Assignment 3
Reinforcement Learning

Vedant Bhatia
2016113

Question 1

Question 1 - Exercise 5.4

Monte Carlo ES, for estimating $\pi \approx \pi_*$

Initialize

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

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

$C(s,a) = 0 \quad \forall s \in S, a \in A(s)$

Loop forever (for each episode):

Choose $S_0 \in S$, $A_0 \in A(S_0)$ randomly such that all pairs have prob > 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 appears in $S_0, A_0, S_1, A_1 \ldots S_{t-1}, A_{t-1}$:

~~Append G to~~

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{1}{C(S_t, A_t) + 1}(G - Q(S_t, A_t))$

$C(S_t, A_t) \leftarrow C(S_t, A_t) + 1$

$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

The changes made in the code are to compute the value for the state-value pairs incrementally.

$Q(S_t, A_t) = average(Returns(S_t, A_t))$

in the inefficient version

$average(Returns(S_t, A_t)) = \frac{sum(Returns(S_t, A_t))}{length(Returns(S_t, A_t))}$

Where $length(Returns(S_t, A_t))$ is represented as $C(S_t, A_t)$ or the number of times the state-action pair has been first-visited

If we store the average (Returns (Q_t, A_t)) for a $C(S_t, A_t) = t$ as $Q(S_t, A_t)^t$, we can represent $Q(S_t, A_t)^{t+1}$ as

$$Q(S_t, A_t)^{t+1} = \frac{sum(Returns(S_t, A_t))}{C(S_t, A_t) + 1}$$

$$= \frac{sum(Returns(S_t, A_t)^t) + G}{C(S_t, A_t) + 1}$$

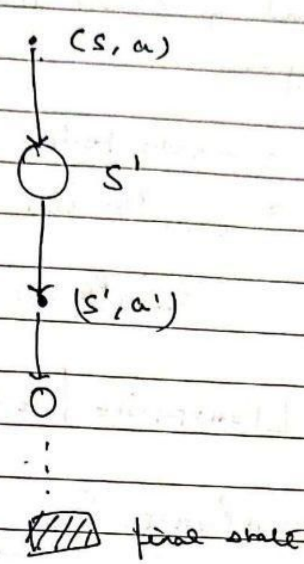
$$= \frac{C(S_t, A_t)\,Q(S_t, A_t)^t + G}{C(S_t, A_t) + 1}$$

$$Q(S_t, A_t)^{t+1} = Q(S_t, A_t)^t + \frac{(G - Q(S_t, A_t))}{C(S_t, A_t) + 1}$$

Hence we can represent the update using just the mean and a counter variable.

Question 2

2. The backup diagram for q_π:



The difference is that we consider state-action pairs. The first node represents the initial state action pair of an episode, and the remainder of the episode follows.

Question 3.
Exercise 5.6:

Q3. Equation analogous to (5-6) for action values $Q(s,a)$ instead of $V(s)$, given returns generated using b.

$(5.6) \rightarrow \cancel{V(s) = \sum_{t \in \mathcal{J}(s)} \rho_t T_t}$

$$V(s) = \frac{\sum_{t \in \mathcal{J}(s)} \rho_{t:T(t)-1} G_t}{\sum_{t \in \mathcal{J}(s)} \rho_{t:T(t)-1}}$$

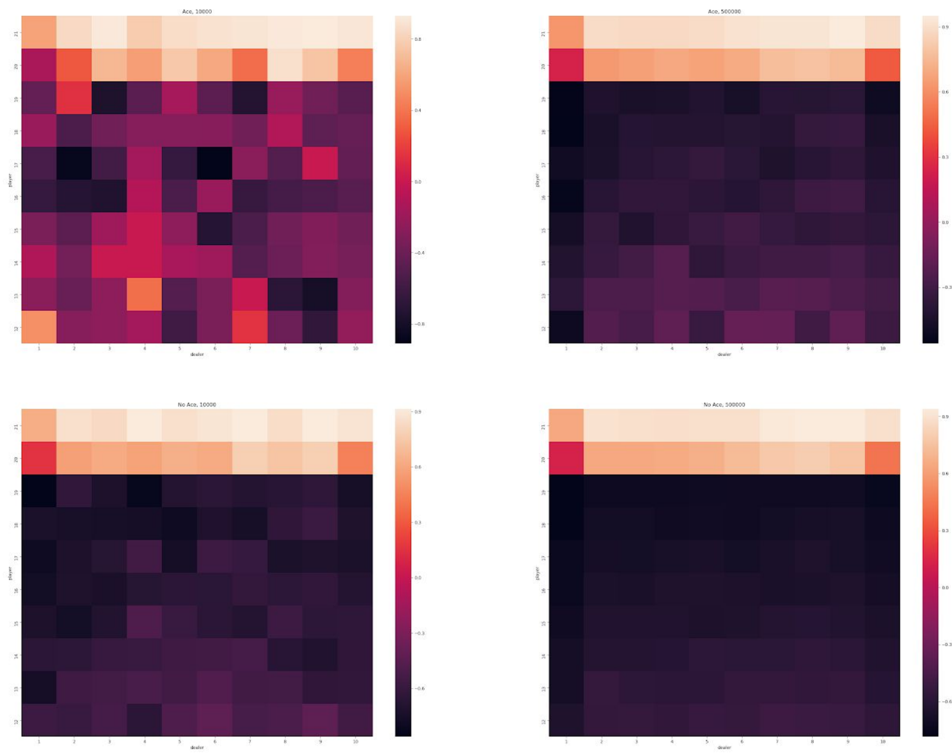
$$q_\pi(s,a) = E\left[\rho_{t+1:T-1} G_t \mid S_t = s, A_t = a\right]$$

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

T now represents times of termination for (s,a) instead of s.

Question 4

Blackjack Game
Figure 5.1

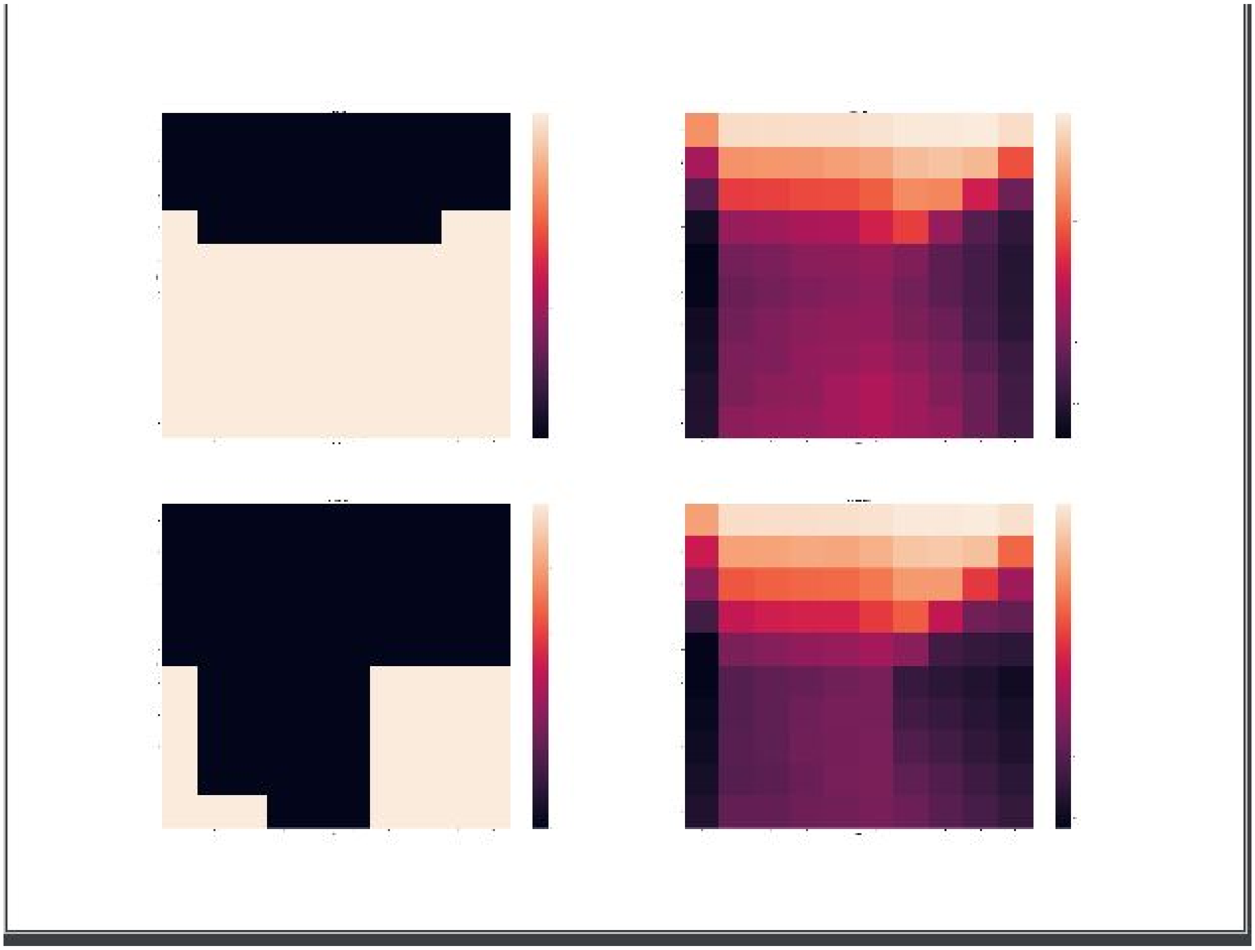


Here the figures from left to right, top to bottom are:

a) 10,000 episodes and with usable ace.
b) 500,000 episodes and a usable ace
c) 10,000 episodes and no usable ace
d) 500,000 episodes and no usable ac

The X axis of each figure is the dealer’s showing, and the y axis is the player sum.

Figure 5.2

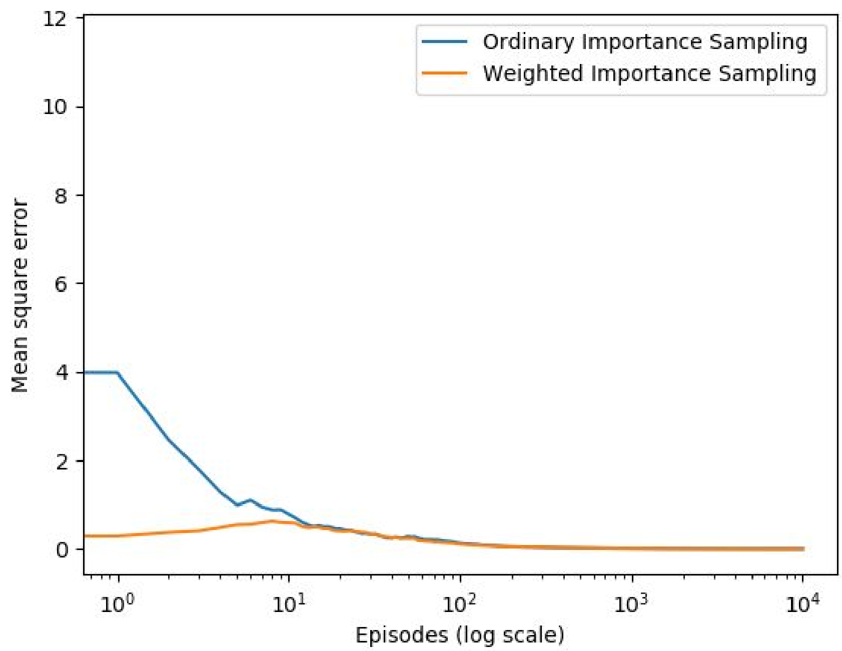


The figures from left to right, top to bottom are:

a) Policy (black for stick, white for hit) for a usable ace
b) Value function for a usable ace
c) Policy for no usable ace
d) Value function for a usable ace

As above, the X axis represents the dealer showing and Y axis shows player turn. The difference in the figures generated here and in the book is simply that this graph starts at 12 and not 11 like in the book. As the decision making starts at 11, i.e. policy is known always for 11 (as mentioned in the book as well), this does not change the information represented.

Figure 5.3

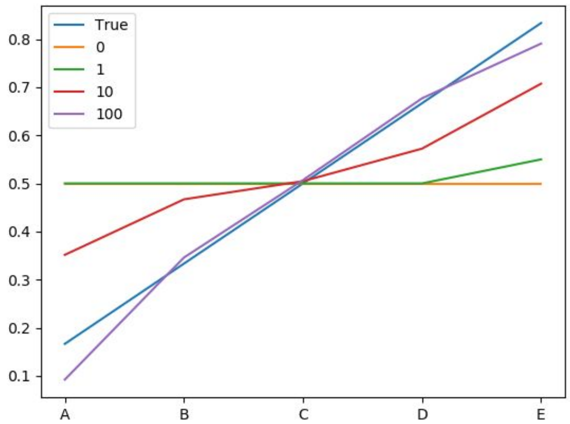


The code is explained with the help of comments.

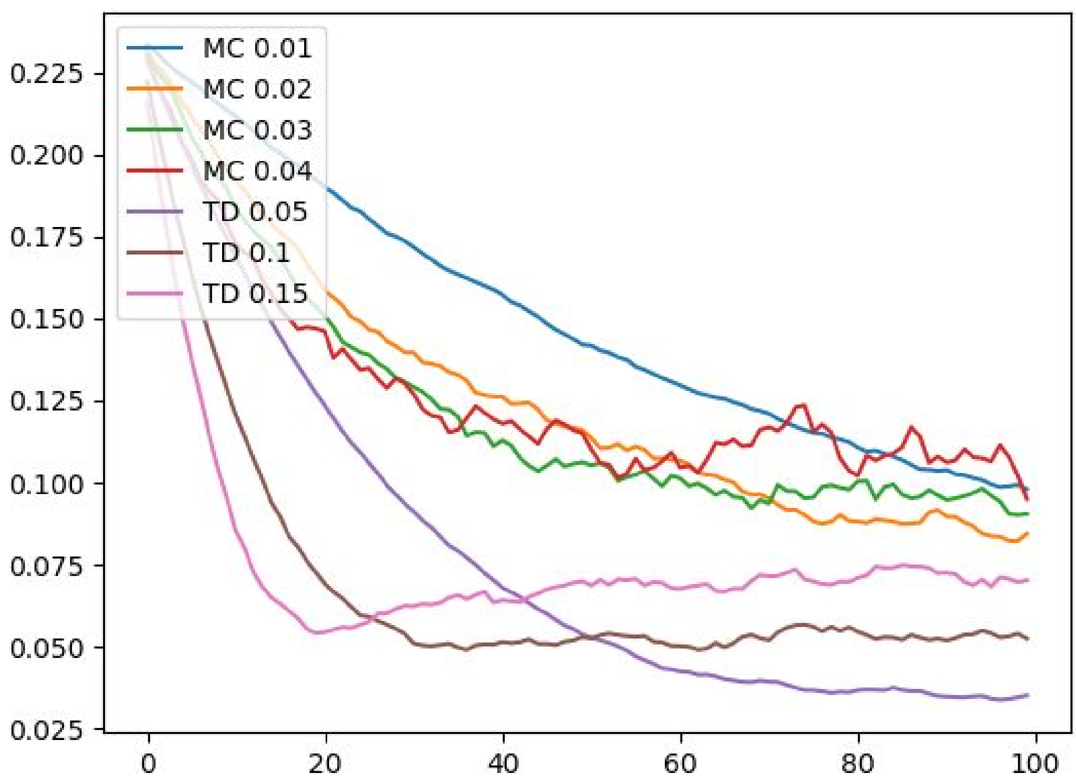
Question 5.

Exercise 6.2: Due to the bootstrapping and online learning nature of TD, it will be more efficient that MC methods. It uses the prior information we have. If we have a sequence of states S_1, … S_T which we have estimates on, then we can use this in TD for the sequence S_0,S_1,.. as TD uses bootstrapping i.e.set the values of the next state as the target, which we can initialize to the good estimates that we have. For example, many states are the same once the highway comes, and hence the value estimates we have should result in faster convergence when used as targets. On the other hand, MC methods need multiple entire episodes to learn about a state.

Question 6.



.



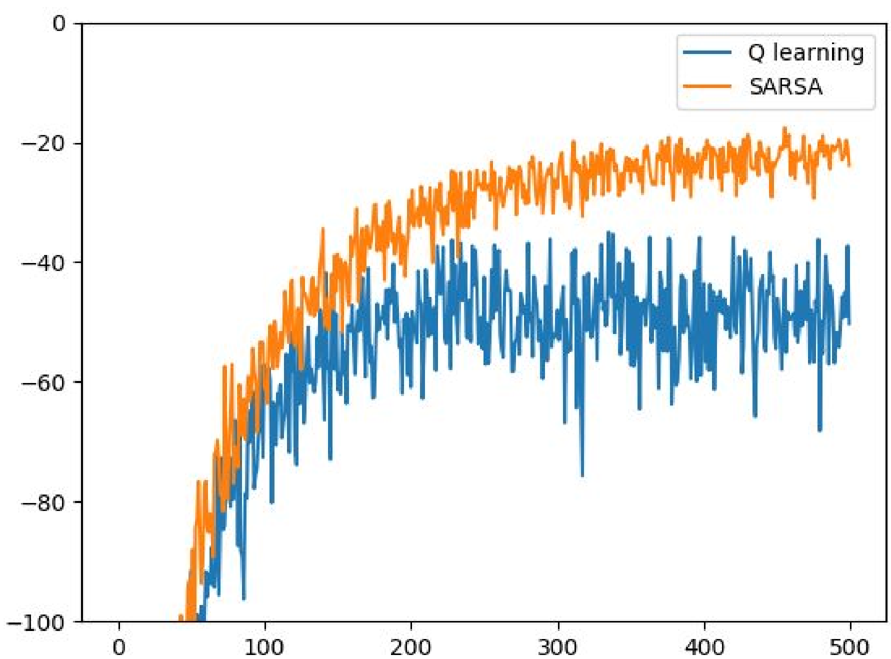
Exercise 6.3: The first episode results in a change only in V(A) because initially, all the states had the same values and the terminal states had value 0. Therefore TD(0) updates do nothing to states which have non-terminal transitions. In the first run, the agent terminated on the left, therefore the value of the state with a terminal transition, i.e. A, changes, while the rest do not.

It changed by alpha*(reward+V(left_terminal)-V(A)) = 0.1*(-0.5)=-0.05

Exercise 6.4: The benefits of TD prediction methods (in section 6.2) are independent of the alpha parameter. Keeping a lower alpha will allow more precise convergence at the cost of a slower convergence and higher alpha would increase the curve. The width of the range here shows us the general trend which is suitable for making decisions. There is no particular alpha at which it will perform better as the values are already quite small and smaller would not give a different result to this trend.

Exercise 6.5: At first, the values for the outer states change and hence the error reduces. We have initialized the value of C to its true value, so when the changes propogate to C, it is moved from its ideal value and hence the error increases, proportional to the alpha parameter. Later as we converge to the true value again, the error decreases. If it was not initialized at its true value, this may not have occurred but initial error would have been higher.

Question 7.



Question 8.

Exercise 6.12: No they would not be the same. In Q learning, we take the next action after updating the state action values, while in SARSA we take the next action before updating the action values. This means that in case the best option according to greedy policy changes after updating, the next action would differ for SARSA and Q learning.