Reinforcement Learning Homework 3

Lavanya Verma (lavanya18155@iiitd.ac.in)

October 25, 2021

1. Problem 1

Algorithm 1: Modified MC Prediction Algorithm

Initialise:

$\pi(s) \in A(s)(arbitrarily), \forall s \in S$

$Q(s,a) \in \mathbb{R}(arbitrarily), \forall s \in S, a \in A(s)$

$Q_{count}(s,a) = 1(arbitrarily), \forall s \in S, a \in A(s)$

Loop forever(for each episode): Choose $S_0 \in S$, $A_0 \in A(S_0)$ randomly such that all pair have probability > 0

Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless the pair S_t, A_t appear in $S_0, A_0, S_1, A_1, \ldots, S_{t-1}, A_{t-1}$:

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{(G - Q(S_t, A_t))}{Q_{count}(S_t, A_t)}$

$Q_{count} \leftarrow Q_{count} + 1$

$\pi(S_t) \leftarrow \text{argmin}_a \, Q(S_t, a)$

‘

The above algorithm handles the memory inefficiency of algorithm provided in the text book. It does so by using the incremental updates for each state-action pair instead of storing the returns.

2. Problem 3

$$Q(s,a) \doteq \frac{\sum_{t \in \tau(s,a)} \rho_{t:T(t)-1} G_t}{\sum_{t \in \tau(s,a)} \rho_{t:T(t)-1}}$$

$\tau(s,a)$ is set of time stamps at which state pair had occurred. $\rho_{t:T(t)-1}$ is the importance sampling ratio. G_t is discounted return upon following the behaviour policy.

3. Problem 4

Code is self explanatory. Dealer, Player and Game(BlackJack) are designed using oop framework. There are two functions one for MC prediction and one for MC control, then the rest of the two function are for visualisation.

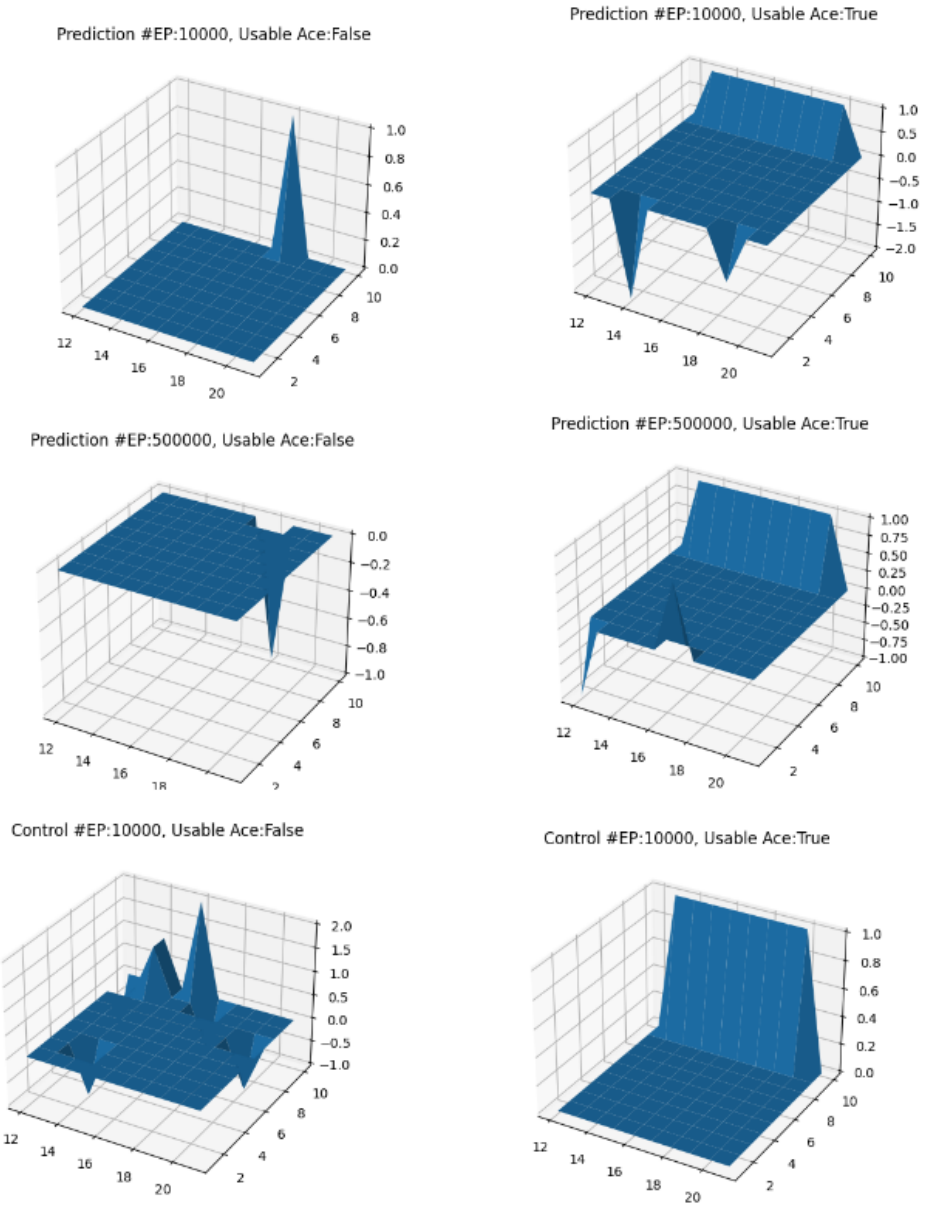

Figure 1: State Values

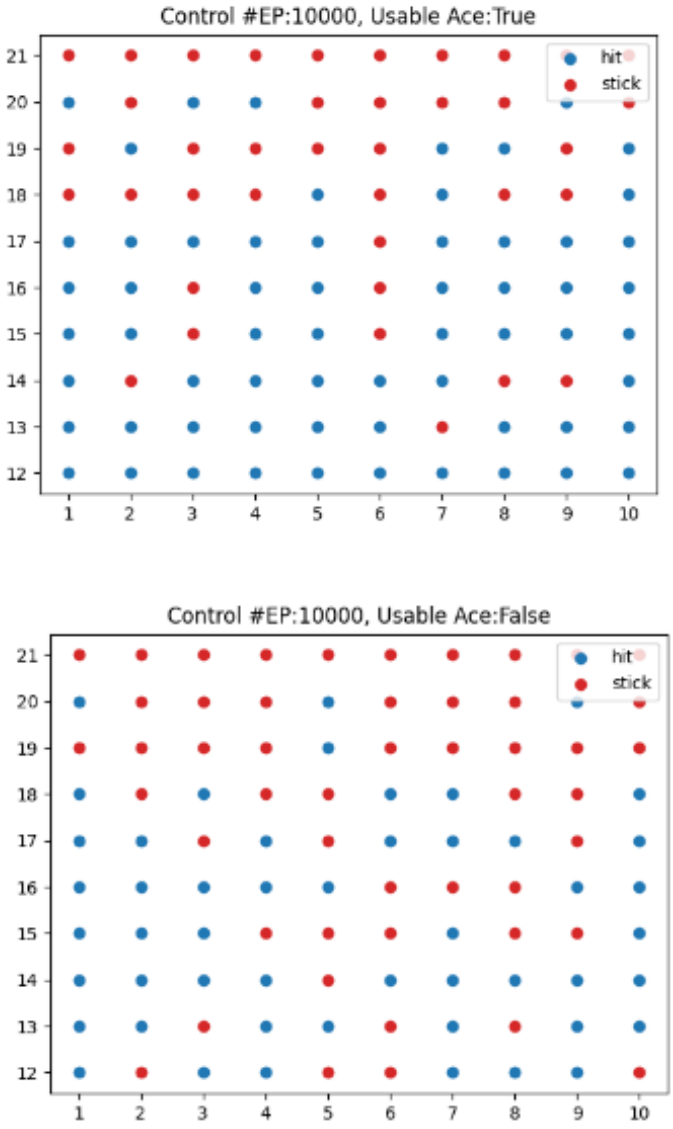

Figure 2: Policy

4. Problem 6

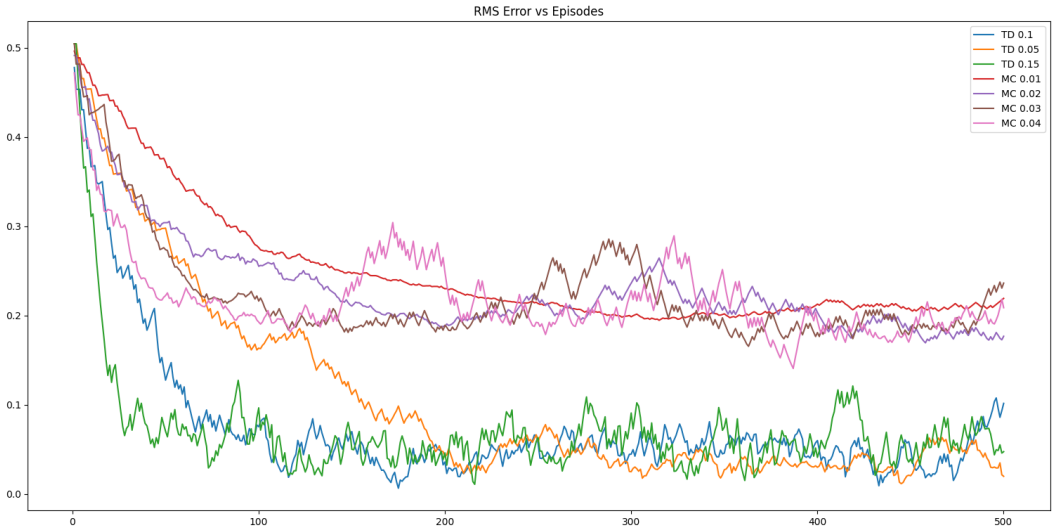

Figure 3: RMS Errors

5. Problem 7

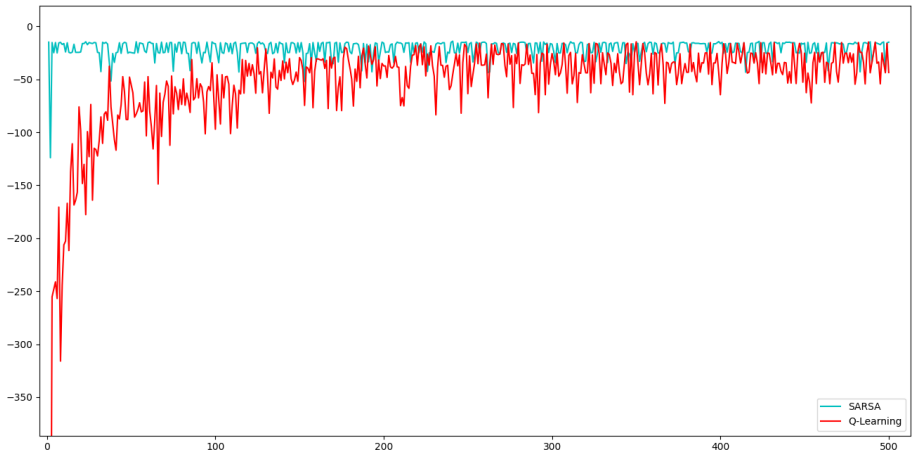

Figure 4: Sum of Rewards during episode

6. Problem 8

Even if both target policy and behaviour policy are same in qlearning and action selection is greedy in sarsa and qlearning. These algorithm would still not be same due to the following reasons.

- In Q-learning action at time t A_t is selected on the basis of state S_t and estimates Q_t, however in sarsa action at time t A_t is selected on the basis of state S_t and estimates Q_{t-1}.
- Incremental update rule of Qlearning uses the maximum value over any action of subsequent state, but sarsa uses the Q value of action selected in the subsequent state based on prior estimates.