CE 273
Markov Decision Processes

Lecture 19

Approximation in Value Space - Part II

Previously on Markov Decision Processes

Consider a policy μ. Suppose we simulate S trajectories and each trajectory is indexed by s. A trajectory s can be written as

$$i_0, \mu(i_0), i_1, \mu(i_1), \ldots, i_t, \mu(i_t), \ldots, i_{t(s)}$$

where $i_{t(s)}$ represents the terminal state of trajectory s. Given this trajectory, we can compute the sample future cost at every time step $t = 0, \ldots, t(s)$ as follows

$$G_t(s) = g(i_t, \mu(i_t), i_{t+1}) + \alpha g(i_{t+1}, \mu(i_{t+1}), i_{t+2}) + \ldots + \alpha^{t(s)-1-t} g(i_{t(s)-1}, \mu(i_{t(s)-1}), i_{t(s)})$$

Suppose we want to find the approximate value of some state i. Then, we look at the first occurrence of i in each trajectory and calculate the discounted cost from that point onward.

This process is repeated for all trajectories and the costs are averaged. This method is also called **First-Visit Monte Carlo Method**.

Previously on Markov Decision Processes

Thus, one can incrementally update the approximate value functions for each sample trajectory s.

For every i_t in the trajectory $i_0, \mu(i_0), i_1, \mu(i_1), \ldots, i_{t(s)}$, update

$$\text{numVisits}(i_t) \leftarrow \text{numVisits}(i_t) + 1$$

$$\tilde{J}_\mu(i_t) \leftarrow \tilde{J}_\mu(i_t) + \frac{1}{\text{numVisits}(i_t)}\Big(G_t(s) - \tilde{J}_\mu(i_t)\Big)$$

This method can be generalized as

$$\tilde{J}_\mu(i_t) \leftarrow \tilde{J}_\mu(i_t) + \gamma\Big(G_t(s) - \tilde{J}_\mu(i_t)\Big)$$

Comparing this with gradient descent methods, γ can be interpreted as a step size and $G_t(s)$ can be thought of a target. One can let γ shrink to zero over time.

This is ideal for scenarios in which the system dynamics are not time-invariant.

Previously on Markov Decision Processes

Mathematically, the MC update

$$\tilde{J}_\mu(i_t) \leftarrow \tilde{J}_\mu(i_t) + \gamma\Big(G_t(s) - \tilde{J}_\mu(i_t)\Big)$$

is transformed to

$$\tilde{J}_\mu(i_t) \leftarrow \tilde{J}_\mu(i_t) + \gamma\Big(g(i_t, \mu(i_t), i_{t+1}) + \alpha\tilde{J}_\mu(i_{t+1}) - \tilde{J}_\mu(i_t)\Big)$$

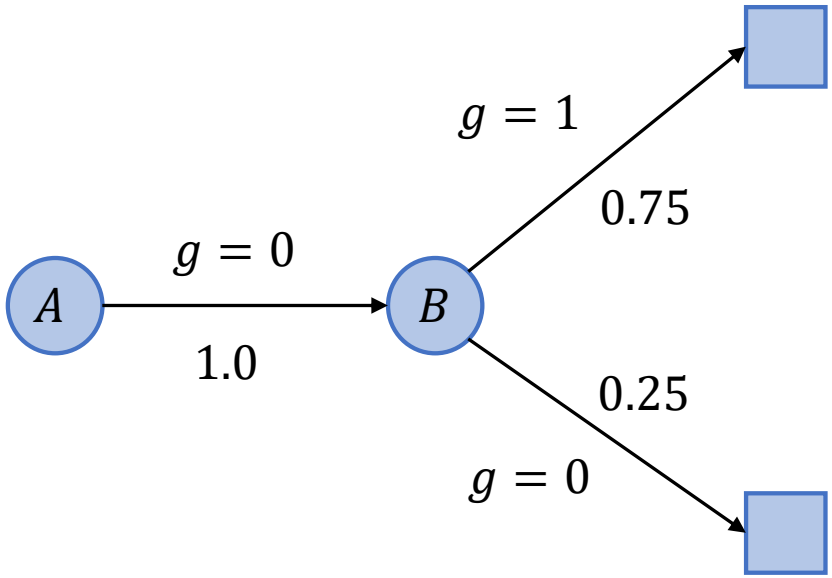
This method is also called the TD(0) algorithm, and

- $g(i_t, \mu(i_t), i_{t+1}) + \alpha\tilde{J}_\mu(i_{t+1})$ is called the TD target
- $g(i_t, \mu(i_t), i_{t+1}) + \alpha\tilde{J}_\mu(i_{t+1}) - \tilde{J}_\mu(i_t)$ is called the TD error

Previously on Markov Decision Processes

Thus, MC methods attempt to fit the value functions to the sample means and minimize the mean squared error.

TD methods on the other hand create a Markov chain by discovering transitions and costs along the lines of maximum-likelihood estimation and use it estimate approximate value functions.



Previously on Markov Decision Processes

Suppose for each state i, we extract m features. Let k represent a generic feature. Then, the vector of approximate value functions can be written as

$$\tilde{J} = \Phi r$$

where Φ is

$$\Phi = \begin{bmatrix} \phi_1(1) & \dots & \phi_m(1) \\ \phi_1(2) & \dots & \phi_m(2) \\ \vdots & \vdots & \vdots \\ \phi_1(n) & \dots & \phi_m(n) \end{bmatrix}_{n \times m} \qquad r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}_{m \times 1}$$

The rows of the Φ matrix are features and the columns can be interpreted as basis functions/vectors.

Thus, we can think of the subspace $S = \{\Phi r | r \in \mathbb{R}^m\}$ as the subspace spanned by the basis vectors (columns of Φ).

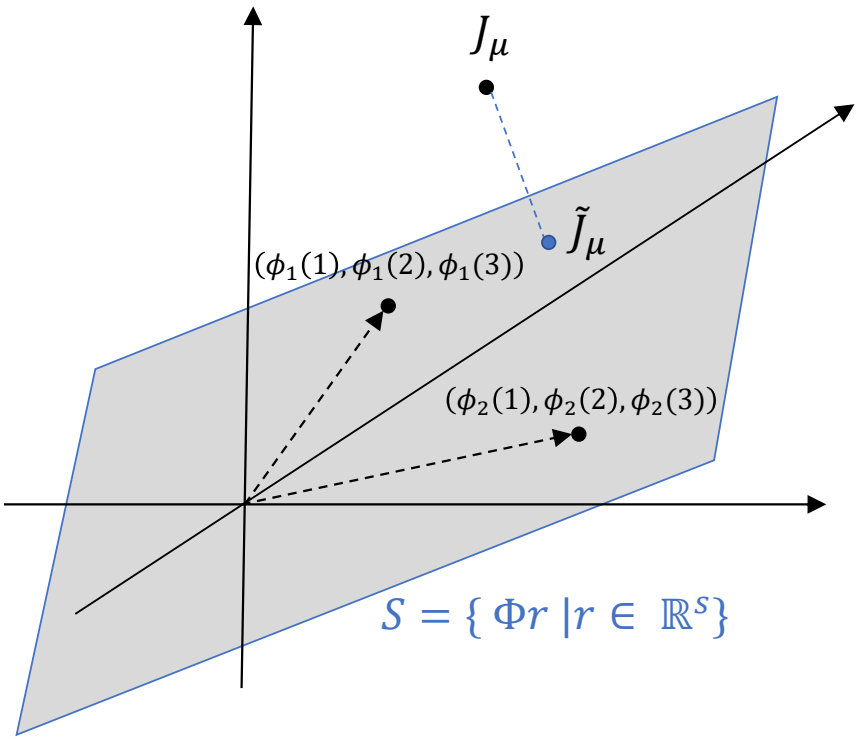
Previously on Markov Decision Processes

Suppose we have access to the true value function J_μ. In the direct method, the optimal parameters can be obtained by solving

$$\min_{r \in \mathbb{R}^m} \|J_\mu - \Phi r\|$$

Suppose J_μ and r have dimensions 3×1 and 2×1 respectively. Is r uniquely determined?

$$\tilde{J}_\mu = \begin{bmatrix} \tilde{J}_\mu(1) \\ \tilde{J}_\mu(2) \\ \tilde{J}_\mu(3) \end{bmatrix} = \begin{bmatrix} \phi_1(1) & \phi_2(1) \\ \phi_1(2) & \phi_2(2) \\ \phi_1(3) & \phi_2(3) \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

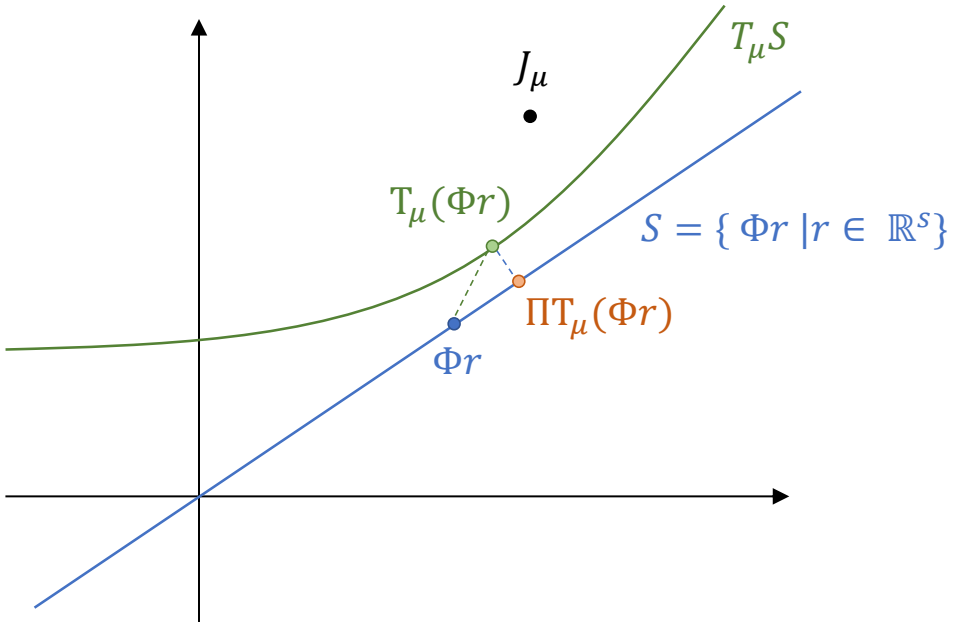


Previously on Markov Decision Processes

In the indirect approximation approach, we replace the Bellman equations $J_\mu = T_\mu J_\mu$ with

$$\Phi r = \Pi T_\mu(\Phi r)$$

where Π denotes the projection of a point on the subspace S.



This approach has connections with the TD(0) method which we will discuss in detail in the next class.

Lecture Outline

1. Parametric Methods
2. Policy Improvement

Parametric Methods

Recall that in the direct approach, we try to find the best r by minimizing the distance between J_μ and Φr. Mathematically,

$$r^* = \arg\min_{r \in \mathbb{R}^m} \|J_\mu - \Phi r\|_\xi^2$$

where $\|\cdot\|$ is a weighted Euclidean norm defined as

$$\|J\|_\xi^2 = \sum_{i=1}^{n} \xi_i (J(i))^2$$

The point Φr^* is still the projection but with respect to the weighted Euclidean norm.

What are the optimality conditions of the above minimization problem?

Parametric Methods

Direct Approach

$$r^* = \arg\min_{r \in \mathbb{R}^m} \sum_{i=1}^{n} \xi_i \Big(\phi(i)' r - J_\mu(i) \Big)^2$$

The gradient of the objective is

$$2 \sum_{i=1}^{n} \xi_i \phi(i) \Big(\phi(i)' r - J_\mu(i) \Big)$$

Setting it to zero at r^*,

$$r^* = \left(\sum_{i=1}^{n} \xi_i \phi(i) \phi(i)' \right)^{-1} \sum_{i=1}^{n} \xi_i \phi(i) J_\mu(i)$$

If the columns of Φ are linearly independent, then r^* can be uniquely identified. However, there are two issues with the above procedure:

- J_μ is not known.
- The size of the state space n may be extremely large and hence calculating sums could be intractable.

To address these issues, a Monte Carlo simulation method like the one discussed in the previous class can be used.

Suppose $\xi = (\xi_1, \xi_2, \ldots, \xi_n)$ is a probability distribution. Then the sum $\sum_{i=1}^{n} \xi_i a_i$ can be interpreted as the expectation of a random variable whose support is $a_1, \ldots, a_n$ with a pmf ξ.

A simulation-based approach to compute the above expectation is to sample from $a_1, \ldots, a_n$ according to the distribution ξ and form Monte Carlo averages.

Suppose $k = 1, \ldots, K$ are a set of states (not episodes or state transitions) sampled according to the distribution ξ, then

$$\sum_{i=1}^{n} \xi_i a_i \approx \frac{1}{K} \sum_{k=1}^{K} a_k$$

Using a similar logic, instead of finding entire sums in the following expression,

$$r^* = \left(\sum_{i=1}^{n} \xi_i \phi(i) \phi(i)' \right)^{-1} \sum_{i=1}^{n} \xi_i \phi(i) J_\mu(i)$$

we sample a set of states $k = 1, \ldots, K$ using the distribution ξ and construct an estimate of the optimal solution $\hat{r}^*$ as follows

$$\begin{aligned} \hat{r}^* &= \left(\frac{1}{K} \sum_{i=1}^{K} \phi(i_k) \phi(i_k)' \right)^{-1} \frac{1}{K} \sum_{i=1}^{K} \phi(i_k) J_\mu(i_k) \\ &= \left(\sum_{i=1}^{K} \phi(i_k) \phi(i_k)' \right)^{-1} \sum_{i=1}^{K} \phi(i_k) J_\mu(i_k) \end{aligned}$$

Can you reverse engineer the objective that $\hat{r}^*$ would optimize?

Parametric Methods

Alternate Interpretation

Another way to look at the problem is that instead of optimizing the original objective

$$\sum_{i=1}^{n} \xi_i \Big(\phi(i)' r - J_\mu(i) \Big)^2$$

we optimize its sample approximation

$$\sum_{i=1}^{K} \frac{1}{K} \Big(\phi(i_k)' r - J_\mu(i_k) \Big)$$

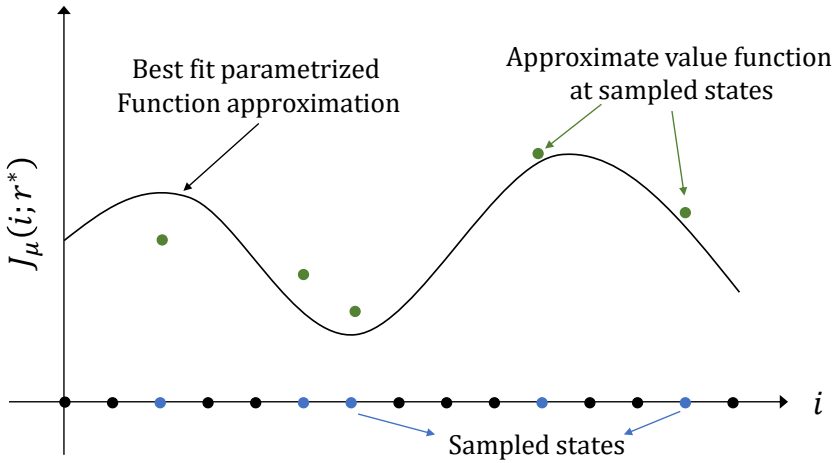
The above discussion addresses the issue of large state spaces and makes it easier to calculate sums. But how do we estimate $J_\mu(i_k)$? One could use MC or TD methods for this purpose.

Parametric Methods

Direct Approach Summary

If we use MC and TD methods to find the approximate value functions, why do we need parametric methods? In other words, what do we gain from the parametric methods?

MC and TD methods estimate the value at each state. In the parametric approach, the sampled states $i_1, \ldots, i_K$ could be a small subset of the entire state space and we fit a function based on $\tilde{J}_\mu$ at these states.



The r^* values help in fitting a curve (shape is determined by choice of the basis functions) through these sampled points and thus provides an approximation of the value function for all the remaining states.

ξ can be arbitrary chosen but it helps to select the limiting distribution of the DTMC associated with policy μ. (Why?) In practice, one can simulate trajectories/episodes and use the no. of visits to states as a proxy for ξ.

Parametric Methods

Indirect Approach

Given a policy μ, the standard way of finding J_μ is by solving the Bellman equation $J_\mu = T_\mu J_\mu$. This involves solving n equations with n unknowns.

The other alternative parametric method is the indirect approach in which instead of fitting functions, we find an alternate equation in the subspace $S = \{\Phi r | r \in \mathbb{R}^m\}$ which resembles Bellman equations.

Assumptions:

- For the theory to work, we will assume that the DTMC induced by the policy μ is irreducible and thus has a unique limiting distribution ξ. We will later see why this is a reasonable assumption to make when we study policy improvement.
- We will also suppose that the column vectors of Φ are linearly independent.

Parametric Methods

Indirect Approach

Let Π be the projection mapping on the subspace S. Recall that by definition of the projection ΠJ is the closest point in S according to the weighted norm $\|\cdot\|_\xi$. Mathematically,

$$\Pi J = \arg\min_{J' \in S} \|J - J'\|_\xi^2$$

Proposition

The mapping T_μ and the composite mapping Π and T_μ are contractions with respect to the weighted norm $\|\cdot\|_\xi$ with a modulus of contraction α

We find the best parameter vector by solving the **projected Bellman equations**,

$$\Phi r = \Pi T_\mu(\Phi r)$$

Using the fact that $T_\mu = g_\mu + \alpha P_\mu$, and that Φr^* solves $\Phi r = \Pi T_\mu(\Phi r)$, we can write

$$
\begin{aligned}
r^* &= \arg \min_{r \in \mathbb{R}^m} \|\Phi r - (g_\mu + \alpha P_\mu \Phi r^*)\|_\xi^2 \\
&= \arg \min_{r \in \mathbb{R}^m} \Big(\Phi r - (g_\mu + \alpha P_\mu \Phi r^*)\Big)' D \Big(\Phi r - (g_\mu + \alpha P_\mu \Phi r^*)\Big)
\end{aligned}
$$

where D is a $n \times n$ diagonal matrix $\text{diag}(\xi_1, \ldots, \xi_n)$. What are the optimality conditions of the above problem?

$$
\Phi' D \Big(\Phi r^* - (g_\mu + \alpha P_\mu \Phi r^*)\Big) = 0
$$

Hence, solving the above set of equations gives the optimal r values. How many equations and unknowns are in the above system? m and m which is $\lll n$.

We will refer to this as the matrix form of the projected equations.

$$\Phi' D\Big(\Phi r^* - (g_\mu + \alpha P_\mu \Phi r^*)\Big) = 0$$

Substituting A_μ for $\Phi' D(I - \alpha P_\mu)\Phi$ and b_μ for $\Phi' D g_\mu$, we can write the above system compactly as

$$A_\mu r^* = b_\mu$$

Thus, r^* can be written as $A_\mu^{-1} b_\mu$ just like how J_μ was $(I - \alpha P_\mu^{-1}) g_\mu$. What are the dimensions of A_μ and b_μ?

But calculating them requires us to work with D, P_μ and g_μ all of which are large matrices or vectors.

Parametric Methods

Matrix Form of Projected Bellman Equations

Matrix and vector multiplications are essentially sums and hence instead of computing these matrices, we can use Monte Carlo simulation as done earlier.

We will try to find $\hat{A}_\mu$ that approximates

$$A_\mu = \Phi' D(I - \alpha P_\mu)\Phi$$

and vector $\hat{b}_\mu$ that approximates

$$b_\mu = \Phi' D g_\mu$$

and then solve $\hat{r}^* = \hat{A}_\mu^{-1}\hat{b}_\mu$. There are three difficult matrix operations in constructing the above approximations

$$\Phi' D\Phi \qquad \Phi' D P_\mu \Phi \qquad \Phi' D g_\mu$$

Can you write these three matrices as expectation-like sums?

Parametric Methods

Matrix Form of Projected Bellman Equations

$$\Phi' D \Phi = \sum_{i=1}^{n} \xi_i \phi(i) \phi(i)'$$

$$\Phi' D P_\mu \Phi = \sum_{i=1}^{n} \sum_{j=1}^{n} \xi_i p_{ij}(\mu(i)) \phi(i) \phi(j)'$$

$$\Phi' D g_\mu = \sum_{i=1}^{n} \sum_{j=1}^{n} \xi_i p_{ij}(\mu(i)) \phi(i)' g(i, \mu(i), j)$$

The first equation is the expected values with respect to the pmf ξ and the second and third can be treated as an expectation using a joint probability mass function $\{\xi_i p_{ij}(\mu(i)) | i, j = 1, \ldots, n\}$.

Approximating the first is easy but how do we approximate the second and third expressions using sampling?

Run a long simulation of the DMTC induced by μ. Suppose the states are $i_1, \ldots i_K$. Then, the estimates of the earlier matrices can be expressed as

$$\Phi' D \Phi = \sum_{i=1}^{n} \xi_i \phi(i) \phi(i)' \approx \frac{1}{K} \sum_{t=0}^{K} \phi(i_t) \phi(i_t)'$$

$$\Phi' D P_\mu \Phi = \sum_{i=1}^{n} \sum_{j=1}^{n} \xi_i p_{ij}(\mu(i)) \phi(i) \phi(j)' \approx \frac{1}{K} \sum_{t=1}^{K} \phi(i_t) \phi(i_{t+1})'$$

$$\Phi' D g_\mu = \sum_{i=1}^{n} \sum_{j=1}^{n} \xi_i p_{ij}(\mu(i)) \phi(i)' g(i, \mu(i), j) \approx \frac{1}{K} \sum_{t=1}^{K} \phi(i_t) g(i_t, \mu(i_t), i_{t+1})$$

Substituting these approximations in the expressions for $A_\mu = \Phi' D \Phi - \alpha \Phi' D P_\mu \Phi$ and $b_\mu = \Phi' D g_\mu$, we get

$$\hat{A}_\mu = \frac{1}{K} \sum_{t=1}^{K} \phi(i_t)\phi(i_t)' - \alpha \frac{1}{K} \sum_{t=1}^{K} \phi(i_t)\phi(i_{t+1})'$$

$$= \frac{1}{K} \sum_{t=1}^{K} \phi(i_t)\Big(\phi(i_t) - \alpha\phi(i_{t+1})\Big)'$$

$$\hat{b}_\mu = \frac{1}{K} \sum_{t=1}^{K} \phi(i_t) g(i_t, \mu(i_t), i_{t+1})$$

We then solve $\hat{A}_\mu \hat{r}^* = \hat{b}_\mu$ exactly or using some iterative method to find an estimate of the optimal r^*.

This method is also called Least Squares Temporal Differences (LSTD). To see why, rewrite the above equation as $\hat{A}_\mu \hat{r}^* - \hat{b}_\mu = 0$ and expand it as

$$\hat{A}_\mu \hat{r}^* - \hat{b}_\mu = \frac{1}{K} \sum_{t=1}^{K} \phi(i_t) \Big(\phi(i_t) - \alpha \phi(i_{t+1}) \Big)' \hat{r}^* - \frac{1}{K} \sum_{t=1}^{K} \phi(i_t) g(i_t, \mu(i_t) i_{t+1})$$

$$= \frac{1}{K} \sum_{t=1}^{K} \phi(i_t) \Big(\phi(i_t) \hat{r}^* - \alpha \phi(i_{t+1}) \hat{r}^* - g(i_t, \mu(i_t), i_{t+1}) \Big)$$

The expressions $\phi(i_t) \hat{r}^* - \alpha \phi(i_{t+1}) \hat{r}^* - g(i_t, \mu(i_t), i_{t+1})$ resembles the TD error! The LSTD method is part of a larger class of Galerkin approximation methods that can be used to solve the projected Bellman equations.

Policy Improvement

Policy Improvement

Introduction

So far, we have discussed how to approximately evaluate the value function of a given policy.

Using this approximate $\hat{J}_\mu$, we perform one policy improvement step to get a new policy μ' and repeat this procedure. Note that almost all methods involve simulation of the following type and hence we run into the risk of not visiting/rarely visiting certain states.

$$\text{numVisits}(i_t) \leftarrow \text{numVisits}(i_t) + 1$$

$$\tilde{J}_\mu(i_t) \leftarrow \tilde{J}_\mu(i_t) + \frac{1}{\text{numVisits}(i_t)}\Big(G_t(s) - \tilde{J}_\mu(i_t)\Big)$$

Thus, the value functions at these states is likely to be very poor because of which in the policy improvement step, we may never/rarely take actions that lead us to these states.

This is a cause for concern since we may never be able search over all policies and may end up reaching very sub-optimal policy. One way to get out of this cyclic problem is to use randomized policies.

Policy Improvement

ϵ-Greedy Exploration

We alter the policy improvement step by

- Assigning a weight of (1-ϵ) on the optimal action at each state, i.e., on u^* belonging to

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

- Selecting one of the remaining controls randomly with probability ϵ.

This randomization procedure ensures that the state space is constantly explored and hence we do not run into the earlier issue.

One can also show that the construction of one ϵ-greedy policy from another ϵ-greedy policy preserves the policy improvement property (PIP)!

Policy Improvement

ϵ-Greedy Exploration

There are a still a few issues with this procedure:

- If ϵ is chosen to be very small, how is exploration guaranteed?
- The optimal policy is usually deterministic. So how do we get there using randomized policies?
- Finding the optimal control requires transition probabilities and involves calculations over the entire state space.

The first two can be addressed by gradually shrinking ϵ to zero (say $\epsilon_k = 1/k$, where k is the iteration number). This method is called Greedy in the Limit with Infinite Exploration (GLIE).

The second can be addressed using $\tilde{Q}_\mu(i, u)$ instead of $\tilde{J}_\mu(i)$. The simulation-based and parametric methods that we discussed so far are applicable to the Q functions as well.

Policy Improvement

ϵ-Greedy Exploration

We can get good estimates of value functions using MC, TD, and parametric methods and then improve policies using the above method. This works well when we have a simulator of the system and adequate time to learn the optimal policies.

However, if we are operating in an real-time setting, can we update the policies more frequently without accurately estimating the approximate value functions? Yes. Think of this as begin analogous to modified policy iteration.

Policy Improvement

MC and GLIE

Consider the MC method in which we simulate or observe episodes. For each episode, we can update the Q values using

$$\text{numVisits}(i_t, u_t) \leftarrow \text{numVisits}(i_t, u_t) + 1$$

$$\tilde{Q}(i_t, u_t) \leftarrow \tilde{Q}(i_t, u_t) + \frac{1}{\text{numVisits}(i_t, u_t)} \Big(G_t(s) - \tilde{Q}(i_t, u_t) \Big)$$

and then improve the policy using the GLIE approach. In other words, we do not use a single policy over multiple episodes, but change our policy after every episode.

For example, imagine you are playing Tetris using one policy and after each game you use a 'improved' policy.

Policy Improvement

TD(0) and GLIE = SARSA

We can use similar ideas to combine the TD(0) method with the GLIE policy improvement approach.

Since the TD(0) method updates the value functions after every state transition, the policy is also updated after every transition!

This method is also called SARSA since each transition can be described using the letters S, A, R, S, and A. In state S, take an action A and observe reward R and move to a new state S' where you use the GLIE policy to take a new action A'.

For example, if you use SARSA on a game of Tetris, the policy is updated as soon as each tetromino lands in the well.

Your Moment of Zen