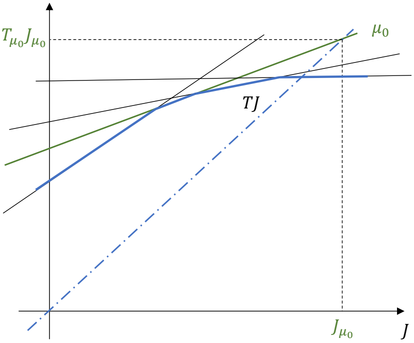


Figure: Policy Evaluation

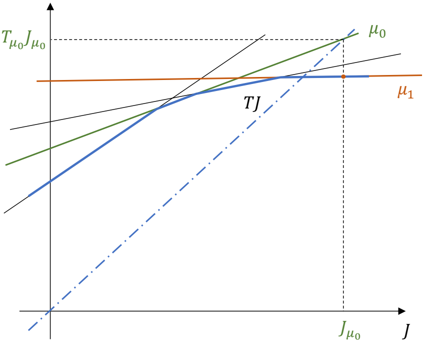


Figure: Policy Improvement

Policy Iteration

Geometric Interpretation

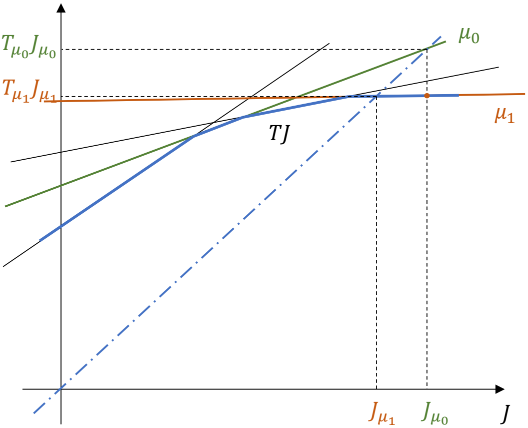


Figure: Policy Iteration

Lecture Outline

Modified Policy Iteration

Empirically, convergence of PI is much better than that of VI. In fact, it exhibits a significant improvement within the first few iterations. (One can still construct pathological instances which converge slowly.)

Under some assumptions on the one-step costs and probabilities, there are theoretical results which show that the convergence is superlinear.

Given a policy μ_k, the policy evaluation step involves finding J_{μ_k} by solving the system of equations

$$(I - \alpha P_{\mu_k}) J_{\mu_k} = g_{\mu_k}$$

This step can be computationally expensive if we have a large number of states. Is there an alternate method to estimate J_{μ_k}?

Modified Policy Iteration

Recall that J_μ is not only a solution to $J_\mu = T_\mu J_\mu$ but also equals $\lim_{k\to\infty} T_\mu^k J$.

Thus, one can use a VI-like method to calculate J_{μ_k} by repeatedly applying T_{μ_k} on some initial guess J. The exact values of J_{μ_k} are of course obtained only in the limit and when we stop after a finite number of iterations we get an approximate J_{μ_k}.

The modified policy iteration method uses a finite number of VI-type steps instead of calculating the inverse of a large matrix. The policy improvement step is then carried out and the process is repeated.

Empirically MPI does better than regular PI. This method is also called **Optimistic Policy Iteration**.

Modified Policy Iteration

Pseudocode

Let $m_0, m_1, \ldots$ be a sequence of positive integers.

MODIFIED POLICY ITERATION

$k \leftarrow 0$

Pick an initial policy μ_0 and estimate $J_{\mu_0} \approx J_0$

Set μ_1 such that $T_{\mu_1} J_0 = T J_0$ and estimate $J_{\mu_1} \approx J_1$

while $\|J_{k+1} - J_k\| > \frac{\epsilon(1-\alpha)}{2\alpha}$ **do**

  $k \leftarrow k + 1$

  Compute a new policy μ_{k+1} that satisfies ▷ **Policy Improvement**

$$T_{\mu_{k+1}} J_k = T J_k$$

  Compute an approximate $J_{\mu_{k+1}}$ by solving ▷ **Policy Evaluation**

$$J_{\mu_{k+1}} \approx J_{k+1} = T_{\mu_{k+1}}^{m_{k+1}} J_k$$

end while

Modified Policy Iteration

Geometric Interpretation

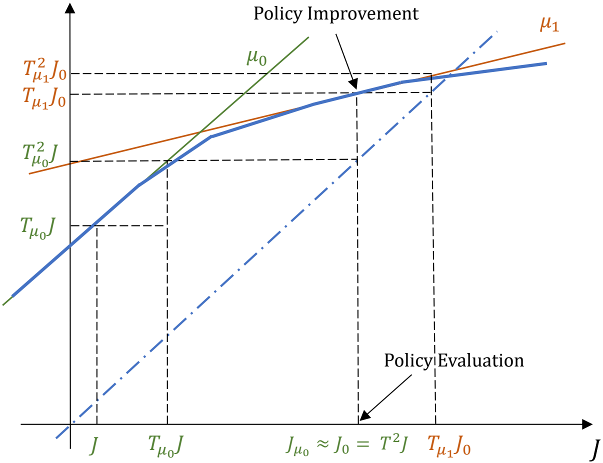


Figure: Modified Policy Iteration

Modified Policy Iteration

Convergence

Since VI-like steps are involved, we used the distance between successive value function iterates to define convergence.

Therefore the sequence of value functions can be shown to converge in the limit. It is also possible to show that μ_k is optimal for all k greater than some index K.

What happens if

- $m_k = 1 \, \forall \, k$? MPI is same as VI
- $m_k = \infty \, \forall \, k$? MPI is same as PI

Your Moment of Zen

Proof by Construction

Question: Show that $\left(\boldsymbol{A}^{-1}\right)^{-1} = \boldsymbol{A}$

Answer:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$A^{-1} = \begin{bmatrix} \text{c} & \text{q} \\ \text{ɐ} & \text{p} \end{bmatrix}$$

$$(A^{-1})^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$