RBE 595 — Reinforcement Learning
Week #5 Assignment

Arjan Gupta

Problem 1

When is it suited to apply Monte-Carlo to a problem?

Answer

Monte-Carlo methods are best suited to be applied to problems where we do not have a model of the environment (i.e., the dynamics of the environment are unknown). For example, sometimes it is simply not practical to model the complexity of the environment. In such cases, the agent must learn about the environment by interacting with it and using the obtained rewards to update its policy via the action-value function.

Problem 2

When does the Monte-carlo prediction performs the first update?

Answer

The Monte-Carlo prediction performs the first update after an episode terminates. This is because the Monte-Carlo method is an episodic method, i.e., it learns from a series of state, action, and reward tuples that occur in an episode. Here are the main steps of the algorithm,

1. Initialize $V(s)$ and $Returns(s)$ arbitrarily for all $s \in \mathcal{S}$
2. Loop for the number of episodes:
   (a) Generate an episode following the policy π
   (b) (**Here is where the update happens**) For each state S_t (visited only once) in the episode:
      i. Compute G
      ii. Append G to $Returns(S_t)$
      iii. $V(S_t) \leftarrow \text{average}(Returns(S_t))$

So the first update happens when the algorithm enters step 2(b) for the first time.

Problem 3

What is off-policy learning and why it is useful?

Answer

Off-policy learning is a method of reinforcement learning where the agent learns about the environment by observing the behavior of another agent, called the *behavior policy*, which is the policy responsible for exploration and interaction. However, the agent performs evaluation and optimization using a different policy, called the *target policy*.

Off-policy learning is useful for the following reasons,

- It avoids the unlikely assumption of exploring starts. In some Monte Carlo algorithms, the exploratory behavior comes from random starting states, however this is not always possible. Off-policy learning allows the agent to learn about the environment without this assumption.
- Existing knowledge can be leveraged by learning from the behavior of other agents. The behavior policy can be a simple random policy, or it can be a policy that has been learned from past experience.
- It avoids the situation where the agent is stuck in a suboptimal policy because it is not exploring enough. This would happen if the agent is using a greedy and deterministic policy to learn about the environment.
- It avoids unexpected actions that may occur during exploration. Instead, the behavior policy can continuously explore while the target policy learns. This is particularly useful in cases where the environment is dangerous or expensive to explore, or if humans are involved.

Problem 4

(Exercise 5.5, page 105) Consider an MDP with a single nonterminal state and a single action that transitions back to the nonterminal state with probability p and transitions to the terminal state with probability $1 - p$. Let the reward be +1 on all transitions, and let $\gamma = 1$. Suppose you observe one episode that lasts 10 steps, with a return of 10. What are the first-visit and every-visit estimators of the value of the nonterminal state?

Answer

Since this problem does not involve a behavior and target policy, we will not use the importance sampling ratio. Instead, we can manually calculate the first-visit and every-visit estimators of the value of the nonterminal state.

Given episode

Let the nonterminal state be s (and let s_i denote the i^{th} time s was visited) and the terminal state be s'. Let the action be a (and let a_i denote the i^{th} time s was visited). The reward is $r = 1$ for all transitions. Also, $\gamma = 1$.

The given episode is as follows,

$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_3 \xrightarrow{a_4} s_4 \xrightarrow{a_5} s_5 \xrightarrow{a_6} s_6 \xrightarrow{a_7} s_7 \xrightarrow{a_8} s_8 \xrightarrow{a_9} s_9 \xrightarrow{a_{10}} s'$$

As we can see by the subscript of a, there are 10 rewards of 1 each. Therefore, the total return is 10.

First-Visit Estimator
The first-visit estimator of the value of the nonterminal state is calculated as follows,

$$\begin{aligned} V(s) &= 1+1+1+1+1+1+1+1+1+1 \\ &= 1(10) = 10 \end{aligned}$$

Every-Visit Estimator
The every-visit estimator of the value of the nonterminal state is calculated as follows,

$$\begin{aligned} V(s) &= \frac{1}{10}(1+2+3+4+5+6+7+8+9+10) \\ &= \frac{1}{10}(55) = 5.5 \end{aligned}$$

Problem 5

(Exercise 5.7, page 108) In learning curves such as those shown in Figure 5.3 error generally decreases with training, as indeed happened for the ordinary importance-sampling method. But for the weighted importance-sampling method error first increased and then decreased. Why do you think this happened?

Answer

As shown in the lectures by Dr. Navid Dadkhah Tehrani, here is a table showing the bias and variance comparison between the ordinary importance-sampling method and the weighted importance-sampling method.

	Ordinary Importance-Sampling	**Weighted Importance-Sampling**
Bias	Un-biased	Biased (eventually unbiased)
Variance	Large	Low

For weighted importance-sampling, the bias is initially high, but it eventually becomes unbiased. That initial bias is what causes the error to increase initially. However, after a large number of episodes, the bias decreases and the error decreases as well.

Problem 6

(Exercise 5.8, page 108) The results with Example 5.5 and shown in Figure 5.4 used a first-visit MC method. Suppose that instead an every-visit MC method was used on the same problem. Would the variance of the estimator still be infinite? Why or why not?

Answer

No, the variance of the estimator would not be infinite. In fact, after the first episode that ends with the left action, the value of the state $V(s)$ would be 1 and would remain so for all subsequent episodes.

Problem 7

(Exercise 5.10, page 109) Derive the weighted-average update rule (5.8) from (5.7). Follow the pattern of the derivation of the unweighted rule (2.3).

Answer

Objective

We want to obtain,

$$V_{n+1} = V_n + \frac{W_n}{C_n}[G_n - V_n], \text{ for } n \geq 1 \tag{5.8}$$

Given that,

$$C_{n+1} = C_n + W_{n+1}$$

From,

$$V_n = \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \text{ for } n \geq 2 \tag{5.7}$$

Derivation

$$\begin{aligned}
V_n &= \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k} \\
V_{n+1} &= \frac{\sum_{k=1}^{n} W_k G_k}{\sum_{k=1}^{n} W_k} \\
&= \frac{\sum_{k=1}^{n-1} W_k G_k + W_n G_n}{\sum_{k=1}^{n} W_k} \\
&= \frac{\sum_{k=1}^{n-1} W_k G_k + W_n G_n}{\sum_{k=1}^{n} W_k} \cdot \frac{\sum_{k=1}^{n-1} W_k}{\sum_{k=1}^{n-1} W_k} \\
&= \frac{\sum_{k=1}^{n-1} W_k G_k + W_n G_n}{\sum_{k=1}^{n-1} W_k} \cdot \frac{\sum_{k=1}^{n-1} W_k}{\sum_{k=1}^{n} W_k} \\
&= \frac{\sum_{k=1}^{n-1} W_k G_k + W_n G_n}{\sum_{k=1}^{n-1} W_k} \cdot \frac{C_{n-1}}{C_n} \\
&= \left[\frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k} + \frac{W_n G_n}{\sum_{k=1}^{n-1} W_k}\right] \cdot \frac{C_{n-1}}{C_n} \\
&= \left[V_n + \frac{W_n G_n}{C_{n-1}}\right] \cdot \frac{C_{n-1}}{C_n} \\
&= V_n \cdot \frac{C_{n-1}}{C_n} + \frac{W_n G_n}{C_{n-1}} \cdot \frac{C_{n-1}}{C_n} \\
&= V_n \cdot \frac{C_{n-1}}{C_n} + \frac{W_n G_n}{C_n} \\
&= V_n \cdot \frac{C_{n-1}}{C_n} + \frac{W_n G_n}{C_n} + V_n - V_n \\
&= V_n + \frac{V_n C_{n-1}}{C_n} - V_n + \frac{W_n G_n}{C_n}
\end{aligned}$$

Problem 7 continued on next page. . .

Continuing,

$$\begin{aligned} V_{n+1} &= V_n + \frac{V_n C_{n-1} - V_n C_n}{C_n} + \frac{W_n G_n}{C_n} \\ &= V_n + \frac{V_n (C_{n-1} - C_n)}{C_n} + \frac{W_n G_n}{C_n} \end{aligned}$$

Where we know,

$$\begin{aligned} C_n &= C_{n-1} + W_n \\ W_n &= C_n - C_{n-1} \\ -W_n &= C_{n-1} - C_n \end{aligned}$$

Therefore,

$$\begin{aligned} V_{n+1} &= V_n + \frac{-V_n W_n}{C_n} + \frac{W_n G_n}{C_n} \\ &= V_n + \frac{W_n G_n}{C_n} - \frac{V_n W_n}{C_n} \\ V_{n+1} &= V_n + \frac{W_n}{C_n}(G_n - V_n) \end{aligned}$$

Problem 8

(Exercise 5.11, page 111) In the boxed algorithm for off-policy MC control, you may have been expecting the W update to have involved the importance-sampling ratio $\frac{\pi(A_t|S_t)}{b(A_t|S_t)}$, but instead it involves $\frac{1}{b(A_t|S_t)}$. Why is this nevertheless correct?

Answer

In the algorithm, the assumption is that $\pi(A_t|S_t) = 1$ because the target policy is deterministic. In other words, in state S_t, the target policy π will always choose action A_t. Therefore, the importance-sampling ratio becomes $\frac{1}{b(A_t|S_t)}$.