COL-333 ASSIGNMENT-3

Report

-Chaitanya(2019CS50435)
-Nikhil(2019CS10367)

TAXI DOMAIN PROBLEM:

PART-A

1)

a) State Space:

Here, the State Space considered has 25*26*4 = 2600 States.
Possible positions of taxi = 25
Possible Positions of Passenger = 25(in one of cells)+ 1(In Taxi)
And destinations = 4.

Each state in Space State is of the format [taxi, Pass, Dest]
Where, taxi – from [0,0] to [4,4] totally 25
Pass – from [0,0] to [4,4] and (in taxi).
I denoted [45,45] to be a state which Means in taxi.
Dest – Totally 4 depos.

Action-Space: [‘N’, ’S’, ’W’, ‘E’, ‘PU’, ‘PD’]
Where N,E,W,S are navigation actions , PU- Pick-Up
PD- Put-Down
Reward = [-1, -10, 20] are 3 possible rewards for any action.

b) Simulator: We are given a (state,action) pair. And if action is one of the navigation action, then I randomly chose one of the 4 navigation actions with the probability conditions given.

And thus implemented the code such that the next state will be occurred when this random action is performed on the state.

2)Value Iteration:

a) Here, Given Discount factor = 0.9.

On running Value iteration with Disc_fact = 0.9, and Epsilon = 0.1, the number of iterations = 28.

b)Connection Between Disc_fact and number of iterations:

Here, we have to find number of iterations for all the given Disc_fact values.

For , Disc_fact = 0.01 ---> iterations = 4

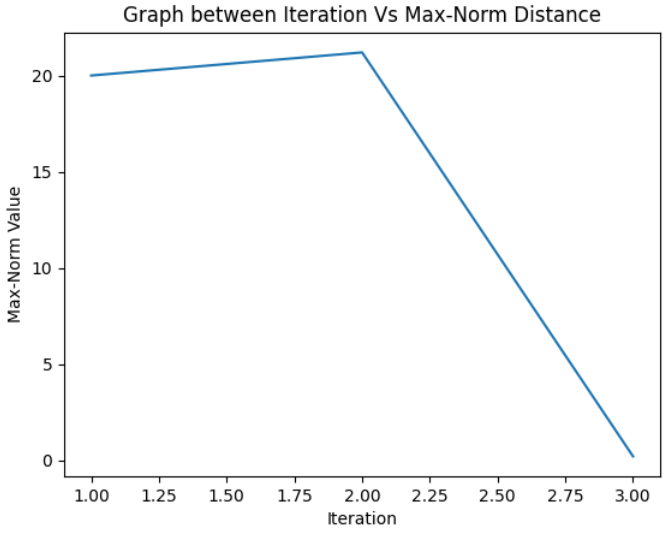
Disc_fact = 0.1 ---> iterations = 5

Disc_fact = 0.5. ---> iterations = 10

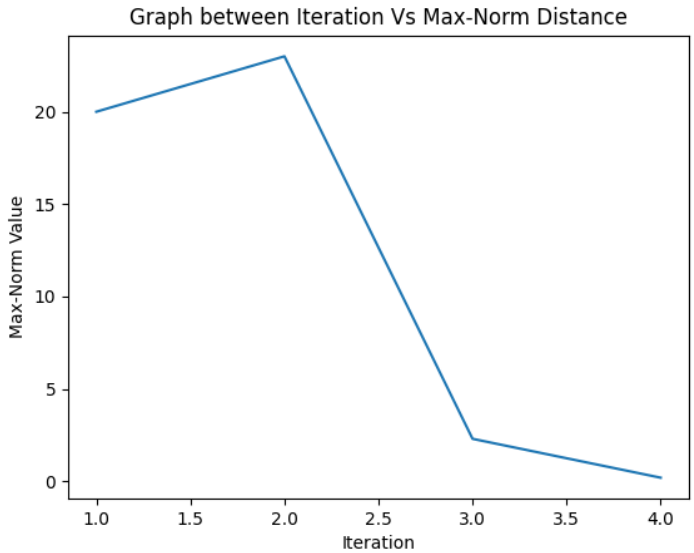
Disc_fact = 0.8. ---> iterations = 21

Disc_fact = 0.99. ---> iterations = 35

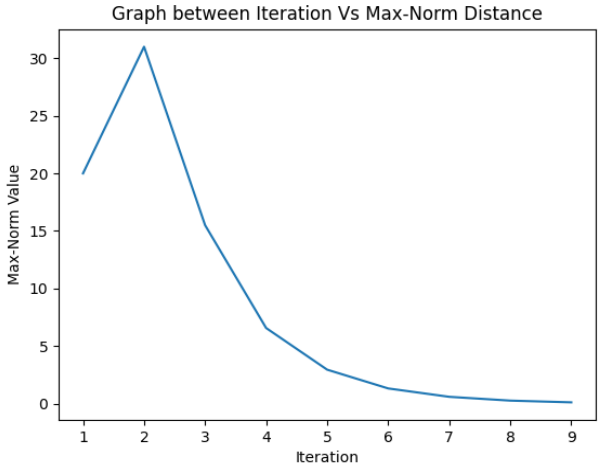
Disc_fact = 0.01



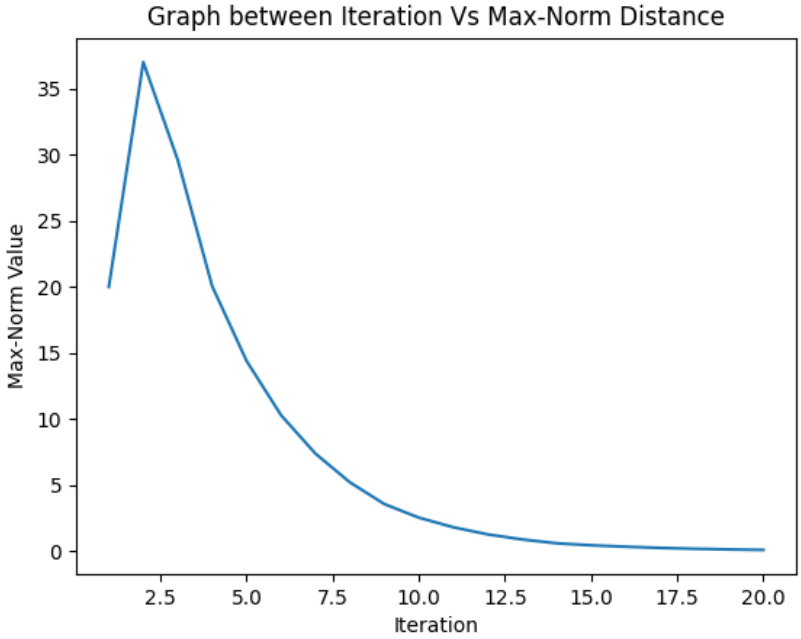
Disc_fact = 0.1



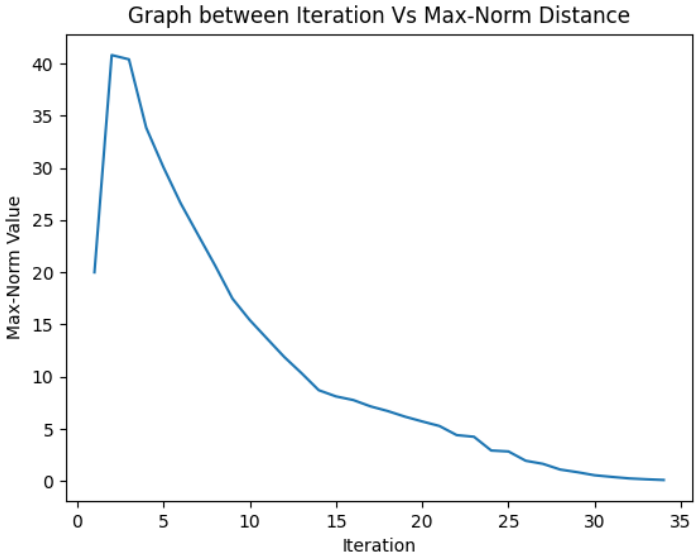
Disc_fact = 0.5.



Disc_fact = 0.8



Disc_fact = 0.99

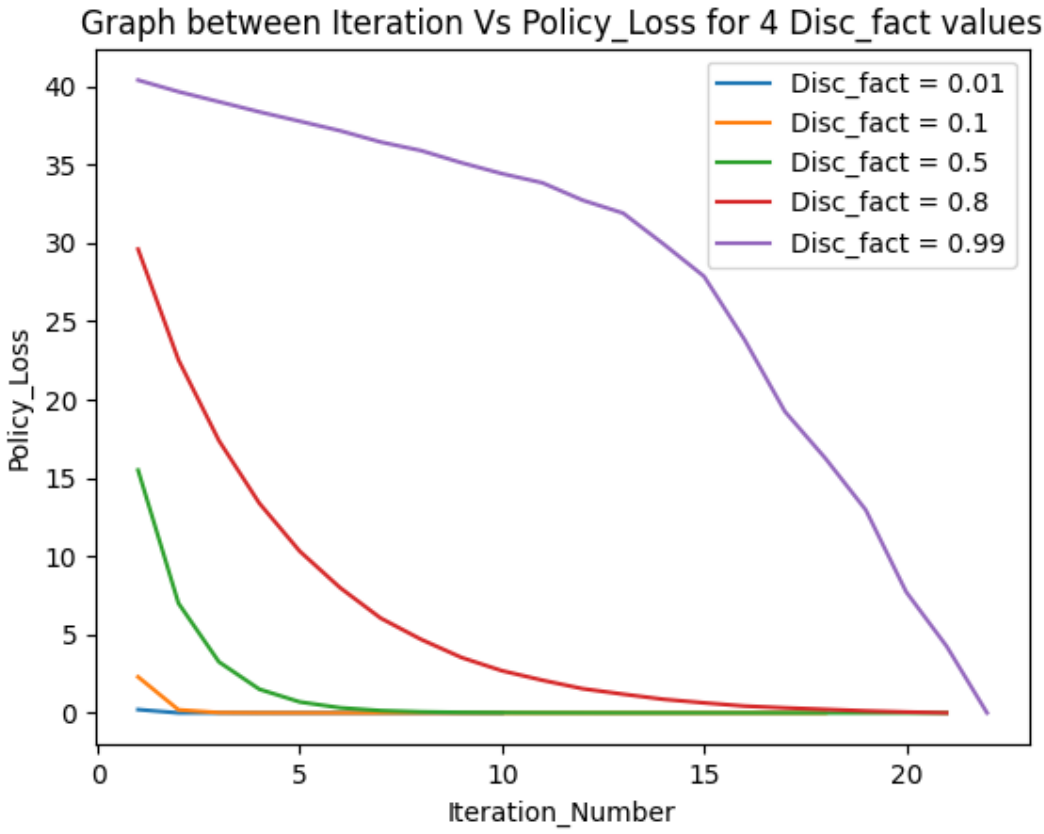


c) Disc_fact = 0.1:
When we observe the first 20 states, we can see that at any state, the algorithm don't consider about the future rewards. It only considers about the present reward. So, the state we get depends mostly on the present reward.

And so, we cant guarantee whether the policy we get is optimal or not.

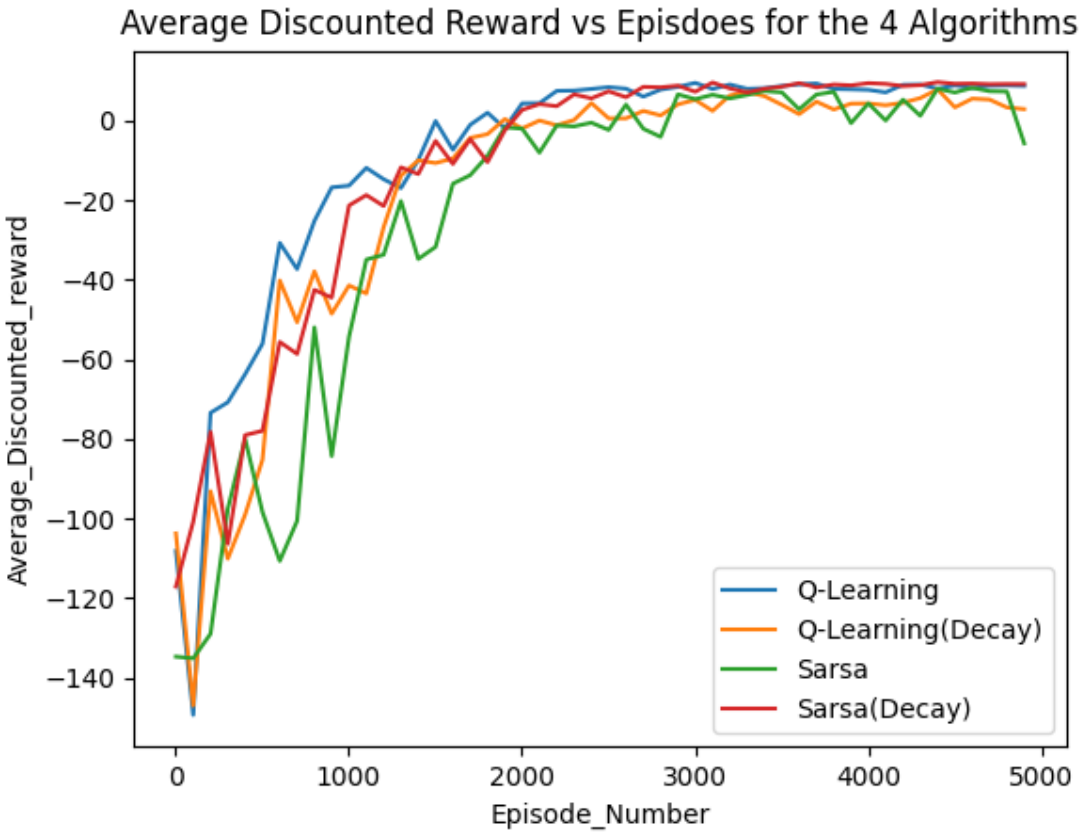
Disc_fact = 0.99
Here, at each evaluation, our algorithm will check about the future reward as well. And so, it will give accurate states, almost right from the beginning.

3)<u>Policy Iteration:</u>
We should implement the policy iteration in both iterative Method as well as linear algebra method.



PART-B – INCORPARATING LEARNING:

2)While executing all the above 4 algorithms for 5000 episodes With alpha(learning rate) = 0.25 and Discount_factor = 0.99 we Get the below graph.

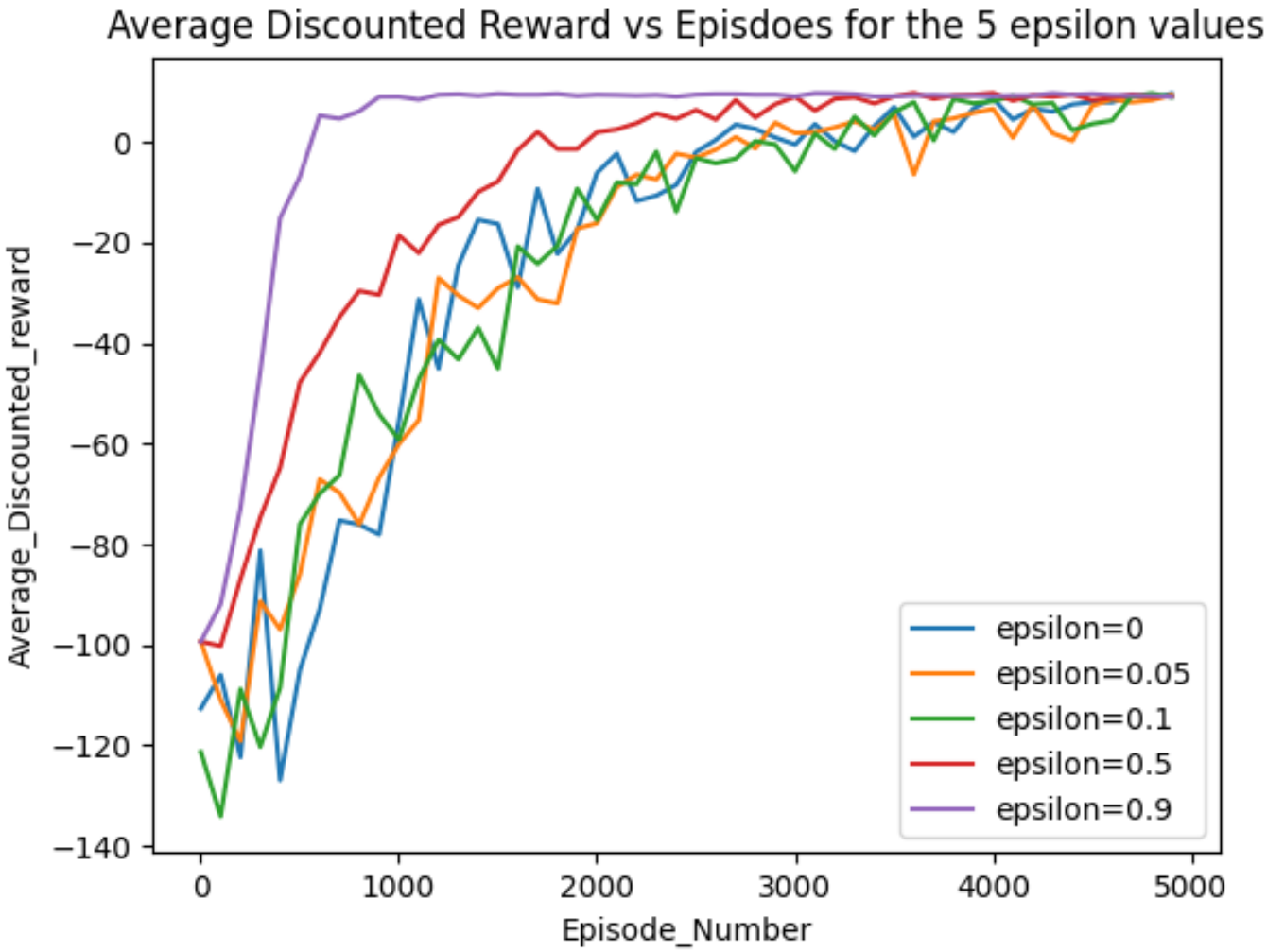


From the above algorithm we can say that, almost all of them Converge at the end. But out of all Q-learning converge quickly.

3) From the above part, I got a high value in case of Q-learning And so I selected that learning algorithm. We should perform it on 5 instances. The values I got are -99.3429, -99.98, -0.988, -1.6778, 0.977678

4) Here, the main is to check the convergence of Q-learning On different alpha values as well as epsilon values.

In first case we check the convergence of Q-learning by taking Alpha = 0.25(fixed) and ε(epsilon) = {0,0.01,0.1,0.5,0.9}



Here, we can see that when epsilon =0.9, the algorithm will converge

Now, we check the convergence of q-learning for different alpha values and a fixed epsilon value.
Alpha = {0.1,0.2,0.3,0.4,0.5} and epsilon = 0.1