National Taiwan University
Department of Electrical Engineering
Name: Pei-Fan Liu
ID: R13921A02

113-1 (Fall 2024)
Reinforcement Learning

Assignment 2 Report

Q1. Run Monte-Carlo prediction and TD(0) prediction for 50 seeds. Compare the resulting values with the GT values. Discuss the variance and bias.

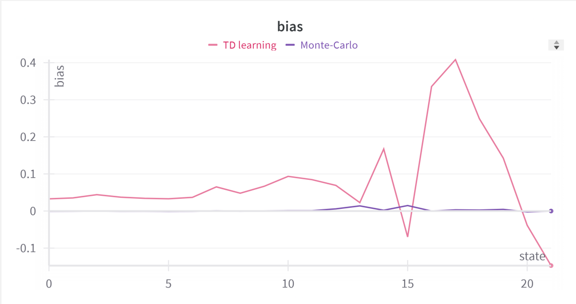

Figure 1: Average Bias of 50 Seeds

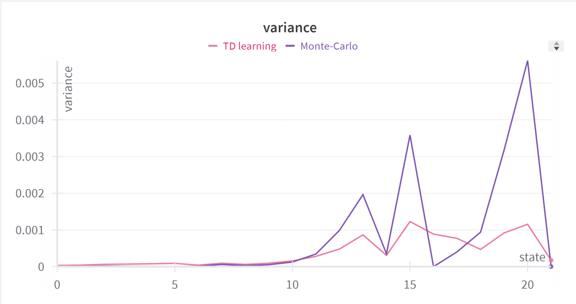

Figure 2: Average Variance of 50 Seeds

To compare the resulting values with the GT values, I run both prediction methods for 50 seeds. Figure 1. shows the average bias of 50 seeds for each state, and Figure 2. shows the average variance for each state.

TD prediction has the following updating formula:

$$V(s_t) = V(s_t) + \alpha[R_{t+1} + \gamma V(s') - V(s_t)] \tag{1}$$

This method is highly biased from the GT values, since it is based on bootstrapping. The target value $V(s')$ used in training is not necessarily near the ground true value, which causes this phenomenon. As shown in figure 1, for each state TD learning has higher bias value. The bias propagates along the learning path, so the bias are higher for the states further from the starting state.

Monte-Carlo prediction has the following updating formula:

$$V(s_t) = V(s_t) + \alpha[G_t - V(s)] \tag{2}$$

where $G_t = \gamma G_t + R_{t+1}$

This method has higher variance compared to the GT values, but it is not biased. Differ from TD learning, Monte-Carlo prediction updates the value function based on complete episodes and involves in more states. Each of them contains a certain degree of randomness determined by the policy, which is in charge of choosing actions, resulting in high variance.

Q2. Discuss and plot learning curves under ϵ values of $(0.1, 0.2, 0.3, 0.4)$ on MC, SARSA, and Q-Learning

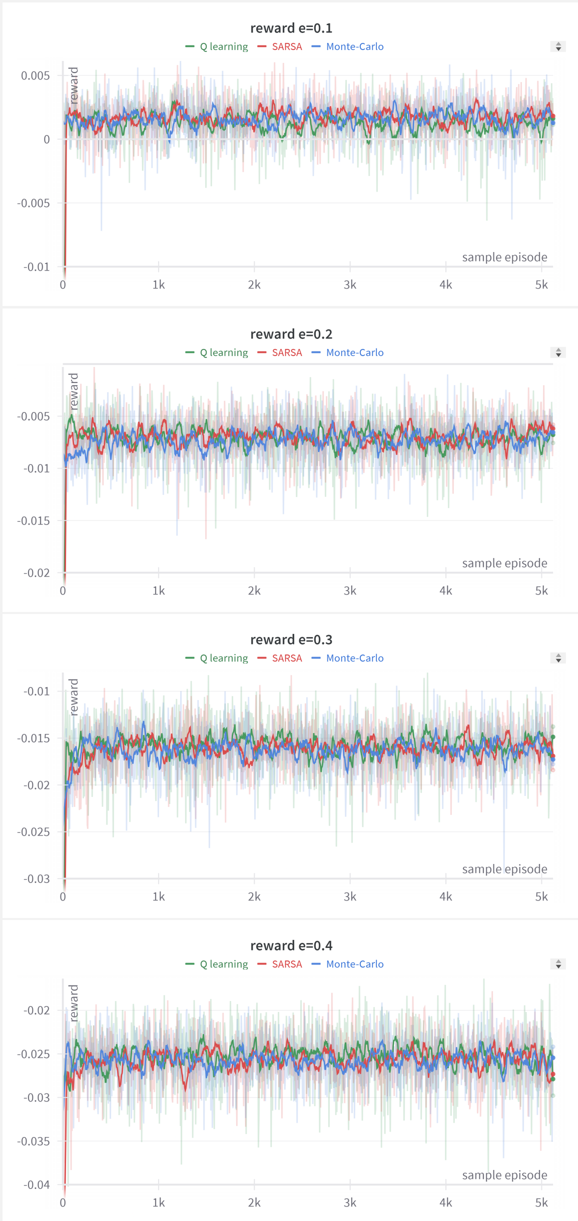

Figure 3: Learning Curve for Different ϵ

In this section, for each ϵ value, Monte-Carlo, SARSA and Q-learning are performed and their average non-discounted rewards are collected and plotted in separate plots. After running for 100 episodes, the average rewards from the last 10 episodes are collected for each sample time. Furthermore, to mitigate the noise interference, running average smoothing (n=10) is applied on the learning curves (solid curve). The results are shown in Figure 3.

When $\epsilon = 0.1$, the final average rewards fall between 0 to 0.005; as ϵ gets larger, the average rewards gets smaller and the oscillating amplitude gets larger. When $\epsilon = 0.4$, the final average rewards fall between -0.025 to -0.03. The reason behind this phenomenon is the simplicity of the test case. In general, we want the ϵ exploration factor larger to prevent from local optimal solutions. However, the given sample environment is relatively simple (only 21 states). It's best just to follow the currently optimal policy.

Also, three learning curves are oscillating around between 0 to 0.005 after around 100 episodes.Q learning obtains an overall high rewards when ϵ is high; Monte-Carlo can perform slightly better overall when ϵ is low, and it obtains a significantly higher reward at the beginning. This happens because Monte-Carlo obtains rewards depending on a whole episode. A more random policy may introduce significant variation to it, leading to poor performance.

Q3. Discuss and plot loss curves under ϵ values of $(0.1, 0.2, 0.3, 0.4)$ on MC, SARSA, and Q-Learning

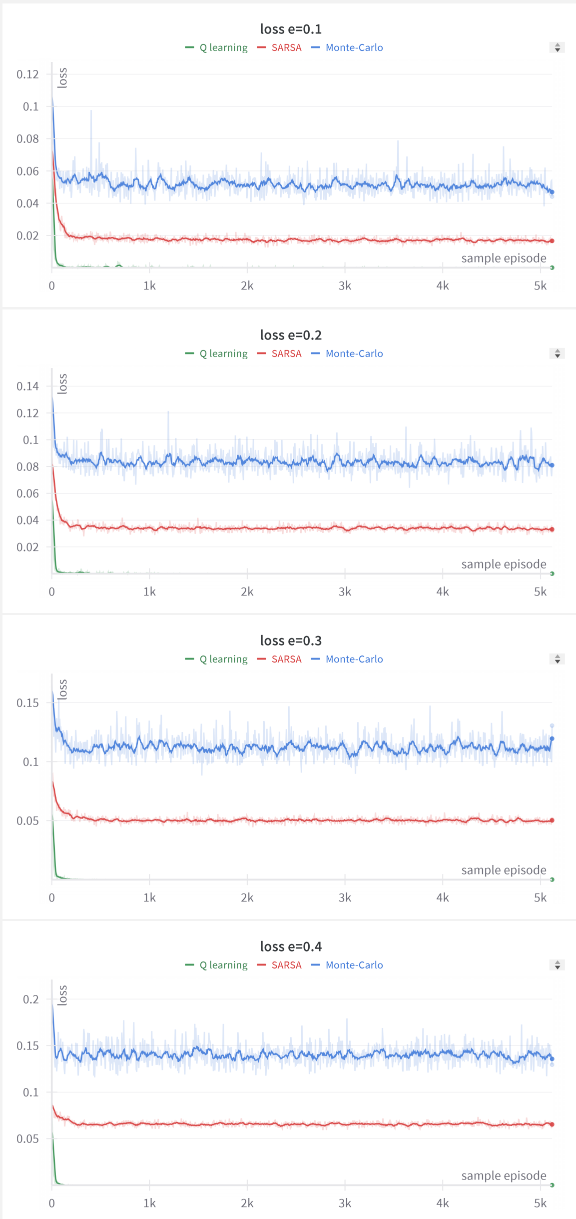

Figure 4: Loss Curve for Different ϵ

In this section, the absolute estimation loss obtained in each episode is collected. Again, the average loss value of last 10 episodes is collected every 100 episodes. For each ϵ value, three different control methods are evaluated and their losses are plotted separately, as shown in Figure 4

The estimation loss of Monte-Carlo prediction at step t is defined as:

$$EL_t = G_t - Q(s_t, a_t) \tag{3}$$

for SARSA, it is:

$$EL_t = R_t + \gamma Q(s', a') - Q(s_t, a_t) \tag{4}$$

for Q-learning, it is:

$$EL_t = R_t + \gamma \max_{a'} Q(s', a') - Q(s_t, a_t) \tag{5}$$

Q-learning obtains the least estimation loss, while Monte-Carlo has the highest loss. The observed data in a sequence are usually considered as non-iid. Q-learning uses replay buffer to randomly sample previous data to update q-values, which mitigate the above issue. Furthermore, Monte-Carlo estimation loss curve oscillates the most, which implies high variance during training.

Also, ϵ value influences estimation loss of Monte-Carlo method significantly. The higher ϵ value gets, the higher the loss becomes. This happens since Monte-Carlo relies on full-episode learning, and has more chance to explore randomly. This can be harmful to performance in this simple test case. However, ϵ value hardly affect the performance of Q-learning due to its updating policy. The loss curve of SARSA remains between those of the other two methods. These two methods are based on bootstrapping, which is the different strategy compared to Monte-Carlo control, and are less sensitive to the exploration factor.