Garrett
Advanced Reinforcement Learning
February 28, 2022

Exercise 1

Question 1. (a) Modify the algorithm for first-visit MC policy evaluation (Section 5.1) to use the incremental implementation for sample averages described in Section 2.4. (b) The pseudocode for Monte Carlo ES is inefficient because, for each state-action pair, it maintains a list of all returns and repeatedly calculates their mean. It would be more efficient to use techniques similar to those explained in Section 2.4 to maintain just the mean and a count (for each state–action pair) and update them incrementally. Describe how the pseudocode would be altered to achieve this.

(a) We are shown in Section 4.1 that incremental averaging can be implemented in the following formula.

$$Q_{n+1} = Q_n + \frac{1}{n}\Big(R_n - Q_n\Big) \tag{1}$$

Therefor we can modify the first visit algorithm to be the following.

(i) Given a random policy π, initialize $V(s)$ arbitrarily.

(ii) In an endless loop, generate an episode following π, set $G = 0$, loop backwards for each step of the episode $T-1, T-2, ...0$, update G to be $\gamma G + R_{t+1}$, and unless S appears in an earlier state, update that state-value function using the incremental averaging formula written above.

(b) We will use the same equation in answer (a) to update the state-action value function, and obtain the following algorithm.

(i) Initialize $\pi(s)$ arbitrarily and initialize $Q(s, a)$ arbitrarily. Note how in the altered algorithm, we do not maintain a list of every state-action return to do the averaging.

(ii) In an endless loop, do the following. Choose a starting (s,a) pair such that all pairs have a probability greater than zero. Generate an episode following π starting from (s,a). Set $G = 0$ and loop backwards for each step of the episode $T-1, T-2, ...0$ while updating G to be $\gamma G + R_{t+1}$. Finally, unless S appears in an earlier state, update that state-value function using the incremental averaging formula written in answer (a).

Question 2. (a) Suppose every-visit MC was used instead of first-visit MC on the blackjack task. Would you expect the results to be very different? Why or why not? (b) 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?

(a) Both methods would be the same since no state can occur twice within the same episode as a result of the monotonically increasing characteristic of blackjack sums.

(b)

$$\textit{first-visit} = 10, \quad \textit{every-visit} = \frac{(\sum_{i=1}^{10} i)}{10} = 5.5 \tag{2}$$

Question 3. (a) Implement first-visit Monte-Carlo policy evaluation (prediction). Apply it to the Blackjack environment for the "sticks only on 20 or 21" policy to reproduce Figure 5.1.

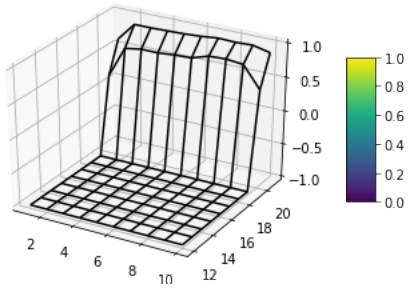

FIGURE 1. (a) No usable Ace

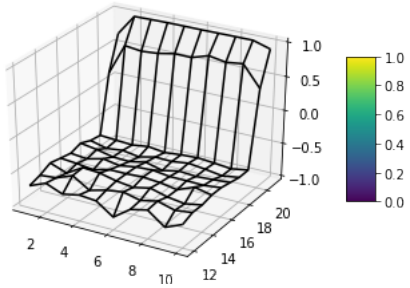

FIGURE 2. (a) Usable Ace

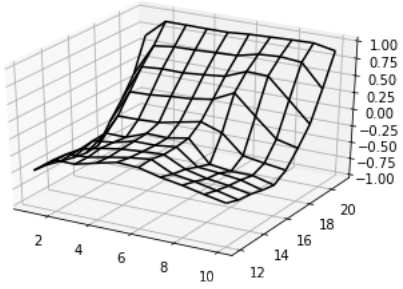

FIGURE 3. (b) No usable Ace

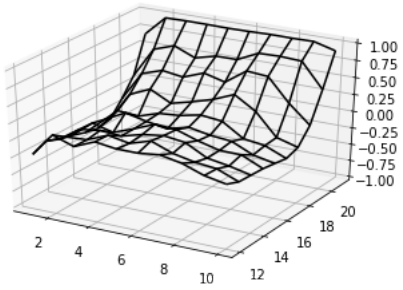

FIGURE 4. (b) Usable Ace

Question 4. *Four rooms*

(a) *Plots for (a) and (b) are below, and the answer to question (c) is here.* When epsilon is 0, following the current policy may not lead the agent to explore all states, and thus the policy is in that way naive.

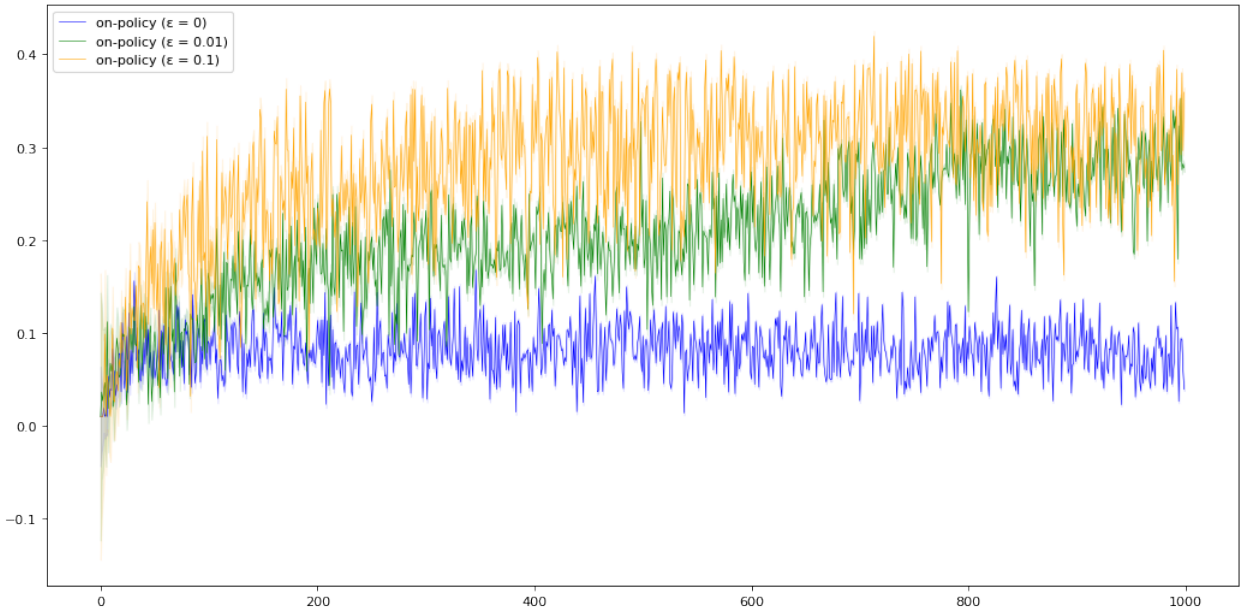

FIGURE 5. (b) Usable Ace

Question 5. (a) Derive the weighted-average update rule (Equation 5.8) from (Equation 5.7). Follow the pattern of the derivation of the unweighted rule (Equation 2.3). (b) In the boxed algorithm for off-policy MC control, you may have been expecting the W update to have involved the importance-sampling ratio, but it does not. Why is this correct?

(a)

(3)

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

(b) The probability of taking that action is always one, since we exit the inner loop if that action does not reflect our target policy.

Question 6. Racetrack

(a) *See Figure 6.*

(b) *See Figure 6.*

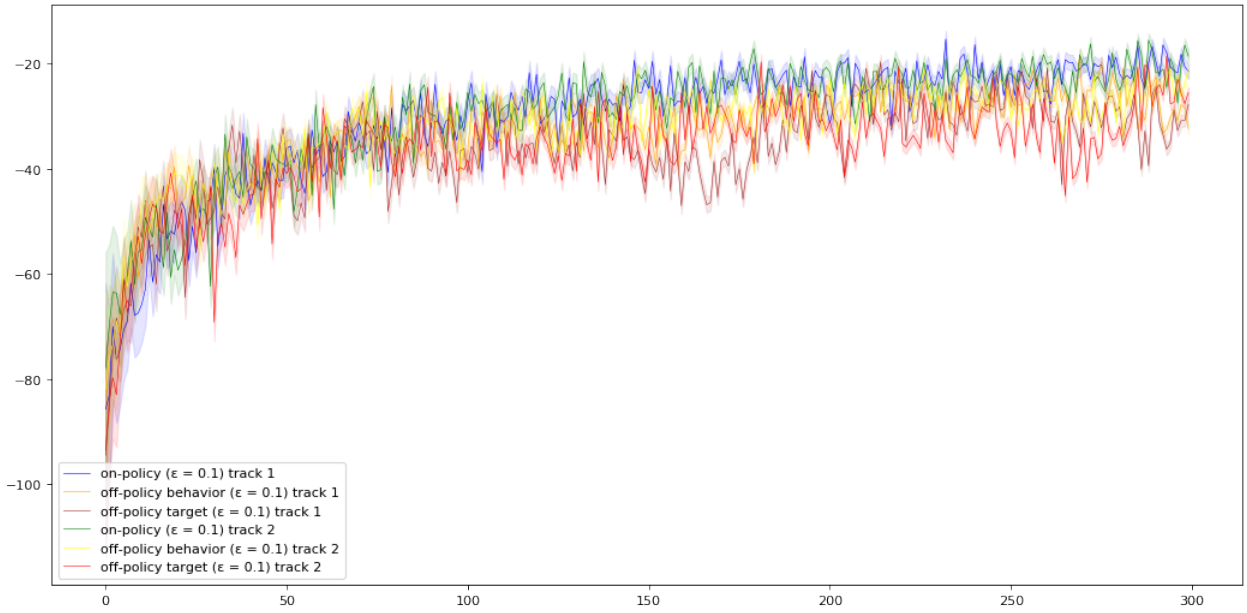

FIGURE 6. Off policy and on policy reward for both tracks

(c) I ran 50 trials for on-policy, off-policy behavior, and off-policy target, for both tracks creating a total of six lines and confidence bands. Looking at the plot, on policy policies performed better than off policy policies. I expect this to reverse if I had ran the trials for more time steps, say 2000 instead of 300. I also can see that race track had better rewards. I suspect this is an effect of the sharper turn in track 1, which takes the model more time to learn and also more time steps (and thus negative reward accumulation) to alter its velocity.