

# RBE595 - Week 5 Assignment

Keith Chester

Due date: February 12, 2023

## Problem 1

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

Monte-Carlo is best applied to problems where the state is not well known and not easy/possible to simulate, and/or the environment has stochastic outcomes. Likewise, Monte-Carlo excels in episodic situations where either many episodes can be ran through or simulated in order to train. Monte-Carlo methods provide an estimation of a reward given a pair of (state, action), usually via an average of returns for the resulting action of that state. In this way Monte-Carlo will eventually converge to the true value of an optimal policy.

## Problem 2

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

In Monte-Carlo prediction, the model performs its first update after the very first episode is completed. The returns are averaged for each (state, action) pair visited throughout the episode.

## Problem 3

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

Off-policy learning is an approach in reinforcement learning where the policy utilized to explore and generate data (behavior policy) is different from the policy utilized for evaluation (target policy), unlike on-policy learning. This approach is useful, as it empowers the agent to learn a broader range of experiences.

Expanding on this, the on-policy can be described as soft, meaning that  $\pi(a|s) > 0$  for all  $s \in S$  and all  $a \in A(S)$ , gradually moving close to the optimal policy. If we hope to continue exploration to continue to find a more optimal policy, we have to use a less optimal policy (like  $\epsilon$ -greedy) such that we sometimes perform unoptimally. Off-policy learning has the behavior policy explore for us, so the target policy can act optimally. This can lead to more stable behavior from the target policy than an on-policy approach could experience.

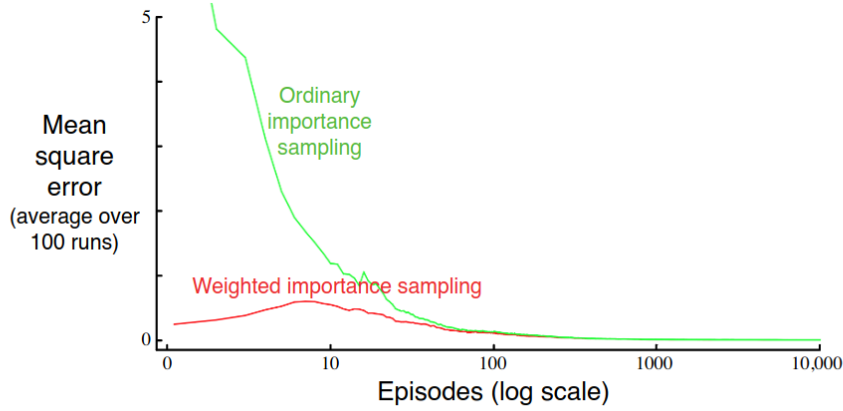
## Problem 4

*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?*

In this problem, first we consider the first-look estimator as it's the simplest. Here,  $V = 10$  as that is the outcome of the episode.

For the every-visit estimator, we must consider the reward we've accumulated at each step and average them. Thus, the every-visit estimator is calculated as:

$$V = \frac{10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1}{10} = \frac{55}{10} = 5.5 \quad (1)$$



**Figure 5.3:** Weighted importance sampling produces lower error estimates of the value of a single blackjack state from off-policy episodes. ■

Figure 1: figure 5.3 from the book

## Problem 5

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?

Weighted importance sampling has an initial bias that results in an increase in error at the beginning, but then ultimately fades away through training.

## Problem 6

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?

In this example, we can represent each visit through the equation:

$$E_b \left[ \prod_{t=0}^{T-1} \left( \frac{\pi_t(A_t|S_t)}{\pi_b(A_t|S_t)} G_0 \right)^2 \right] \quad (2)$$

...where  $\pi_t$  is our target policy and  $\pi_b$  is our behavior policy. Future visits would utilize the equation:

$$E_b \left[ \frac{1}{T-1} \sum_{k=1}^{T-1} \prod_{t=0}^{T-1} \left( \frac{\pi_t(A_t|S_t)}{\pi_b(A_t|S_t)} G_0 \right)^2 \right] \quad (3)$$

Thus we can begin calculating each episode:

...our first episode is:

$$E_{b_0} = \frac{1}{2} (0.1) \left( \frac{1}{0.5} \right)^2 \quad (4)$$

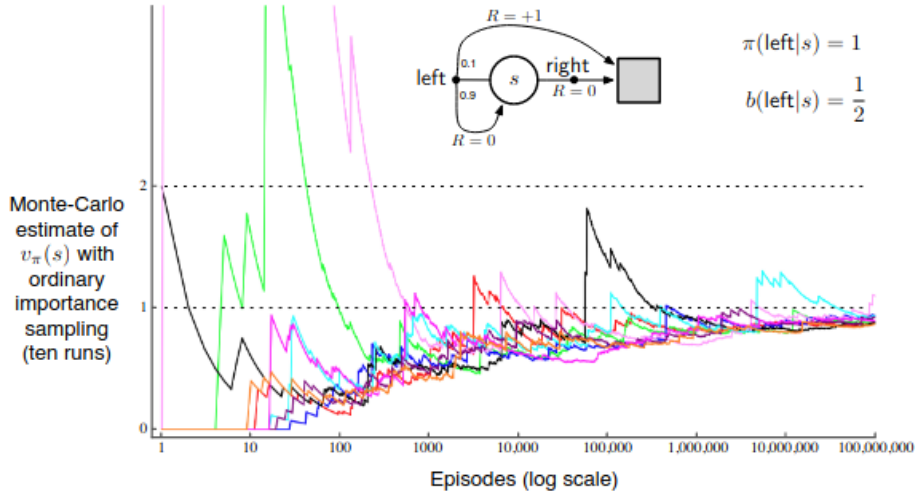
...our second episode contains our first visit:

$$E_{b_1} = \frac{1}{2} \left( \frac{1}{2} (0.9) \frac{1}{2} (0.1) \left( \frac{1}{0.5} \frac{1}{0.5} \right)^2 + E_{b_0} \right) \quad (5)$$

...where our second visit is represented as that part multiplied by the fraction (or  $2 * E_{b_1}$ )...

...and our third equation contains the first two:

$$E_{b_2} = \frac{1}{3} \left( \frac{1}{2} (0.9) \frac{1}{2} (0.9) \frac{1}{2} (0.1) \left( \frac{1}{0.5} \frac{1}{0.5} \frac{1}{0.5} \right)^2 + 2 * E_{b_1} \right) \quad (6)$$



**Figure 5.4:** Ordinary importance sampling produces surprisingly unstable estimates on the one-state MDP shown inset (Example 5.5). The correct estimate here is 1 ( $\gamma = 1$ ), and, even though this is the expected value of a sample return (after importance sampling), the variance of the samples is infinite, and the estimates do not converge to this value. These results are for off-policy first-visit MC.

Figure 2: figure 5.4 from the book

...and so on. We can represent this as

$$0.1 \sum_{k=1}^{\infty} \frac{1}{k} \sum_{i=0}^{k-1} (0.9)^i 2^i 2 \quad (7)$$

...which we can simplify by bringing the 2 to the front and combining our  $n^i$  values:

$$0.2 \sum_{k=1}^{\infty} \frac{1}{k} \sum_{i=0}^{k-1} (1.8)^i \quad (8)$$

Since we see that we are taking the sum an infinite amount of times, we can safely say that this is infinite variance.

## Problem 7

Derive the weighted-average update rule (5.8) from (5.7). Follow the pattern of the derivation of the unweighted rule (2.3).

We start with, from 5.7, the equation for  $V_n$ :

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, n \geq 2 \quad (9)$$

...and also reference 5.8, our update rule, which is our target:

$$V_{n+1} \doteq V_n + \frac{W_n}{C_n} (G_n - V_n), n \geq 1 \quad (10)$$

also, we know that

$$C_{n+1} \doteq C_n + W_{n+1} \dots \quad (11)$$

...where  $C_0 \doteq 0$  and  $V_1$  is arbitrary and doesn't need to be specified, where  $C$  represents the sum of our weights. We can then express start with  $V_{n+1}$ , and derive:

$$V_{n+1} \doteq \frac{\sum_{k=1}^n W_k G_k}{\sum_{k=1}^n W_k} \quad (12)$$

...here we pull out the last term, adjusting our summations:

$$V_{n+1} = \frac{W_n G_n + \sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^n W_k} \quad (13)$$

...in the next step we multiply the equation by a factor that equals one to introduce  $\sum_{k=1}^{n-1} W_k$  to our setup:

$$V_{n+1} = \frac{W_n G_n + \sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^n W_k} \frac{\sum_{k=1}^{n-1} W_k}{\sum_{k=1}^{n-1} W_k} \quad (14)$$

...multiplying this through we find:

$$V_{n+1} = \frac{W_n G_n + \sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k} \frac{\sum_{k=1}^{n-1} W_k}{\sum_{k=1}^n W_k} \quad (15)$$

...we can then separate the fraction and we find that we have found the definition of  $V_n$  within our derivation, so simplify to that:

$$V_{n+1} = \left( \frac{W_n G_n}{\sum_{k=1}^{n-1} W_k} + V_n \right) \frac{\sum_{k=1}^{n-1} W_k}{\sum_{k=1}^n W_k} \quad (16)$$

...then we can introduce  $C_n$  and  $C_{n-1}$  as, again, they are the sum of our weights:

$$V_{n+1} = \left( \frac{W_n G_n}{C_{n-1}} + V_n \right) \frac{C_{n-1}}{C_n} \quad (17)$$

...multiply through:

$$V_{n+1} = \frac{W_n G_n + V_n C_{n-1}}{C_n} \quad (18)$$

We can use a trick to add  $V_n$  to the equation below by also removing it, so our equation is still equivalent.

$$V_{n+1} = \frac{W_n G_n + V_n C_{n-1}}{C_n} + V_n - V_n \quad (19)$$

We rearrange to keep  $V_n$  up front and integrate the  $-V_n$  to our fraction. We're going to break apart the fraction as well to keep things easier.

$$V_{n+1} = V_n + \frac{W_n G_n}{C_n} + \frac{V_n C_{n-1} - V_n C_n}{C_n} \quad (20)$$

$$V_{n+1} = V_n + \frac{W_n G_n}{C_n} - \frac{V_n W_n}{C_n} \quad (21)$$

...finally we can pull out  $\frac{W_n}{C_n}$  as like terms:

$$V_{n+1} = V_n + \frac{W_n}{C_n} \left( G_n - V_n \right) \quad (22)$$

...which is what we set out to prove!

## Problem 8

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?

Since  $\pi$  is a deterministic policy, the probability of it being taken is always 1. Thus  $\frac{1}{b(A_t|S_t)}$  is still correct.