RL - Assignment 1 - Black Jack

Samarth Aggarwal

September 29, 2019

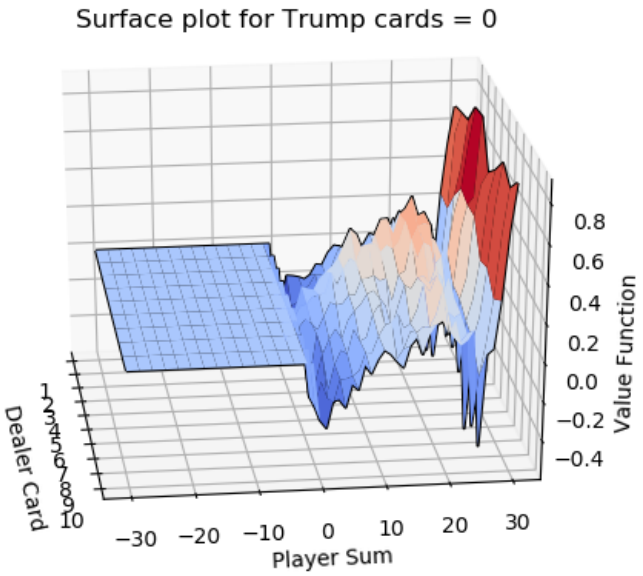
1 State Space

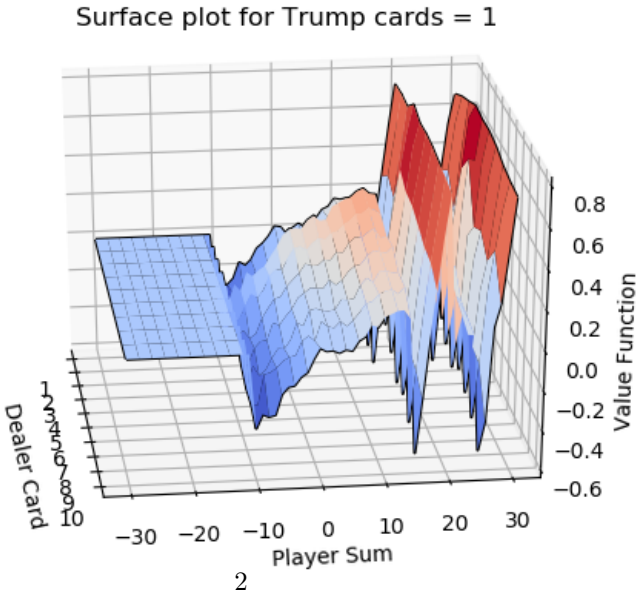
1. General states have 4 attributes:
   (a) raw sum = -30, .., -1, 0,1 ... , 30
   (b) black one = True, False
   (c) black two = True, False
   (d) black three = True, False

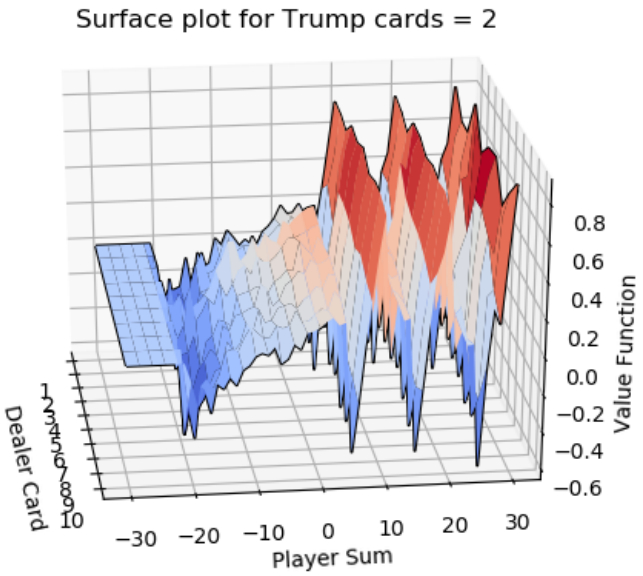
   The general states are actionable.
2. Special States = BUST, SUM31 The special states are non-actionable.
3. Total Combinations of raw sum and special black cards = $31 * (2^3) + 10 * (3C1 + 3C2 + 3C3) + 10 * (3C2 + 3C3) + 10 * (3C3) + 2$ - (20)= 350
4. Total Number of Dealer Cards = 10 If the dealer card is a red card (negative value) then the game ends then and there. There is no action to be made by the agent so they don't result in an actionable state.
5. Total Number of States = 350 * 10 = 3500
6. All states can be reached. General states are actionable and the two special states are non-actionable.

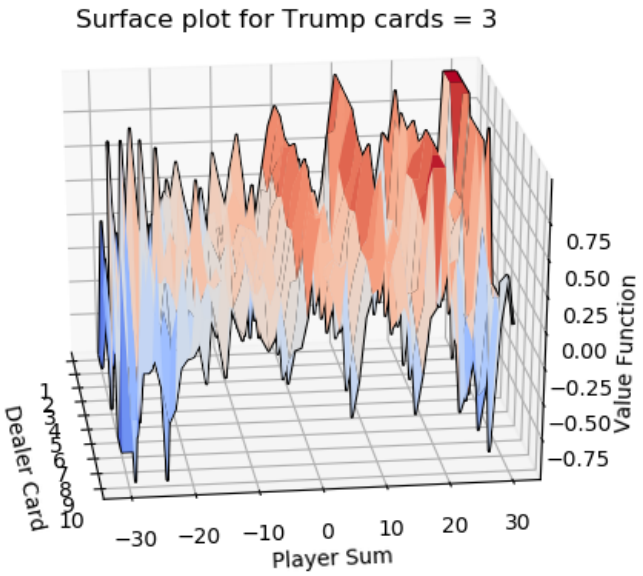
2 Policy Evaluation

2.1 Monte Carlo - First Visit Update

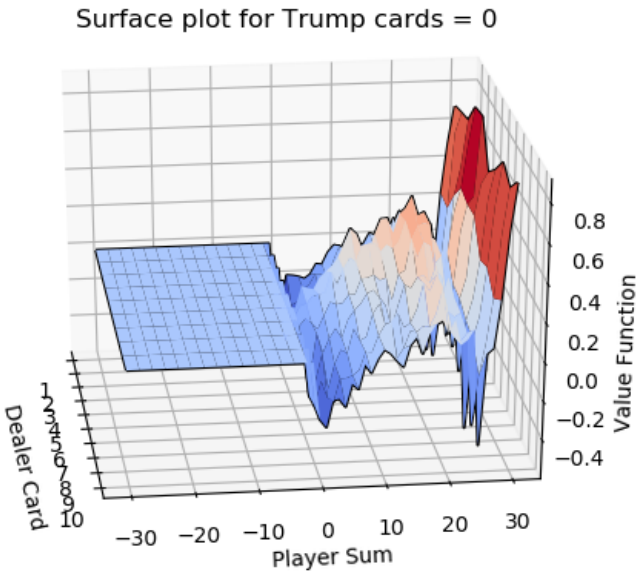


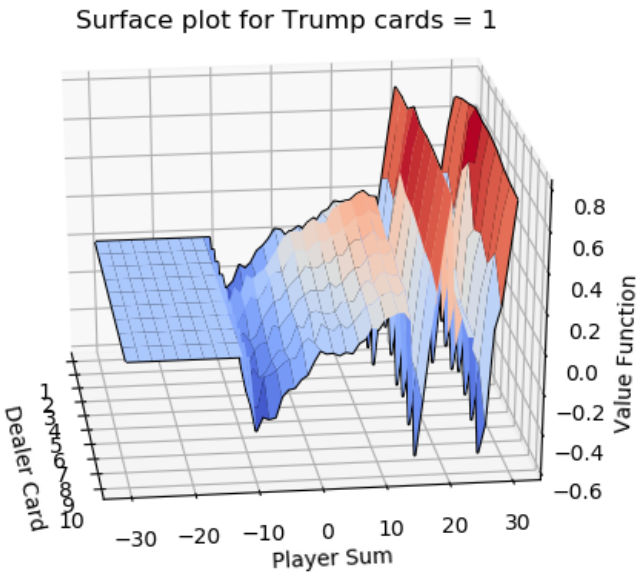


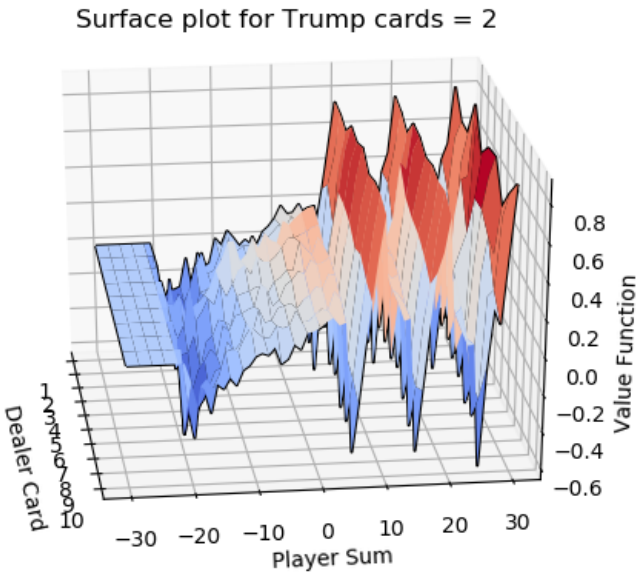

Surface plot for Trump cards = 2
Value Function
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Player Sum
−30
−20
−10
0
10
20
30
Dealer Card
10

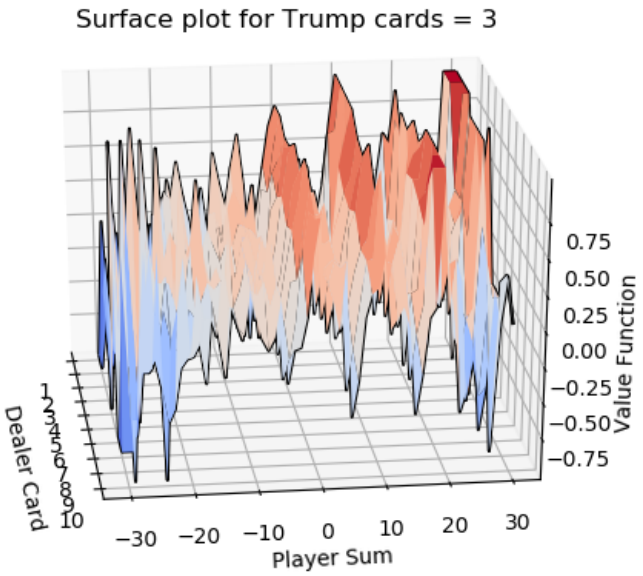

Surface plot for Trump cards = 3
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
Value Function
Dealer Card
10
−30
−20
−10
0
10
20
30
Player Sum

2.2 Monte Carlo - Every Visit Update






Surface plot for Trump cards = 2
Dealer Card
10
−30
−20
−10
0
10
20
30
Player Sum
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Value Function



2.3 K-Step TD

2.3.1 Variation with Number of Runs

The case with 100 runs is smoother as compared to the case with 10 runs. This is because as we increase the number of runs, the variance due to stochasticity decreases.

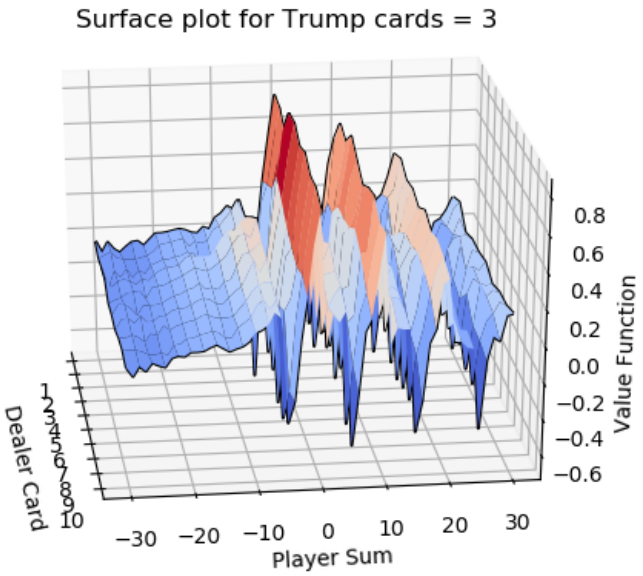


Figure 1: 10 Runs

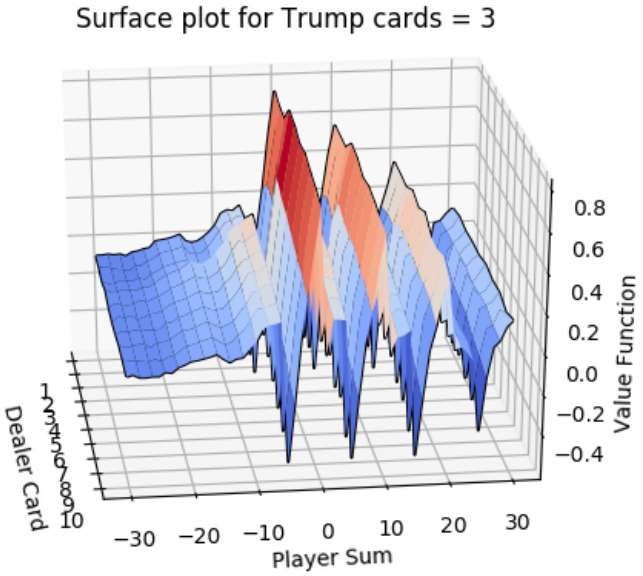
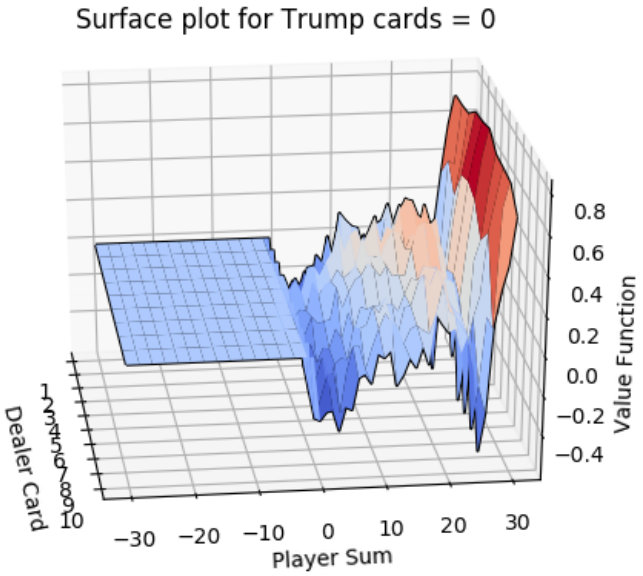
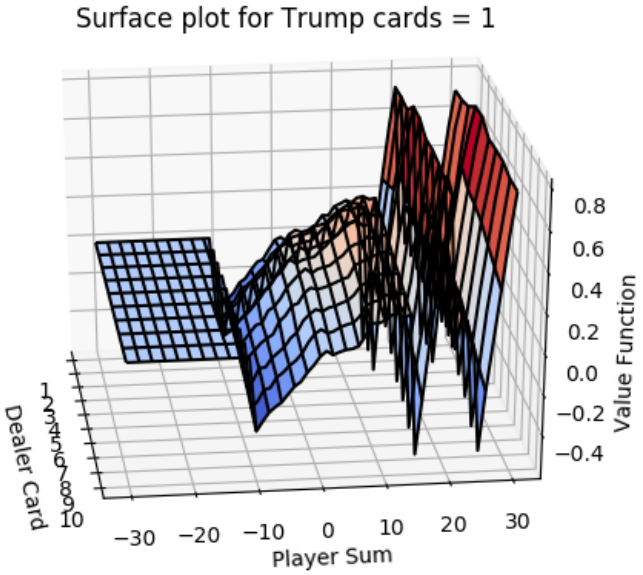
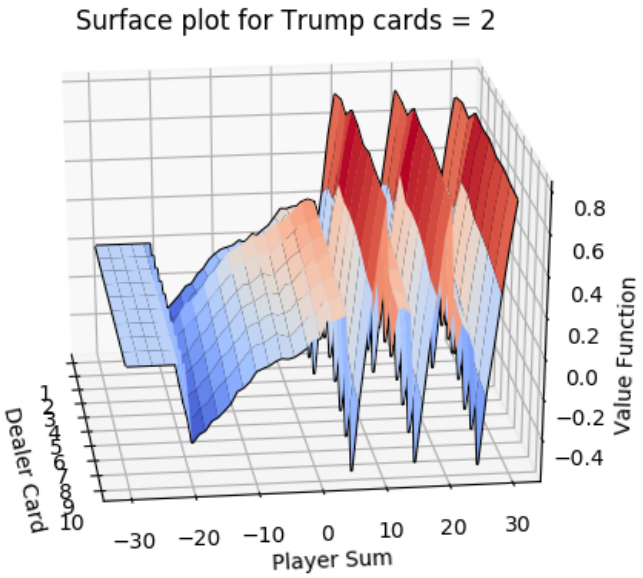


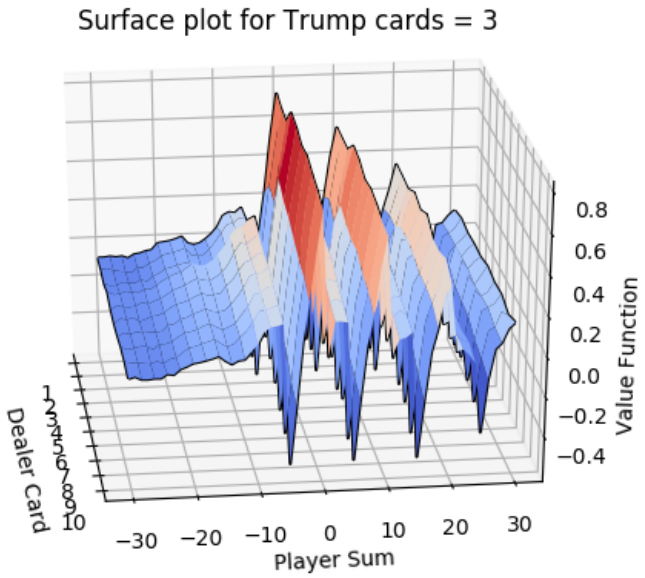
Figure 2: 100 Runs

2.3.2 Variation with number of trump cards



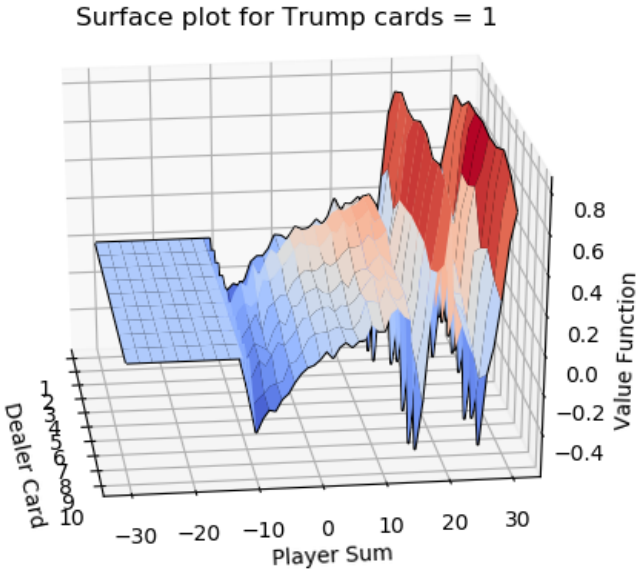


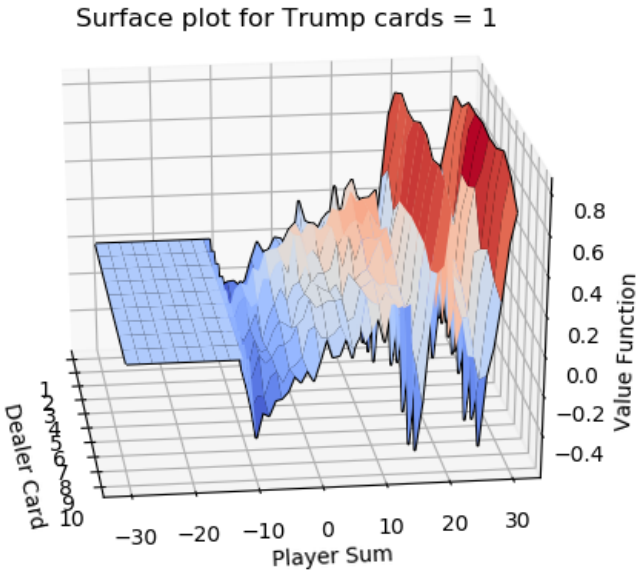

Surface plot for Trump cards = 2
Value Function
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
Player Sum
−30
−20
−10
0
10
20
30
Dealer Card
10

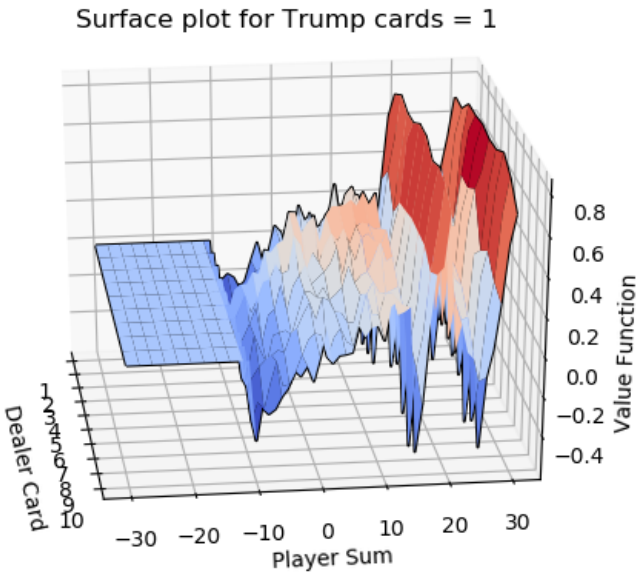


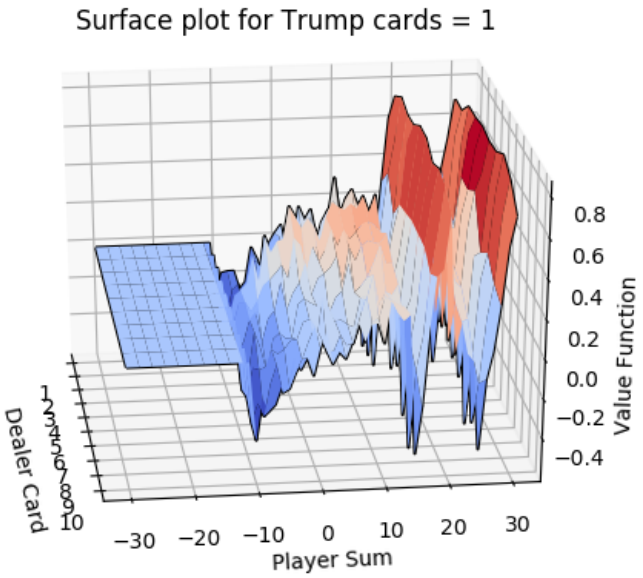
Here we observe that as the number of trump cards increase, the number of states with high values increases. Moreover, since the special cards offer a flexibility of 10 in the sum, we see that the values recur for states with a difference of 10 in their raw sums. Although, this only happens within limits of the soft sum being positive.

2.3.3 Variation with K






Surface plot for Trump cards = 1
Dealer Card
Player Sum
Value Function
10
−30
−20
−10
0
10
20
30
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

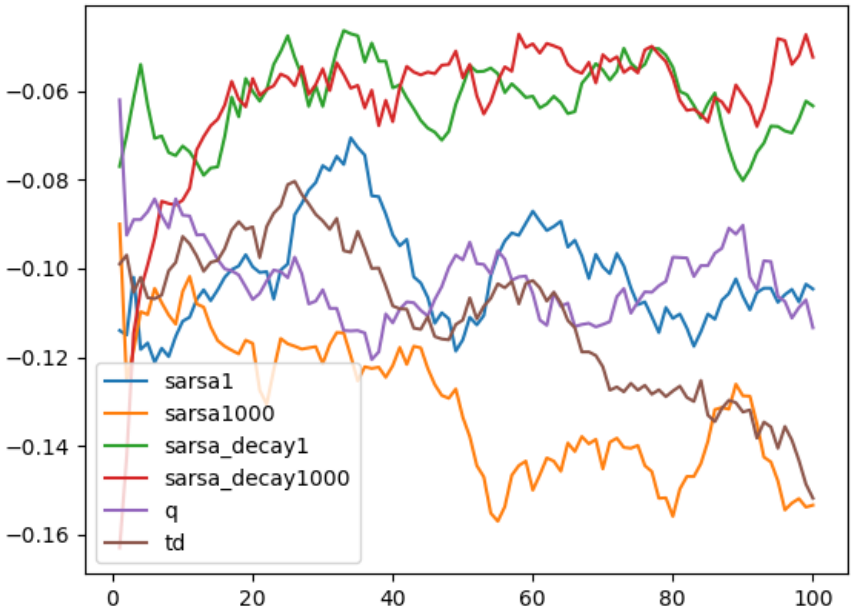


Here, we observe that as k increases, the rewards percolate faster to the states so the value function for higher k are higher. This is because with higher values of k, the look-ahead for each update is higher.

A comprehensive set of all the possible graphs are attached in a separate folder in the submission.

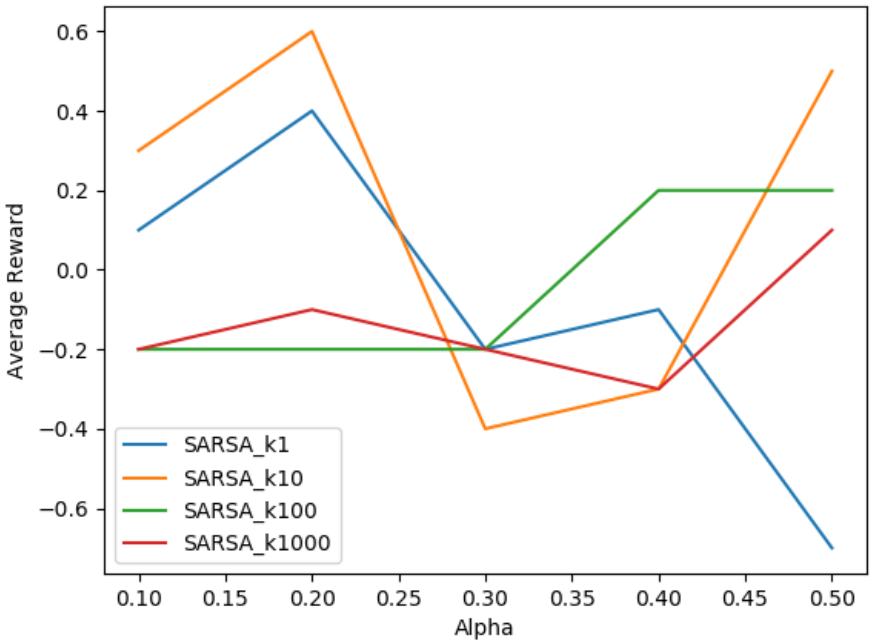
3 Policy Control

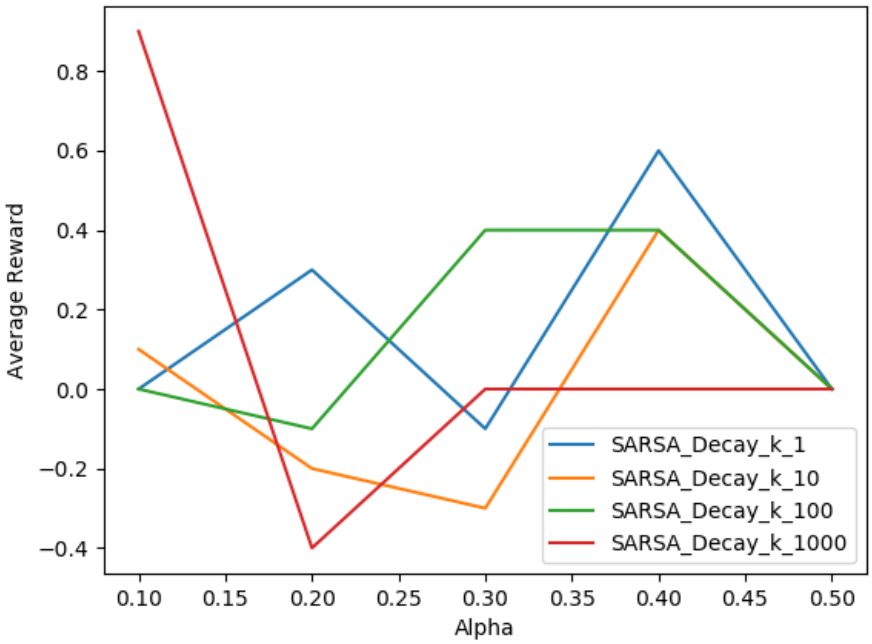
3.1 Fastest Learning

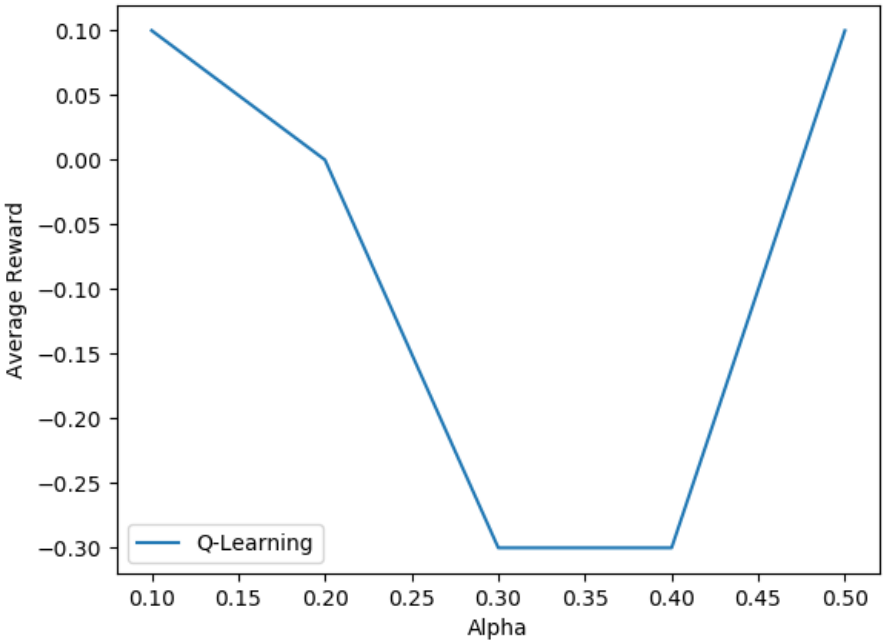


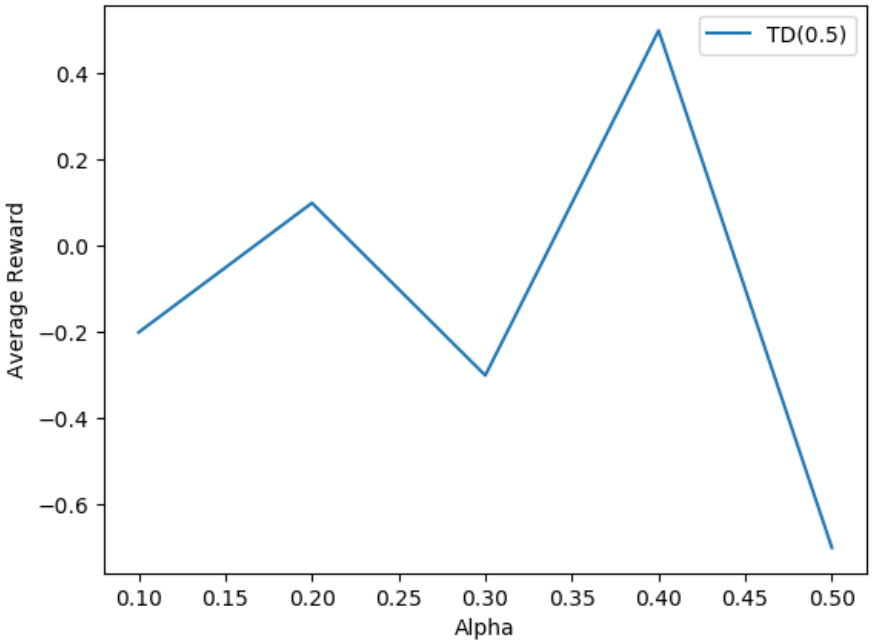
We observe that the SARSA with decay performs the best. Amongst SARSA with different values of k, k = 1000 performs slightly better since it is able to percolate rewards faster. Since the number of episodes is small, this effect becomes prominent. SARSA with constant epsilon performs the worst. This is because since SARSA is an on-policy algorithm, it can only learn the best epsilon-soft policy, not the optimal policy. Also, the value of epsilon used viz. 0.1 was high enough for it to underperform. We also see that Q-learning performs better than SARSA since Q-learning is off-policy and hence can use the greedy policy to update its Q-values. Also, TD(0.5) algorithm performs slightly worse than Q-learning since Q-learning is off-policy learning algorithm.

3.2 Variation with Alpha





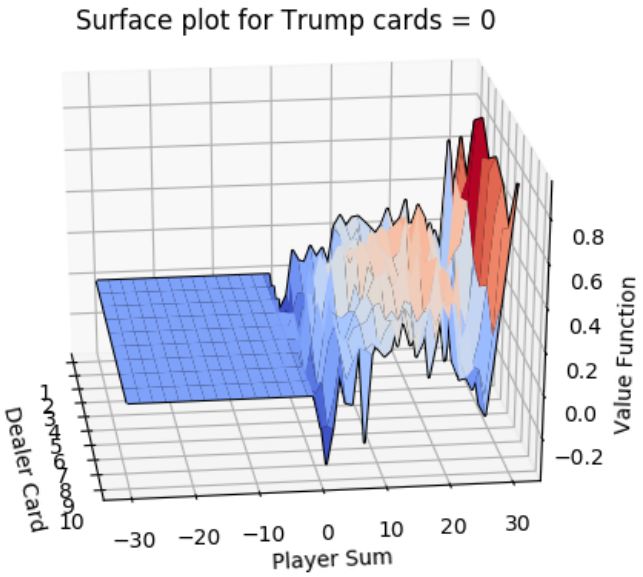

0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
Average Reward
Q-Learning
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
Alpha

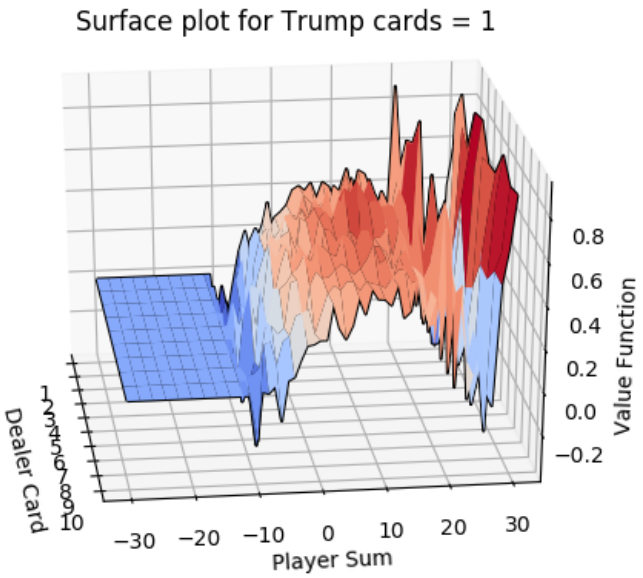


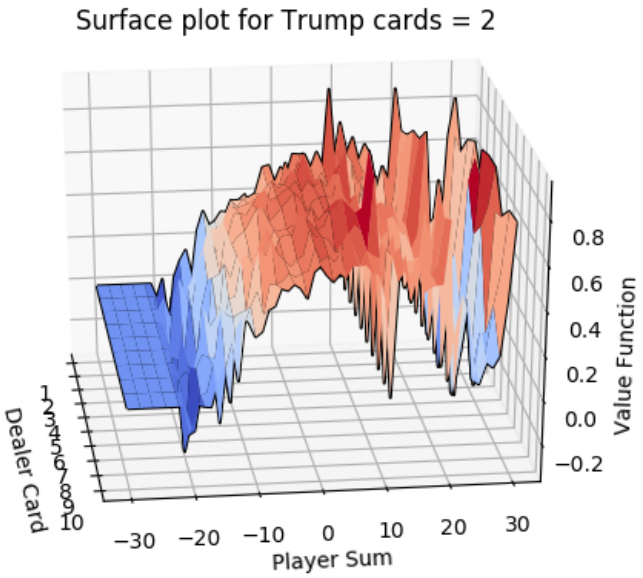
We observe that 10 runs are too few to diminish the effects of stochasticity in since the number of possible states is too high. Over and above that, we see that stochasticity increases as we increase alpha. This is because as alpha increases, the change made in each update becomes larger. Since we are depending more on every new incoming update and less on our prior knowledge, hence the variance increases as alpha increases.

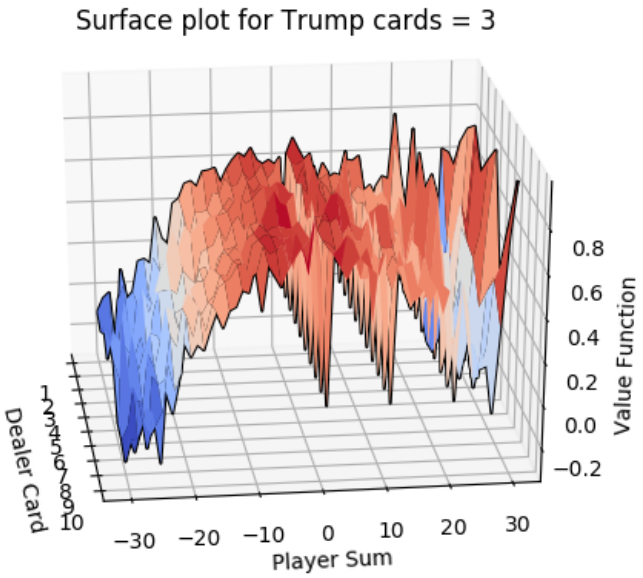
We also notice that at very low values of alpha, the learning becomes very slow as the updates are very small.

3.3 TD(0.5) Policy






Surface plot for Trump cards = 2
Value Function
0.8
0.6
0.4
0.2
0.0
−0.2
Player Sum
−30
−20
−10
0
10
20
30
Dealer Card
10



We observe that since the graphs of the policy learnt by TD(0.5) algorithm are higher than those of dealer's policy, hence the learnt policy is better than the dealer's policy. This is specially evident in the region with negative raw sums, since the dealer's policy had significantly lower expected rewards than the learnt policy in states with negative raw sums.