Reinforcement Learning

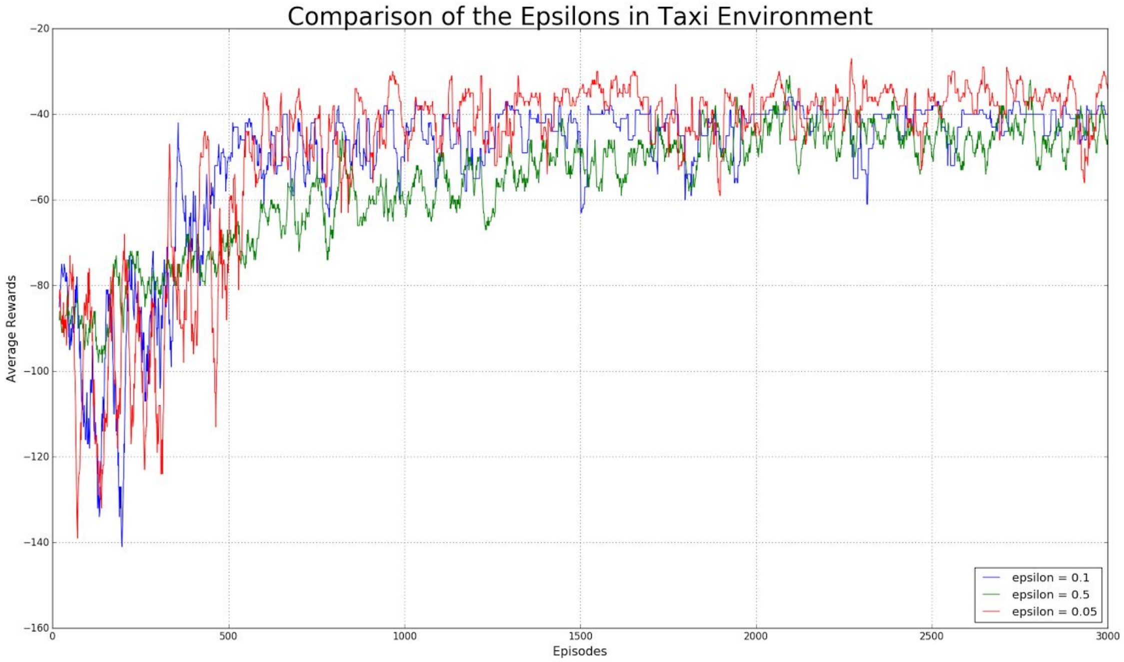
Project 2

Monte Carlo Methods

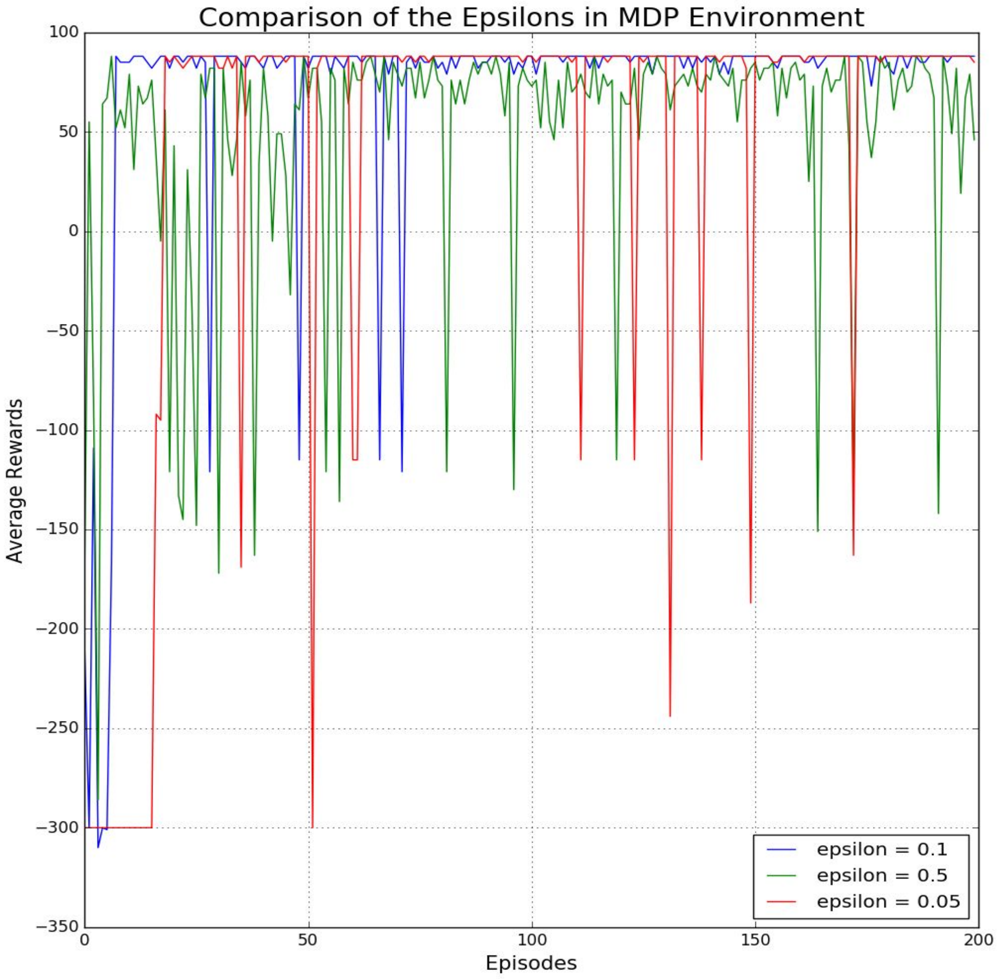
Shivam Goel

WSUID# 11483916

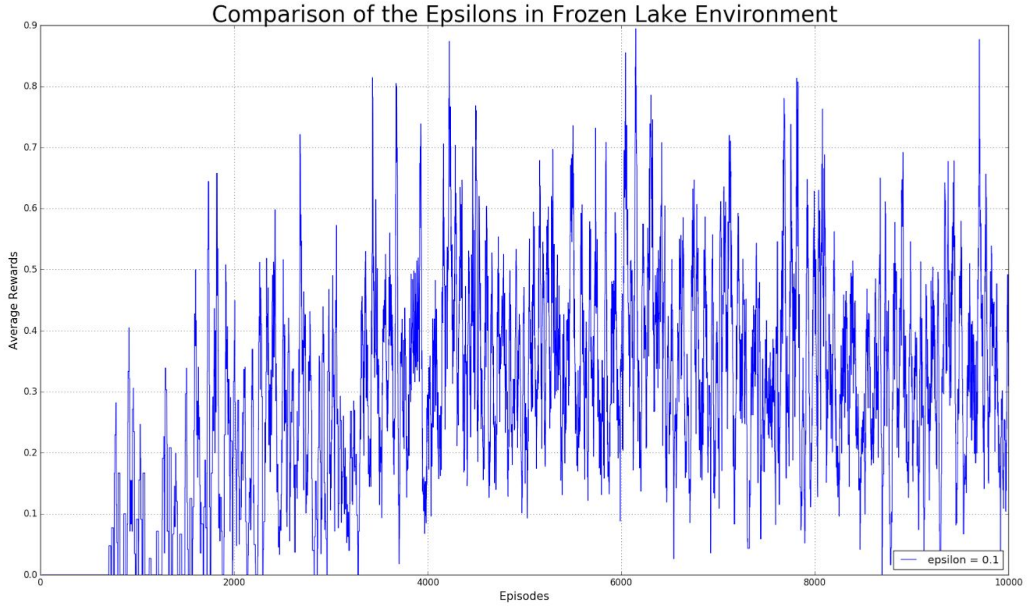
Implemented the On policy Monte Carlo Method for the Assignment.
Compared the algorithm on various environments, viz., MDP (Deterministic), Taxi (Deterministic) and Frozen Lake (Stochastic)
Implementation of the algorithm is in the code and that can be commented out to test in the various environments.
The graphs plotted are discussed below.



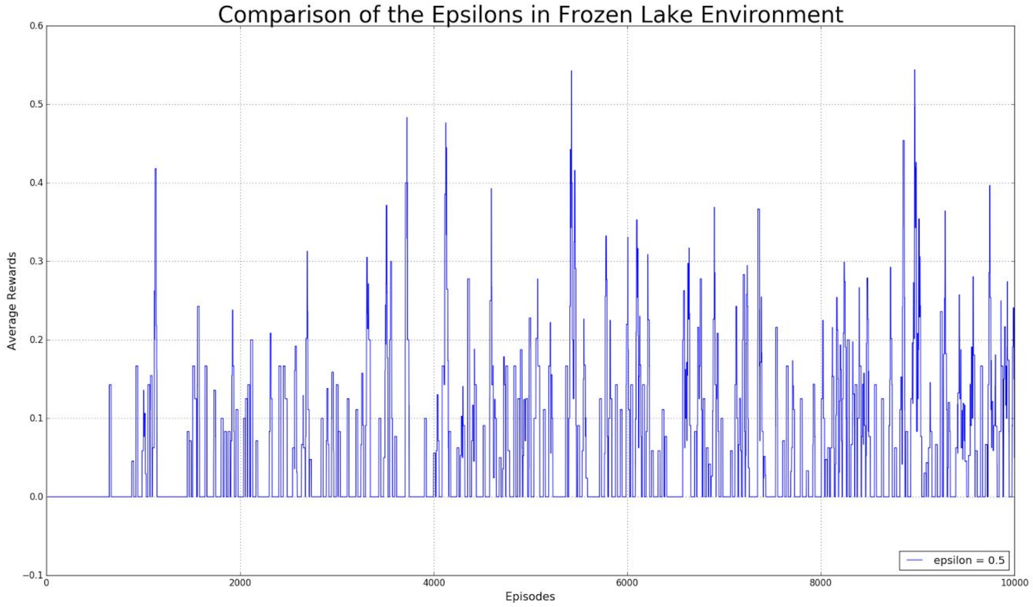
The above graphs shows the performance of the algorithm on the Taxi environment, the Average Rewards over the number of episodes have been plotted in the graph and the average rewards are sampled using the sliding window technique to smooth the graph and hence remove the randomness of the graph, the window size for the current environment is taken to be 15 and the plotting is done for 3000 episodes to know the average rewards. Also, various values of epsilon are compared, 0.1 (blue), 0.5 (green) and 0.05 (red) to compare how well the algorithm performs on changing the randomness.
It is seen that the algorithms performs better as we increase the randomness, hence for lower epsilon values it gives out better average rewards. Overall it was seen that the policy gets stable after 500 episodes.



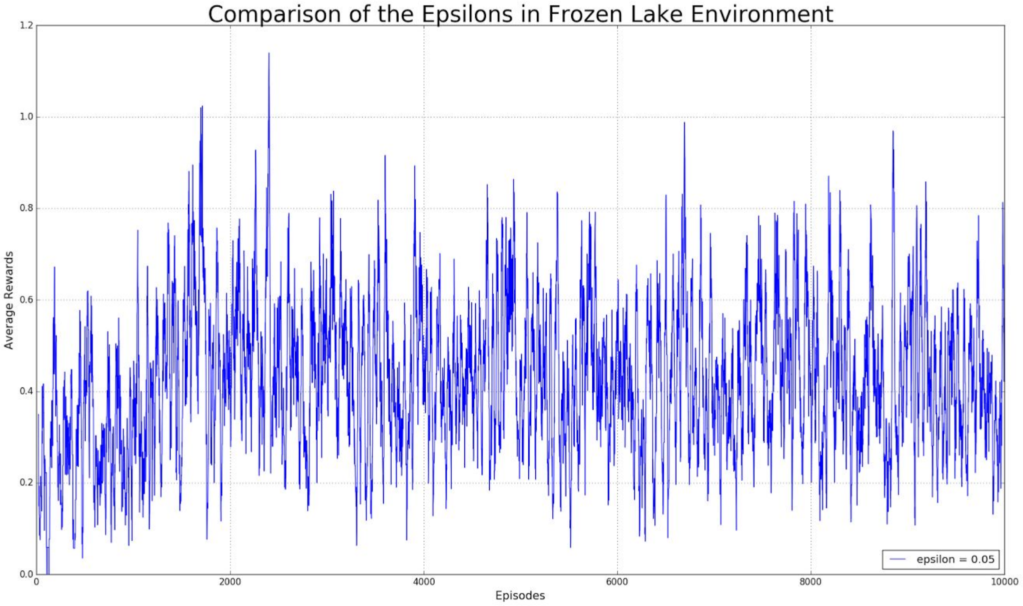
The above graph shows the performance of the algorithm on the MDP environment, the total rewards over each episode is plotted. And compared over the epsilons, 0.1, 0.5 and 0.05 upon comparison it was noticed that the lesser value of the epsilon was more stable than the others. Also the policy became stable after almost 20-25 episodes. Since the environment was deterministic so there was not a significant noise after stabilizing the policy.



The graph shows the performance of the algorithm on the Frozen Lake when epsilon was taken 0.1 Average Rewards were plotted with a sliding window of 20 over 10000 episodes



The graph shows the performance of the algorithm on the Frozen Lake environment when epsilon was taken 0.5, Average rewards were plotted with a sliding window of 20 over 10000 episodes.



The graph shows the performance of the algorithm on the Frozen Lake environment when epsilon was taken 0.05, Average rewards were plotted with a sliding window of 20 over 10000 episodes.

Since, the environment is stochastic so the randomness increased also the graphs showed that the policy starts to yield positive rewards and almost gets stable, can't still stay that it is completely stable because of the randomness of the environment also because of the method - as it explores and learns. But it was observed that the rewards were starting to become positive after 500 episodes (epsilon = 0.1), 800 episodes (epsilon = 0.5) and 100 episodes (epsilon = 0.05), so it clearly shows that lower epsilon performs the best. Also since the environment is stochastic so upon running each time the policy did not converge after same number of episodes.