CE 273
Markov Decision Processes

Lecture 16

Policy Iteration and Linear Programming for Average Cost MDPs

Previously on Markov Decision Processes

Proposition

For any transition matrix P and fundamental matrix H

$$P^* = PP^* = P^*P = P^*P^*$$
$$P^*H = HP^* = 0$$
$$P^* + H = I + PH$$

Previously on Markov Decision Processes

Definition

The gain J_μ of a policy μ is defined as

$$J_\mu = P_\mu^* g_\mu$$

Definition

The bias h_μ of a policy μ is defined as

$$h_\mu = H_\mu g_\mu$$

where $H_\mu = (I - P_\mu + P_\mu^*)^{-1} - P_\mu^*$ and is called the fundamental matrix.

In addition, suppose the associated Markov chain is aperiodic, i.e., if $P_\mu^* = \lim_{N\to\infty} P_\mu^N$ (Case III), then we can interpret h_μ as

$$h_\mu = \lim_{N\to\infty} \sum_{k=0}^{N} P_\mu^k (g_\mu - J_\mu)$$

a relative cost vector, i.e., the difference of the total cost of μ and the total cost if one-stage costs were set to J_μ.

Previously on Markov Decision Processes

Unlike discounted and total cost MDPs, where we could solve a system of equations for a given policy (and use this in the policy iteration algorithm), we cannot simply solve

$$J = P_\mu J$$
$$J + h = g_\mu + P_\mu h$$

to get the average cost of policy μ. (Why?)

If (J_μ, h_μ) solves the above system, then $(J_\mu, h_\mu + \text{constant})$ also satisfies the above system. Hence, there are an infinite number of solutions. We will call these policy evaluation equations for easy referencing.

In general, it can be shown that all solutions to the above system are of the form $(J_\mu, h_\mu + d)$, where $d = P_\mu d$.

Previously on Markov Decision Processes

The earlier proposition and discussion established that a Blackwell optimal policy is optimal to the average cost problem.

Further, optimal policies were found to satisfy some equations which are the necessary conditions for optimality. It can also be shown that they are sufficient.

Proposition

If J' and h' satisfy the following pair of optimality equations

$$J(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u) J(j) \; \forall\, i = 1, \dots, n$$

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

where $\bar{U}(i)$ is the set of controls that attain the minimum in the above equation. Then, $J' = J^*$ is the optimal average cost vector.

Further, if a stationary policy μ attains the minimum in the above equations, then it is the optimal policy μ^*.

Previously on Markov Decision Processes

In summary, if the average cost is independent of the initial state, the following proposition is true

Proposition

If a scalar λ and a vector h satisfy

$$\lambda + h(i) = \min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) h(j) \right\} \;\; \forall \, i = 1, \ldots n$$

then λ is the optimal average cost $J^*(i)$ for all i, i.e.,

$$\lambda = \min_{\mu} J_\mu(i) = J^*(i) \;\forall\; i = 1, \ldots, n$$

Further, if μ^* attains the minimum in the first expression, then $J_{\mu^*}(i) = \lambda \,\forall\, i$.

In shorthand, the first equation can be rewritten as $\lambda e + h = Th$. Think of this as being analogous to $J^* = TJ^*$ in the discounted world.

Previously on Markov Decision Processes

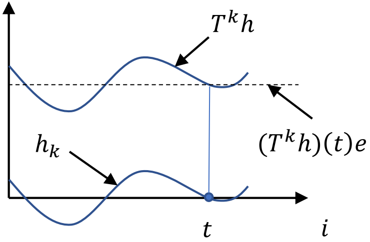
Theorem

$$J^* = \lim_{k\to\infty} \frac{1}{k} T^k h$$

For unichain MDPs,

$$h_k = T^k h - (T^k h)(t)e$$

where t is some arbitrary state. Effectively, we are shifting the entire function by a constant. But note that the constant varies across iterations.



The iterates h_k remain bounded and the bounds do not depend on k.

Previously on Markov Decision Processes

The relative value iteration (RVI) works for unichain MDPs which induce aperiodic Markov Chains.

Define the span semi-norm of a vector h as

$$sp(h) = \max_{i \in X} h(i) - \min_{i \in X} h(i)$$

Relative Value Iteration

Fix a tolerance level $\epsilon > 0$ and select a state t
Select $h_0 \in B(X)$ and $k \leftarrow 0$
$h_1 \leftarrow Th_0 - (Th_0)(t)e$
while $sp(h_{k+1} - h_k) > \epsilon$ **do**
    $k \leftarrow k + 1$
    $h_{k+1} \leftarrow Th_k - (Th_k)(t)e$
end while

Select μ such that

$$\mu(i) \in \arg\min_{u \in U(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) h_k(j) \right\}$$

Lecture Outline

1. Policy Iteration
2. Linear Programming

Lecture Outline

Policy Iteration

Policy Iteration

Introduction

A policy iteration algorithm that alternates between policy evaluation and policy improvement can be used to solve the average cost problem.

Policy iteration works for both unichain and multichain MDPs but the steps for the latter type of problem are more involved.

Consider a unichain MDP. The MDP associated with any policy thus has a single closed communicating class and a set of transient states.

Suppose at the kth iteration, we have a policy μ_k. Then, the policy evaluation is done by solving the system

$$\lambda_k e + h_k = T_{\mu_k} h_k$$

Note that for unichain MDPs $J_k = P_{\mu_k} J_k$ is always satisfied. However, as noted earlier, the above system does not have a unique solution.

Hence, we select an arbitrary state t and set $h_k(t) = 0$. It can be shown that this new system

$$\lambda_k e + h_k = T_{\mu_k} h_k$$
$$h_k(t) = 0$$

has a unique solution and λ_k corresponds to the gain of the policy μ_k.

Note that a h_k that satisfies the above set of equations need not equal h_{μ_k}, which is why we use the subscript k and not μ_k. (We can however call λ_k as λ_{μ_k}.)

Since we know the policy, we could as well compute the gain $J_{\mu_k} = \lambda_k e$ and h_{μ_k} and use it in the next step but it would require more computation.

Policy improvement is done by finding the controls which optimize Th_k, that is,

$$T_{\mu_{k+1}} h_k = Th_k$$

The new policy μ_{k+1} is the same for any h_k that satisfies the policy evaluation equation. (Why?)

As before,

- The algorithm is terminated when $\mu_{k+1} = \mu_k$.
- Ties are broken such that $\mu_{k+1}(i) = \mu_k(i)$ whenever possible.

Policy Iteration

$k \leftarrow 0$

Pick an initial policy μ_0 (say a Greedy policy) and some state t

do

Compute λ_k and h_k by solving i.e., ▷ **Policy Evaluation**

$$\lambda_k e + h_k = T_{\mu_k} h_k$$

$$h_k(t) = 0$$

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

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

$k \leftarrow k + 1$

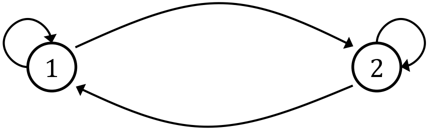
while $\mu_{k+1} \neq \mu_k$

$\mu^* \leftarrow \mu_k$ and $J^* \leftarrow \lambda_k e$

Set $\mu_{k+1}(i) = \mu_k(i)$ whenever possible.

Policy Iteration
Example

Perform two iterations of the PI algorithm for the following example with two states 1 and 2. Assume that state 1 is the reference state t. Start with the policy $\mu_0(1) = u_1$ and $\mu_0(2) = u_2$.



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

Policy Iteration

Main Result

Let's now see why the policy iteration method works.

Proposition

Consider an unichain MDP and a policy μ with gain-bias pair (λ_μ, h_μ). Suppose μ' is obtained from $T_{\mu'} h_\mu = Th_\mu$ and denote using $(\lambda_{\mu'}, h_{\mu'})$ the gain-bias pair of μ'. If $\mu' \neq \mu$, the one of the following is true

1. $\lambda_{\mu'} < \lambda_\mu$
2. $\lambda_{\mu'} = \lambda_\mu$ *and* $h_{\mu'}(i) \leq h_\mu(i)$ *for all* $i = 1, \ldots, n$ *with equality occurring for states that are recurrent and strict inequality for at least one transient state.*

Proof.

We will only prove $\lambda_{\mu'} \leq \lambda_\mu$. To do so, it is enough to show

$$P^*_{\mu'}(T_\mu h_\mu - T_{\mu'} h_\mu) = (\lambda_\mu - \lambda_{\mu'})e$$

(Why?) By construction of μ', $T_{\mu'} h_\mu = Th_\mu \leq T_\mu h_\mu$. Therefore, $P^*_{\mu'}(T_\mu h_\mu - T_{\mu'} h_\mu) \geq 0$.

Proof.

$$P^*_{\mu'}(T_\mu h_\mu - T_{\mu'} h_\mu) = (\lambda_\mu - \lambda_{\mu'})e$$

Consider the LHS:

$$\begin{aligned}
P^*_{\mu'}(T_\mu h_\mu - T_{\mu'} h_\mu) &= P^*_{\mu'}\Big(T_\mu h_\mu - \Big(T_{\mu'} h_{\mu'} + (T_{\mu'} h_\mu - T_{\mu'} h_{\mu'})\Big)\Big) \\
&= P^*_{\mu'}\Big(\lambda_\mu e + h_\mu - \Big(\lambda_{\mu'} e + h_{\mu'} + (T_{\mu'} h_\mu - T_{\mu'} h_{\mu'})\Big)\Big) \\
&= P^*_{\mu'}\Big(\lambda_\mu e + h_\mu - \Big(\lambda_{\mu'} e + h_{\mu'} + P_{\mu'}(h_\mu - h_{\mu'})\Big)\Big) \\
&= P^*_{\mu'}\Big((\lambda_\mu - \lambda_{\mu'})e + (I - P_{\mu'})(h_\mu - h_{\mu'})\Big) \\
&= P^*_{\mu'}(\lambda_\mu - \lambda_{\mu'})e + (P^*_{\mu'} - P^*_{\mu'} P_{\mu'})(h_\mu - h_{\mu'}) \\
&= (\lambda_\mu - \lambda_{\mu'})e + 0
\end{aligned}$$

∎

For multichain MDPs, the same ideas work but both policy evaluation and improvement involve more equations.

Recall that the gain and bias of a policy satisfy

$$J = P_\mu J$$
$$J + h = g_\mu + P_\mu h$$

One cannot solve this system and identify the bias since it is not unique.

Non-uniqueness was an issue even in unichain MDPs, but the choice of the bias did not matter when we perform policy improvement.

However, that is no longer true for multichain MDPs. At every iteration k, we need (J_{μ_k}, h_{μ_k}) to find an improved policy μ_{k+1}!

An obvious option to find (J_{μ_k}, h_{μ_k}) is to estimate $P^*_{\mu_k}$ and the fundamental matrix H_{μ_k}. But this is computationally expensive. Alternately, the following result can be used

Proposition

Consider a stationary policy μ with the gain-bias pair (J_μ, h_μ). The set of solutions (J, h, v) to the following equations

$$J = P_\mu J$$
$$J + h = g_\mu + P_\mu h$$
$$h + v = P_\mu v$$

are of the form $(J_\mu, h_\mu, -H^2_\mu g_\mu + d)$ where d satisfies $d = P_\mu d$.

Once we have (J_{μ_k}, h_{μ_k}), μ_{k+1} is obtained from the following two-stage policy improvement procedure:

Step 1:
Choose a policy μ_{k+1} which satisfies

$$P_{\mu_{k+1}} J_{\mu_k} = \min_{\mu} P_{\mu} J_{\mu_k}$$

In other words,

$$\mu_{k+1}(i) \in \arg\min_{u \in U(i)} \left\{ \sum_{j=1}^{n} p_{ij}(u) J_{\mu_k}(j) \right\}$$

while setting $\mu_{k+1}(i) = \mu_k(i)$ whenever possible. If $\mu_{k+1} \neq \mu_k$, then we can switch to the policy evaluation procedure. Else, go to Step 2.

Step 2:

Choose a policy μ_{k+1} which satisfies

$$P_{\mu_{k+1}} J_{\mu_k} = \min_{\mu} P_{\mu} J_{\mu_k}$$

$$T_{\mu_{k+1}} h_{\mu_k} = \min_{\mu \in \bar{\Pi}} T_{\mu} h_{\mu_k}$$

where $\bar{\Pi}$ is the set of policies which attain the minimum in $\min_{\mu} P_{\mu} J_{\mu_k}$. Alternately we can write,

$$\mu_{k+1}(i) \in \arg \min_{u \in U(i)} \left\{ \sum_{j=1}^{n} p_{ij}(u) J_{\mu_k}(j) \right\}$$

$$\mu_{k+1}(i) \in \arg \min_{u \in \bar{U}(i)} \left\{ g(i,u) + \sum_{j=1}^{n} p_{ij}(u) h_{\mu_k}(j) \right\}$$

where $\bar{U}(i)$ is the set of controls are the optima of $\sum_{j=1}^{n} p_{ij}(u) J_{\mu_k}(j)$. Again set $\mu_{k+1}(i) = \mu_k(i)$ whenever possible.

Policy Iteration

Main Result

The above policy iteration procedure for multichain MDPs works because of the following proposition (which is is very similar to what we saw in unichain MDPs).

Proposition

Let μ_k be a policy with gain-bias pair (J_{μ_k}, h_{μ_k}). Suppose that μ_{k+1} is obtained from policy improvement step and let $(J_{\mu_{k+1}}, h_{\mu_{k+1}})$ be its gain-bias pair. If $\mu_{k+1} \neq \mu_k$ then one of the following is true

1. $J_{\mu_{k+1}}(i) \leq J_{\mu_k}(i)$ *for all* $i = 1, \ldots, n$ *with a strict inequality for at least one state* i.
2. $J_{\mu_{k+1}} = J_{\mu_k}$ *and* $h_{\mu_{k+1}} \leq h_{\mu_k}$ *for all* $i = 1, \ldots, n$ *with strict inequality for at least one state* i *transient under* μ_{k+1}.

Linear Programming

Both unichain and multichian MDPs can also be solved using linear programming.

We will need the following proposition to set up the LPs

Proposition

Let J and h be vectors which satisfy

$$J \leq P_\mu J$$
$$J + h \leq T_\mu h$$

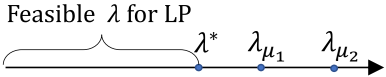
Then, $J \leq J_\mu$. Further, if equality holds in the first two inequalities, $J = J_\mu$.

Consider the case of unichain MDPs. The optimality conditions can be written as

$$\lambda e + h = Th = \min_{\mu} T_{\mu} h$$

Alternately, we can write $\lambda e + h \leq T_{\mu} h$ for all stationary policies μ. And by setting $J = \lambda e$, the previous proposition can be used.

Thus, for all functions h and scalars λ that satisfy $\lambda e + h \leq T_{\mu} h$, $\lambda \leq \lambda_{\mu}$ for every stationary policy μ.



Further for the optimal policy μ^*, the optimal average cost λ^* satisfies $\lambda^* e + h = T_{\mu^*} h$. Thus, λ^* is the largest λ that satisfies $\lambda e + h \leq Th$.

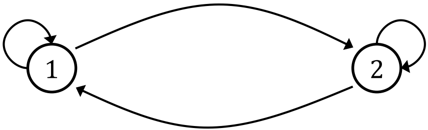
The primal LP for unichain MDPs can thus be written as

$$\begin{aligned} &\max \lambda \\ \text{s.t. } &\lambda + h(i) \le g(i,u) + \sum_{j=1}^{n} p_{ij}(u) h(j) \qquad \forall\, i = 1, \ldots, n, u \in U(i) \end{aligned}$$

which in standard form looks like

$$\begin{aligned} &\max \lambda \\ \text{s.t. } &\lambda + h(i) - \sum_{j=1}^{n} p_{ij}(u) h(j) \le g(i,u) \qquad \forall\, i = 1, \ldots, n, u \in U(i) \end{aligned}$$

Write the primal LP for the following example with two states 1 and 2.



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

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

Linear Programming

Dual Problem

Write the dual of the above LP.

- The number of dual variables equal to the number of constraints in the primal.
- Since the primal constraints are of the $\leq$ form, the dual variables must be ≥ 0.
- Since the primal variables are unconstrained, the dual will have equality constraints. (How many?)

Define variables $z(i, u)$ where $i \in X, u \in U(i)$.

$$
\begin{aligned}
\min \sum_{i=1}^{n} \sum_{u \in U(i)} g(i,u) z(i,u) & \\
\text{s.t.} \quad \sum_{u \in U(i)} z(i,u) - \sum_{j=1}^{n} \sum_{u \in U(j)} p_{ji}(u) z(j,u) = 0 & \qquad \forall\, i = 1, \ldots, n \\
\sum_{i=1}^{n} \sum_{u \in U(i)} z(i,u) = 1 & \\
z(i,u) \geq 0 & \qquad \forall\, u \in U(i), i = 1, \ldots, n
\end{aligned}
$$

If you think of $\sum_{u \in U(i)} z(i, u)$ as a new variable z_i, what do the constraints represent?

The primal problem gives us only the optimal λ. The dual solution on the other hand can be used to construct the optimal policy as well. (How did we do this for the discounted problem?)

Let $z^*(i,u)$ be the optimal dual solution. Define for all i,

$$U^*(i) = \{u \in U(i) | z^*(i,u) > 0\}$$

For average cost MDPs, it is not necessary that the above set is a singleton. Define a new set

$$C^* = \left\{ i \middle| \sum_{u \in U(i)} z^*(i,u) > 0 \right\}$$

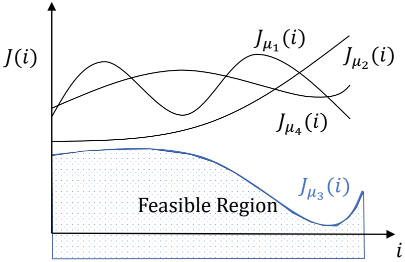
Note that the sets $U^*(i)$ and C^* are non-empty. (Why?) The following policy can be shown to be optimal

$$\mu^*(i) = \begin{cases} \text{any } u \in U^*(i) \text{ if } i \in C^* \\ \text{any } u \in U(i) \text{ if } i \notin C^* \end{cases}$$

It can also be shown that that the set C^* is a closed communicating class under the optimal policy μ^*.

The earlier proposition can also be used to construct a LP for the multichain MDP. Recall that we now have two sets of feasible constraints $J \leq P_\mu J$ and $J + h \leq T_\mu h$.

This implies that any pair of vectors (J, h) that satisfies the above system, $J \leq J_\mu$.



Thus, J^* is the "largest" vector satisfying the two constraints.

The primal problem for the multichain MDP can be written as

$$
\begin{aligned}
& \max \sum_{i=1}^{n} a_i y(i) \\
\text{s.t.} \quad & y(i) \leq \sum_{j=1}^{n} p_{ij}(u) y(j) && \forall\, i = 1, \ldots, n, u \in U(i) \\
& y(i) + h(i) \leq g(i,u) + \sum_{j=1}^{n} p_{ij}(u) h(j) && \forall\, i = 1, \ldots, n, u \in U(i)
\end{aligned}
$$

where a is a row vector of strictly positive reals satisfying $\sum_i a_i = 1$.

Hence, the dual takes the form

$$\min \sum_{i=1}^{n} \sum_{u \in U(i)} g(i,u) z(i,u)$$

$$\text{s.t.} \quad \sum_{u \in U(i)} z(i,u) - \sum_{j=1}^{n} \sum_{u \in U(j)} p_{ji}(u) z(j,u) = 0 \qquad \forall\, i = 1,\ldots,n$$

$$\sum_{u \in U(i)} \big(z(i,u) + r(i,u) \big) - \sum_{j=1}^{n} \sum_{u \in U(j)} r(j,u) p_{ji}(u) = a_i \qquad \forall\, i = 1,\ldots,n$$

$$z(i,u) \geq 0 \qquad \forall\, u \in U(i), i = 1,\ldots,n$$

$$r(i,u) \geq 0 \qquad \forall\, u \in U(i), i = 1,\ldots,n$$

As before, the optimal policy is constructed by first supposing

$$C^* = \left\{ i \middle| \sum_{u \in U(i)} z^*(i,u) > 0 \right\}$$

using which, we define

$$\mu^*(i) = \begin{cases} \text{any } u \text{ such that } z^*(i,u) > 0 \text{ if } i \in C^* \\ \text{any } u \text{ such that } r^*(i,u) > 0 \text{ if } i \notin C^* \end{cases}$$

Your Moment of Zen