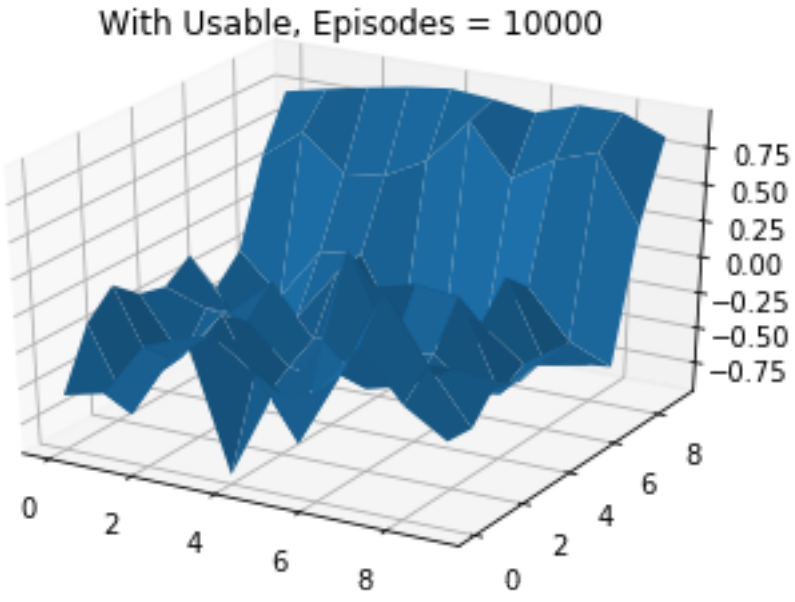
RL Assignment 3

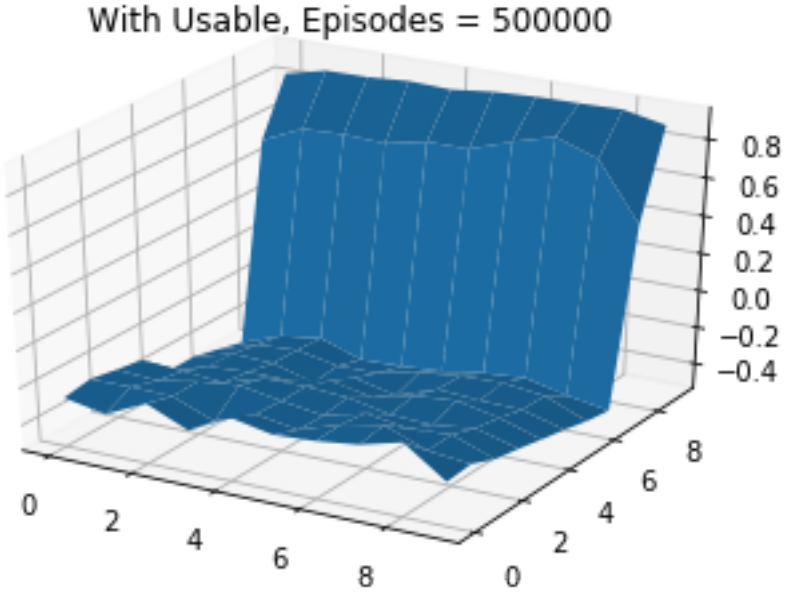
September 25, 2019

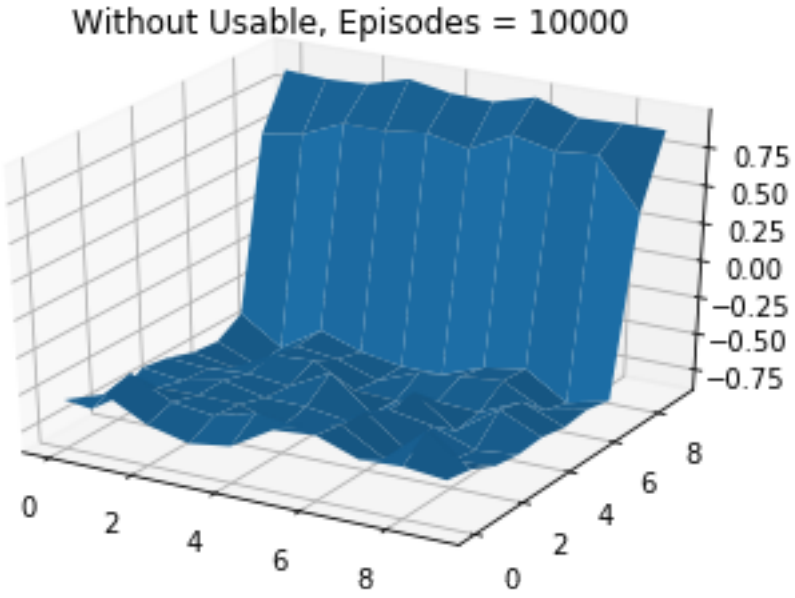
1 Black Jack Game

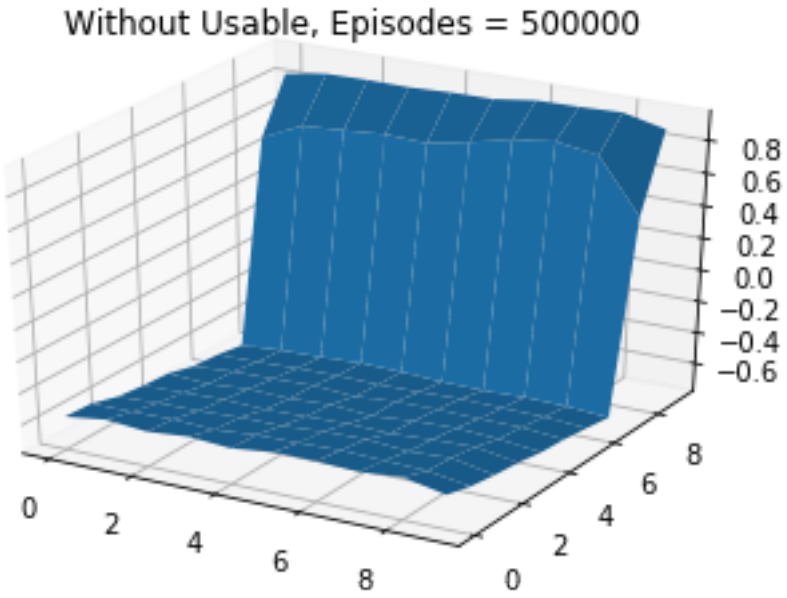
1.1 State Value Function using Policy Evaluation

The 3D surface plots for the state value functions of this problem are as follows:




With Usable, Episodes = 500000
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
2
4
6
8
0
2
4
6
8

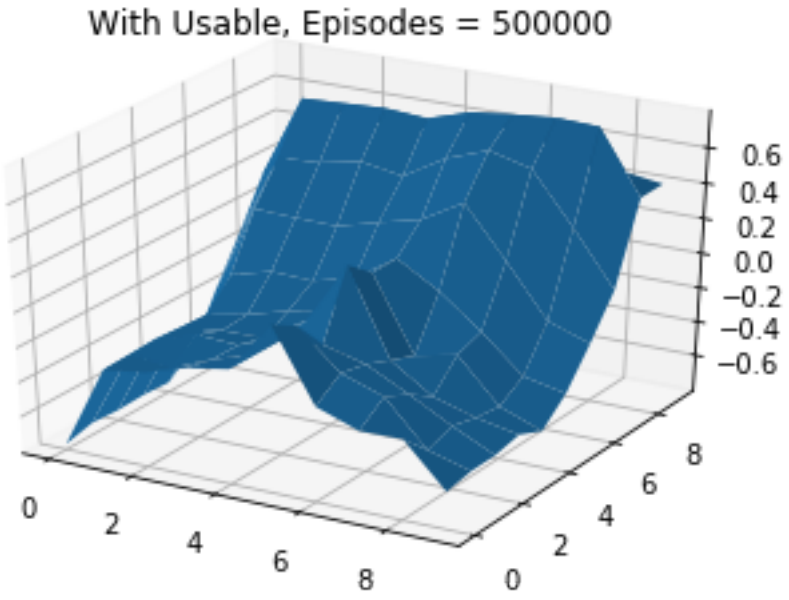

Without Usable, Episodes = 10000
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
0
2
4
6
8
0
2
4
6
8

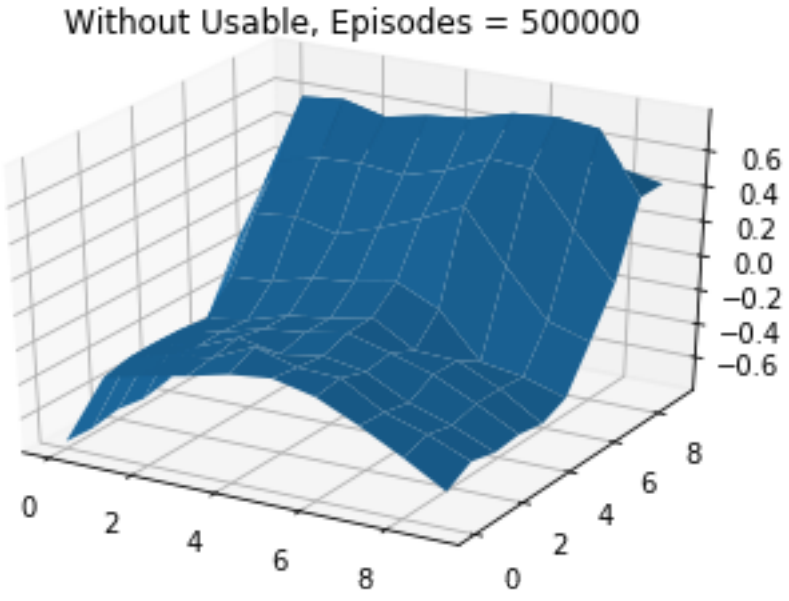


In this, we only evaluate the fixed user policy provided to us and find the state value function corresponding to the given policy.

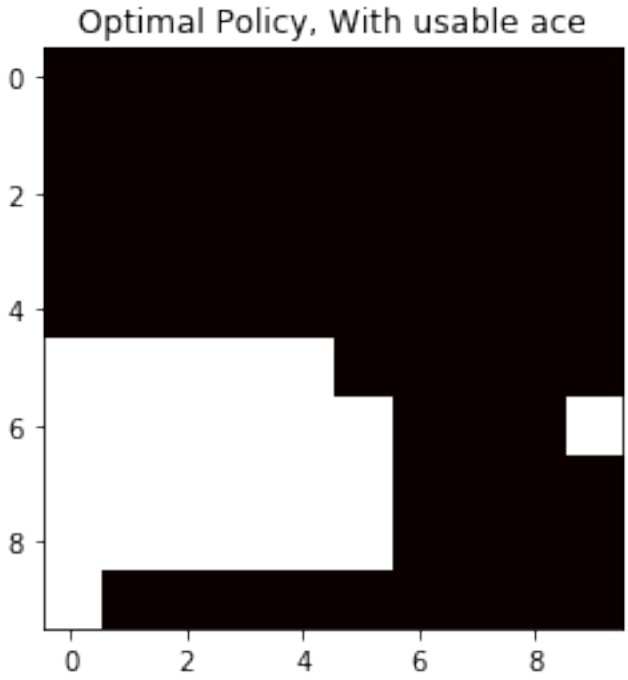
1.2 Exploring Starts

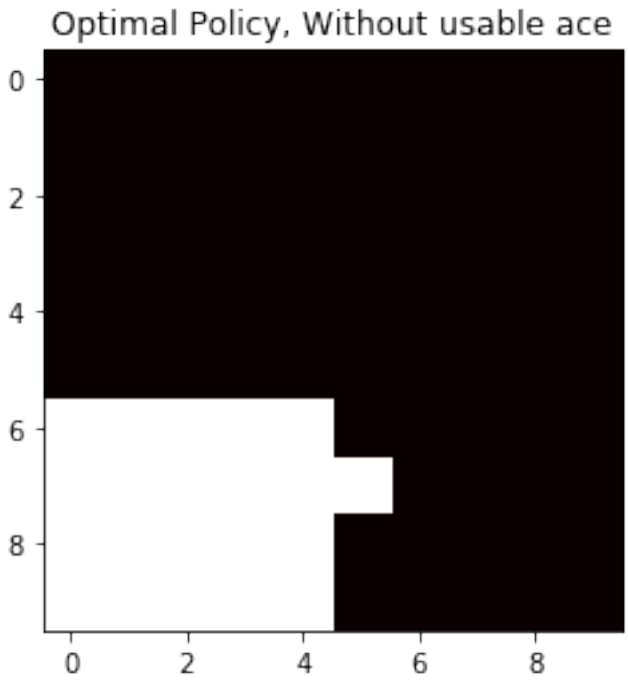
Exploring starts refers to the method of promoting exploration in which we begin by sampling random and different initial states. After sampling initial states, the normal policy followed to evaluate is same as the previous part, and the improvement is done by picking the greedy policy.





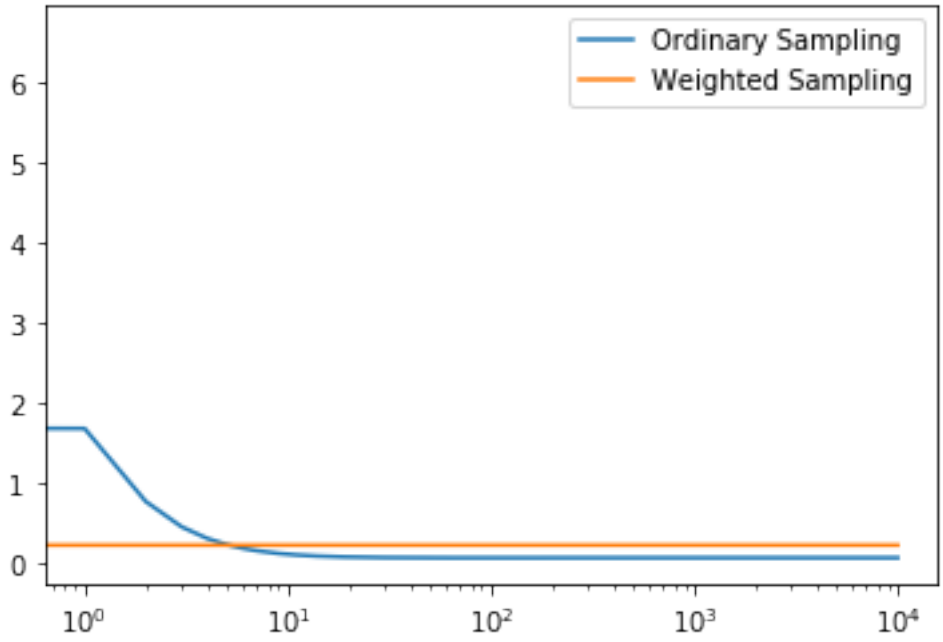
We obtain the policy as a 10x10 grid where each block has a 0/1, that suggests us the best possible action that one should perform when we are in the corresponding state.


Optimal Policy, With usable ace
0
2
4
6
8
0
2
4
6
8



1.3 Importance Sampling

In case of importance sampling, we make use of a behavioural policy i.e. a policy that picks random actions, and use the returns obtained from that in order to update the state action value function for our target policy (π in code). Even though, both eventually converge to the same optimal value the speed of convergence in case of ordinary is slower than that in weighted.


Ordinary Sampling
Weighted Sampling
6
5
4
3
2
1
0
10^0
10^1
10^2
10^3
10^4

2 Random Walk

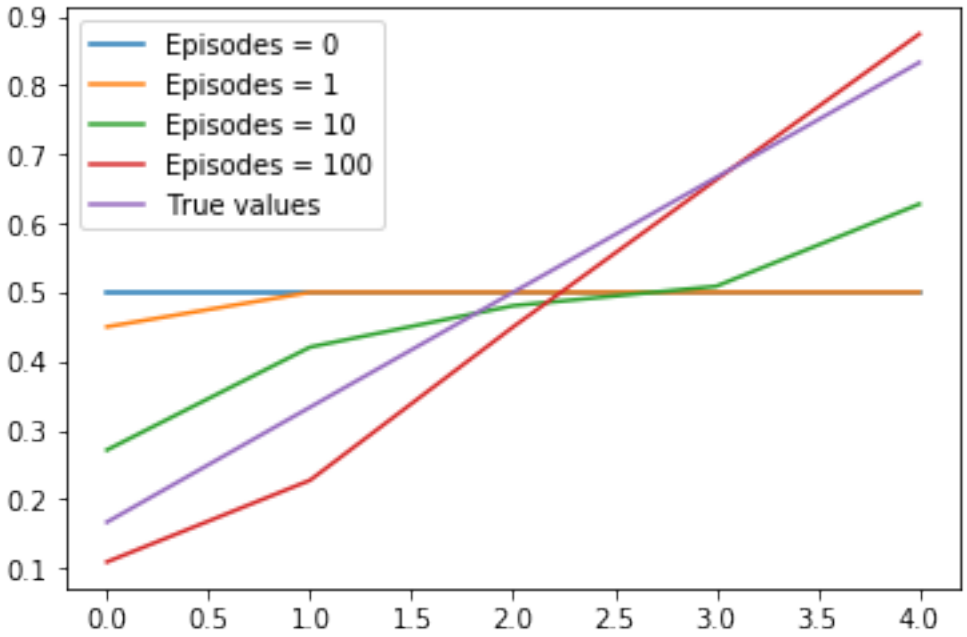


Figure 1: Plot for True and Estimated State Values for States A to E (along the x-axis represented as 0 to 4) for different number of episodes.

We observe that as the number of episodes increase, we progressively begin to get state values similar to the actual ones (Purple Line). Initially, we begin with the estimated values to be all equal to 0.5 (Blue Line) and later converge to a line with constant slope.

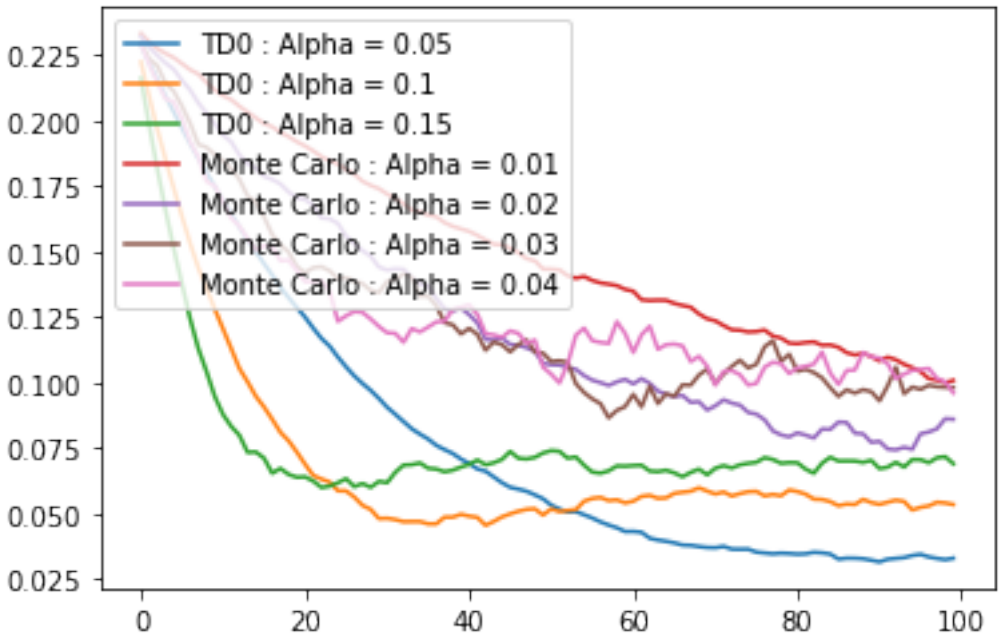


Figure 2: Plot for RMSE for Monte Carlo and TD0 Method for different α values for different episodes (along x-axis).

We observe that the root mean square error for TD0 is lower than that of Monte Carlo for almost all α values.

3 Q-Learning vs SARSA

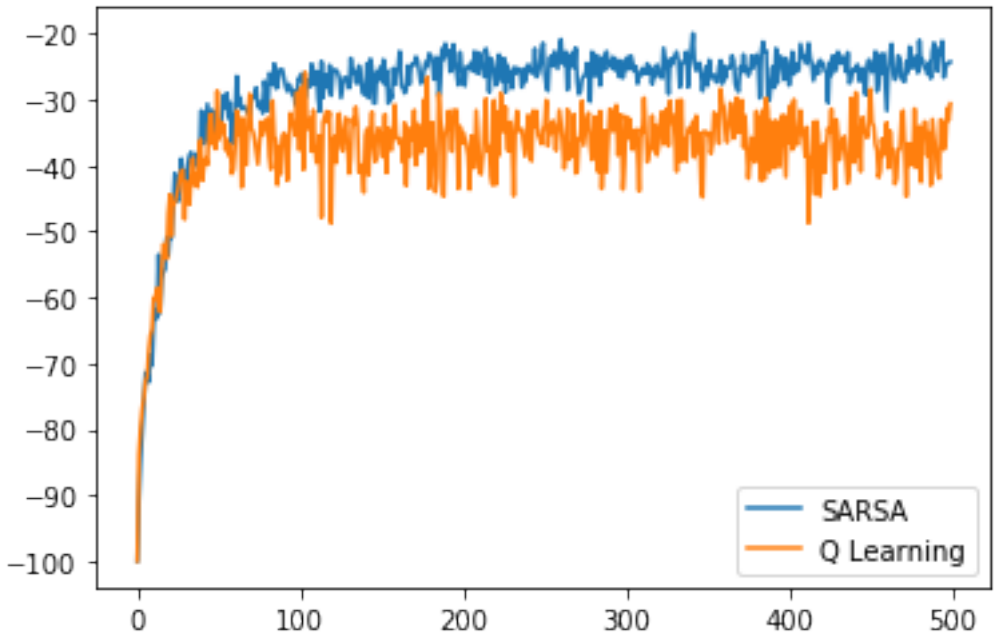


Figure 3: Plot for sum of all rewards in an episode for Q-Learning and SARSA methods for different number of episodes (along x axis).

We observe that SARSA has a larger sum of rewards as it follows a safer policy while Q-Learning tries to learn the optimal policy which might sometimes end up falling off the cliff (as optimal policy is just along the cliff). So, in this case SARSA performs better than Q-Learning.