Reinforcement Learning Homework 1 : Dynamic Programming and Reinforcement Learning

Hugo Cisneros

1 Dynamic programming

1. Optimal policy is [1 1 2], it corresponds to jumping as fast as possible to the state 2 where a reward of 9/10 can reliably be obtained with action 2.

2. $v^* = [15.28615506, 16.44338493, 17.89499783]$

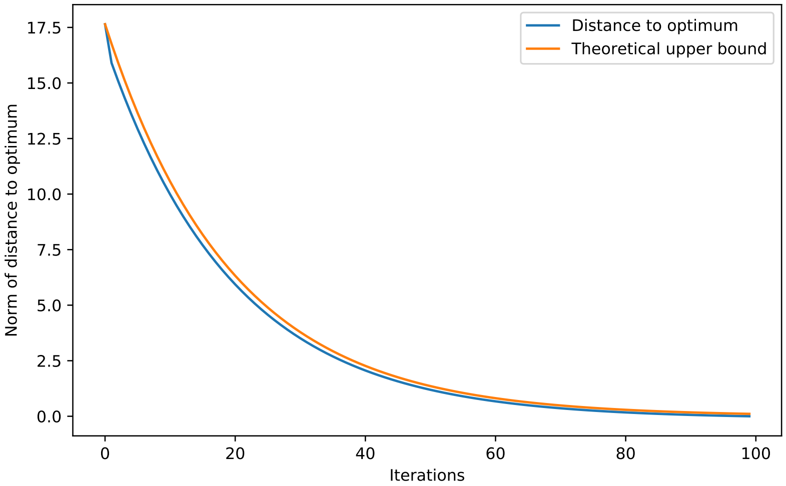


Figure 1: Distance to optimum as a function of number of iterations for value iteration

3. Policy iteration converges much faster than value iteration. It only takes 3 steps to reach the optimal policy. However, each of those steps implies inverting a matrix of size the number of states. This is achieved in $O(n^3)$ time, which isn't significative for such a small problem but becomes prohibitively big for large problems. In that case, value iteration should be preferred.

2 Reinforcement Learning

1.

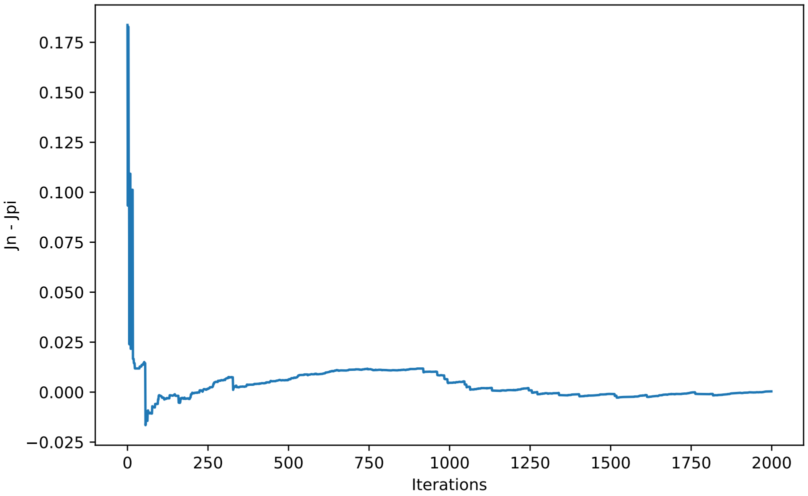
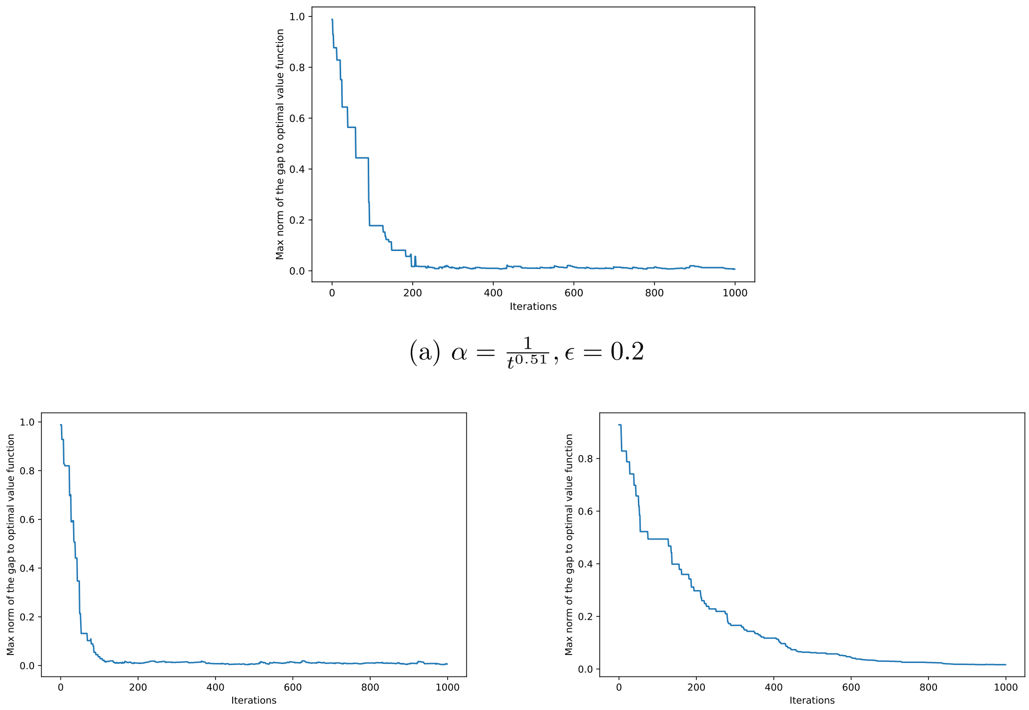


Figure 2: Gap $J_n - J_\pi$ as a function of number of iterations

2. The learning rate affects how fast the algorithm converges to a solution (see Figure 3 for illustrations). The closer α is to decaying like the inverse square root, the faster the algorithm converges. The exploration parameters is also very important : for a low ϵ, the algorithm keeps taking the greedy solution and innovates less often, this can be seen on the constant parts of the curve on Figure 3(a). Overall, high exploration parameter and learning rate seem to yield the best results for this problem.

3. The optimal policy doesn't depend on μ_0 since the actions taken depend only on the current state which is not affected by the initial state



(a) $\alpha = \frac{1}{t^{0.51}}, \epsilon = 0.2$

(b) $\alpha = \frac{1}{t^{0.51}}, \epsilon = 0.9$

(c) $\alpha = \frac{1}{t^{0.8}}, \epsilon = 0.5$

Figure 3: Comparison of learning rates and exploration parameters for the Q-learning algorithm

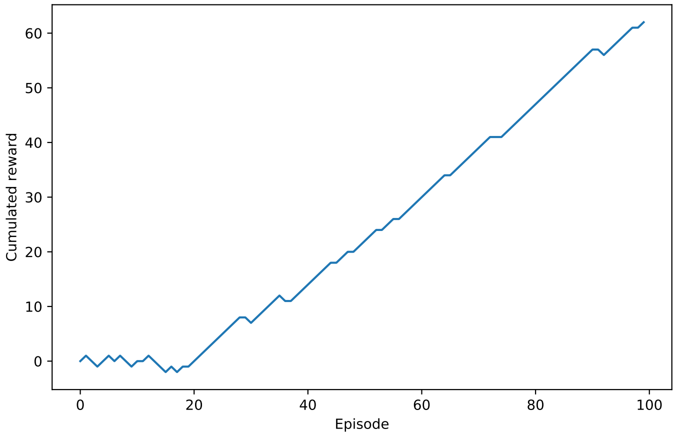


Figure 4: Evolution of the cumulated reward over the 100 first episodes

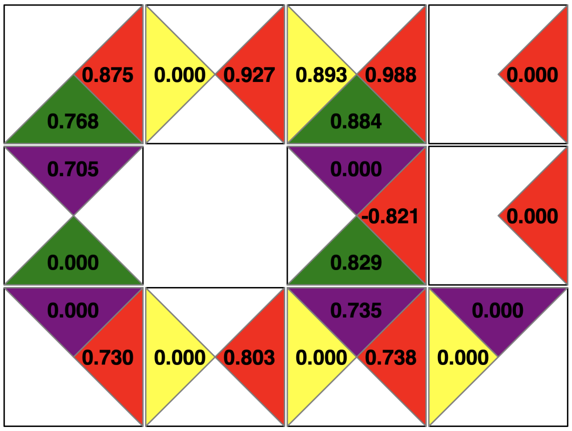


Figure 5: Visualization of the final Q function after Q-learning