TP1 for Reinforcement learning

Leman FENG
Email: flm8620@gmail.com
Website: lemanfeng.com

November 10, 2017

1 Dynamic Programming

1.1 Implement the discrete MDP model

see code in ex1.m

1.2 Value iteration

v^* is calculated by direct computation of policy iteration using matrix inverse.
$||v^{k+1} - v^k||_\infty$ is plotted in Figure 1
$||v^{k+1} - v^*||_\infty$ is plotted in Figure 2

1.3 Policy iteration

$||v^{k+1} - v^k||_\infty$ is plotted in Figure 3
$||v^{k+1} - v^*||_\infty$ is plotted in Figure 4

We can see that this policy has changed twice after each value iteration. I use the iterative policy evaluation to get a ϵ-approximation of V, using the same ϵ as VI in the above question for fairness. PI takes more steps than VI in this case.

But if the problem is not very complex, by using the direct computation, PI can be way faster than VI

2 A Review of RL Agent/Environment Interaction

Question 4&5 is implemented in main_tp1_ex2.m

J_n and J^* is plotted in Figure 5.

For question 5, we need compare the learning curve of $||v^* - v^{\pi_n}||_\infty$ for different choice of ϵ. But for each evaluation, we only have one learning curve. To show the performance, we should try multiple times for each ϵ.

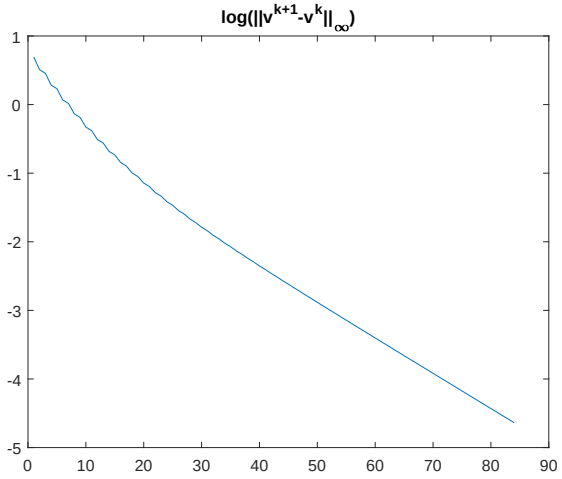


Figure 1:

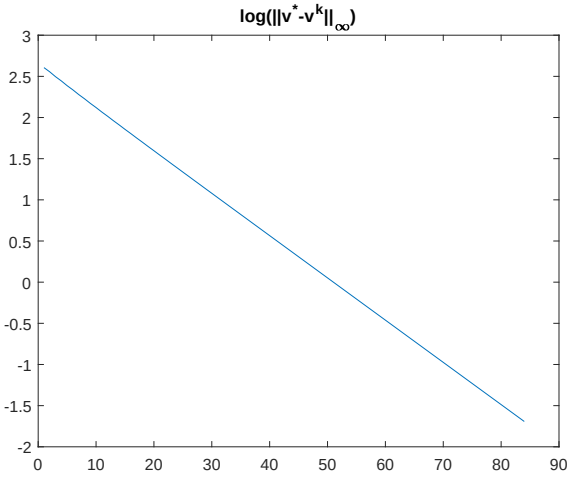


Figure 2:

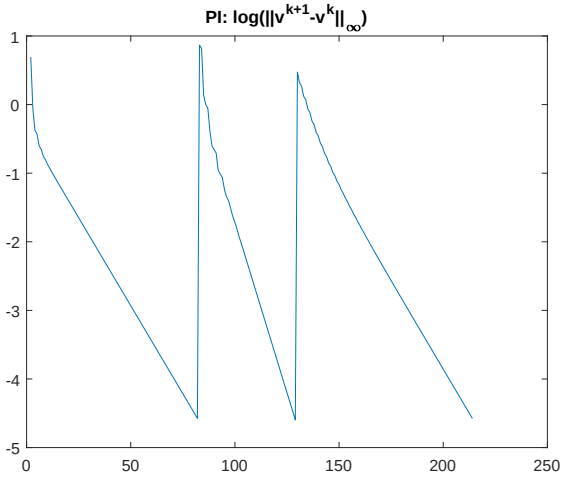


Figure 3:

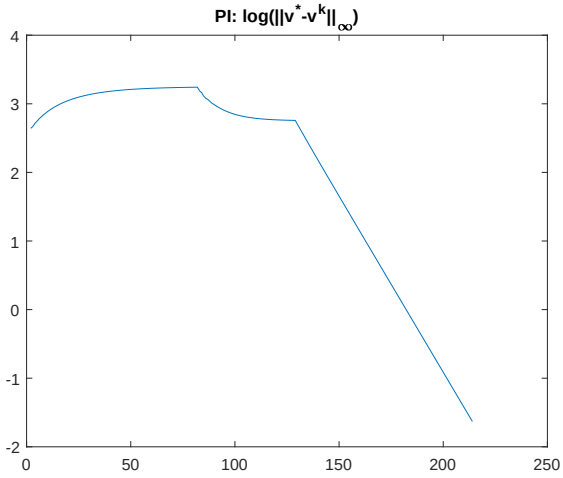


Figure 4:

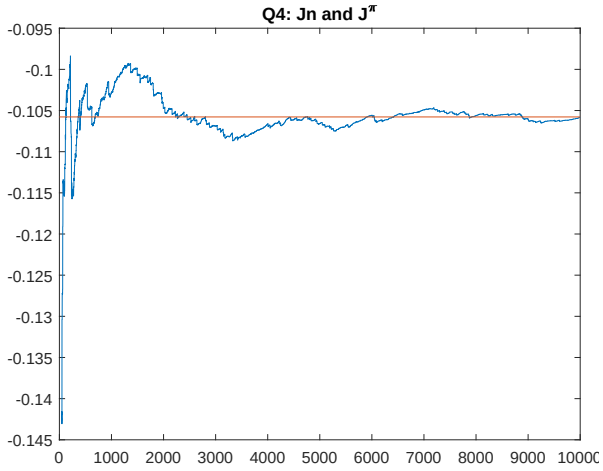
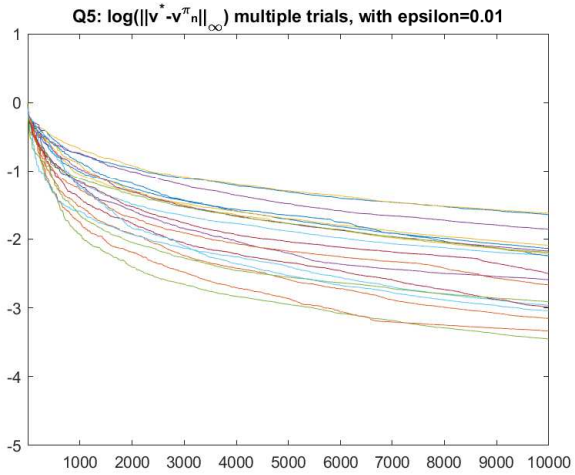


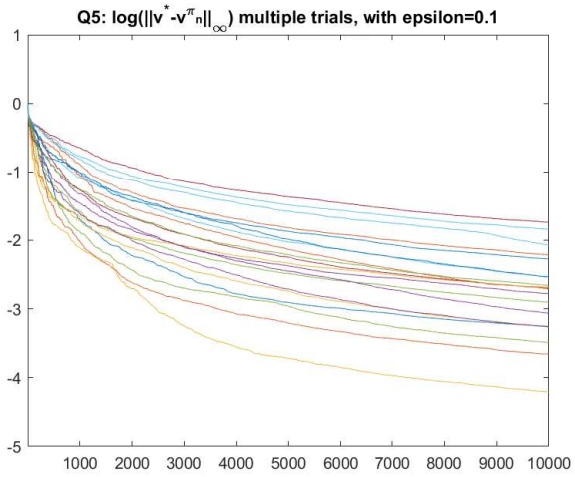
Figure 5:

In Figure 6, we see the choice of epsilon has less influence on the learning curve. But for the cumulated reward curve, a large epsilon brings less reward, which is logical that random actions will lead to more regrets of reward.

2.1 Is the optimal policy of an MDP affected by the change of the initial distribution?

No. according to bellman's principle of optimality:" An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. "


Q5: log(||v* -vπn||∞) multiple trials, with epsilon=0.01
1
0
-1
-2
-3
-4
-5
1000 2000 3000 4000 5000 6000 7000 8000 9000 10000


Q5: log(||v* -vπn||∞) multiple trials, with epsilon=0.1
1
0
-1
-2
-3
-4
-5
1000 2000 3000 4000 5000 6000 7000 8000 9000 10000

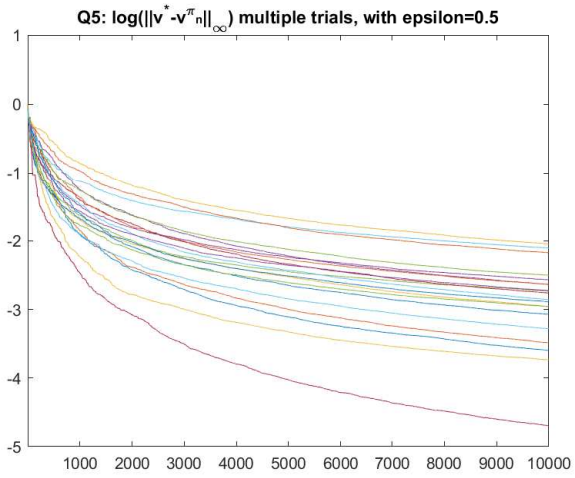
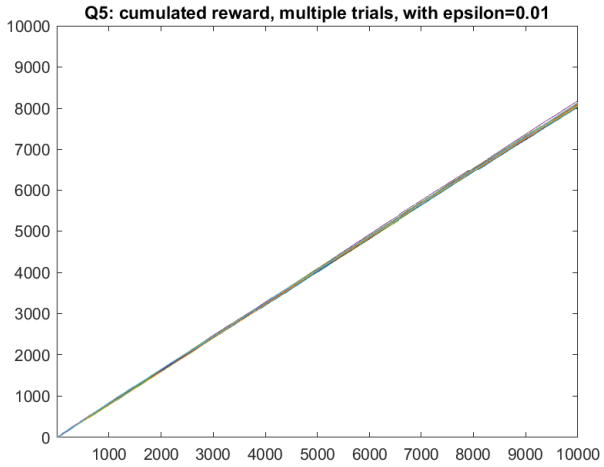
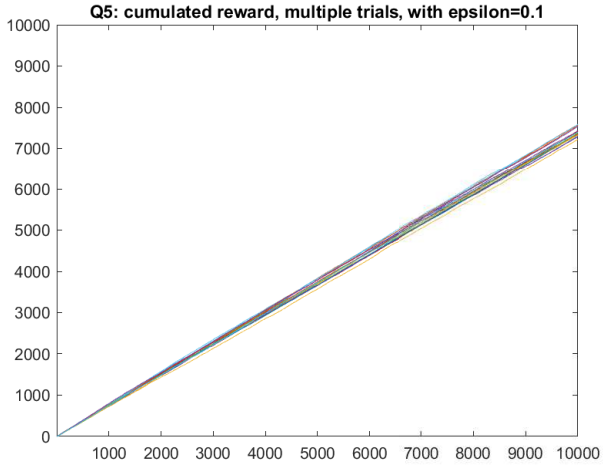

Q5: log(||v* -vπn||∞) multiple trials, with epsilon=0.5
1
0
-1
-2
-3
-4
-5
1000 2000 3000 4000 5000 6000 7000 8000 9000 10000

Figure 6:


Q5: cumulated reward, multiple trials, with epsilon=0.01
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
1000 2000 3000 4000 5000 6000 7000 8000 9000 10000


Q5: cumulated reward, multiple trials, with epsilon=0.1
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
1000 2000 3000 4000 5000 6000 7000 8000 9000 10000

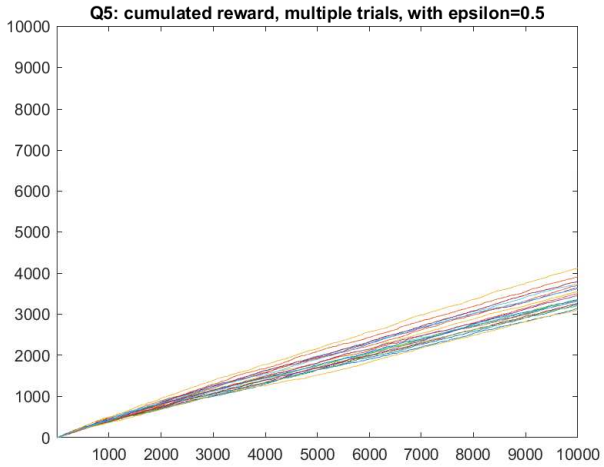

Q5: cumulated reward, multiple trials, with epsilon=0.5
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
1000 2000 3000 4000 5000 6000 7000 8000 9000 10000

Figure 7: