Dynamic Programming and Reinforcement Learning

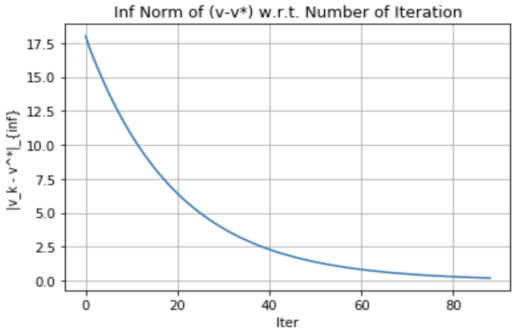
MVA - Reinforcement Learning

Tong ZHAO (tong.zhao@eleves.enpc.fr)

1 Dynamic Programming

Q1 It is simple to guess the optimal policy since the action a_2 for the state s_2 has a reward 0.9 with probability 1, while s_0 has a small reward 0.05 with probability 1 and s_1 has no stable reward. Hence we deduce that the optimal policy $\pi^\star = [a_1, a_1, a_2]$.

Q2 We first implement the policy evaluation algorithm to compute $v^\star$. And then we run a value iteration to identify a 0.01-optimal policy. The following figure shows the the evoluation of $\|v^k - v^\star\|$ with respect to the number of the iteration k.



Q3 We implement exact policy iteration with initial policy $\pi_0 = [a_0, a_0, a_0]$. We observe that the PI converges at the third iteration, which is much faster than the one of VI (It converges at the 89th iteration). In terms of the calculation time, PI takes around 0.001s while VI takes around 0.008s. Here we compare the two approaches from following aspects.

Value Evaluation

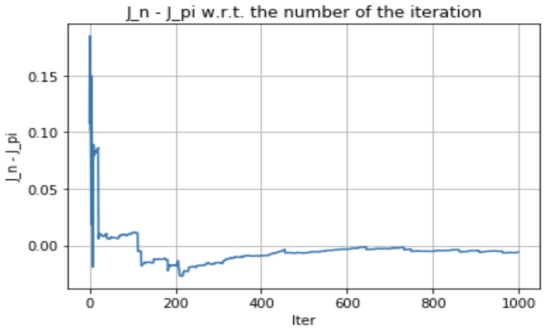
- VI applies the optimal Bellman operator to the value function in a recursive manner, so that it converges to the optimal value. It needs only a single loop and each iteration is easy to solve.
- The speed of convergence is quite low since we need to find the fixed point.

Policy Evaluation

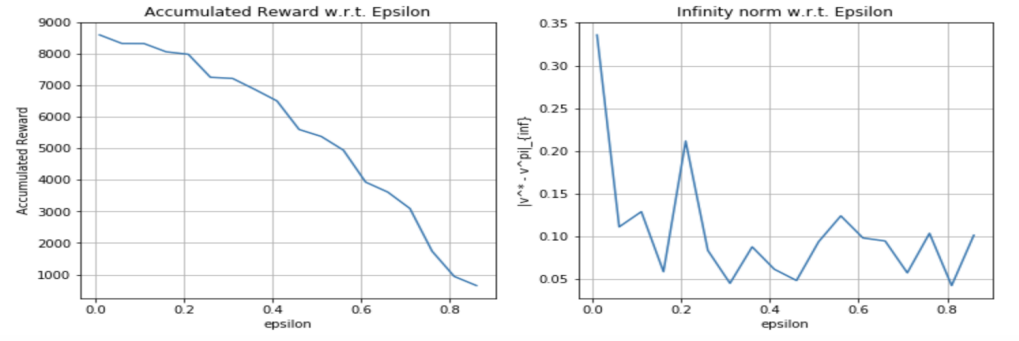
- PI needs to solve a linear system during each iteration which is very costly, especially for large state-action sets.
- PI converges faster since it looks all the state-action pairs and then update the policy.

2 Reinforcement Learning

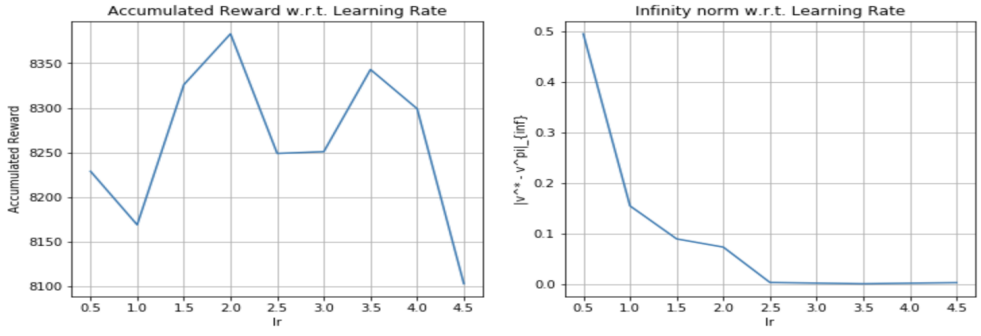
Q4 We implement value estimation using Monte-Carlo and use the method to evaluate the given deterministic policy. We plot $J_n - J^\pi$ as a function of n where $n \in \{1, \ldots, 1000\}$.



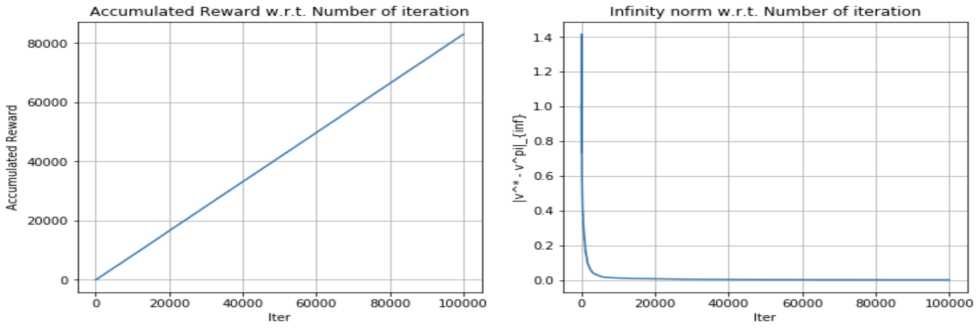
Q5 We use an ϵ-greedy exploration policy which each episode. In order to choose a good ϵ, we sample 18 λs uniformly from the interval $[0.01, 0.9]$ and evaluate the results based on two metrics, namely the $\|v^\star - v^{\pi_n}\|_\infty$ and the cumulated reward over the episode.



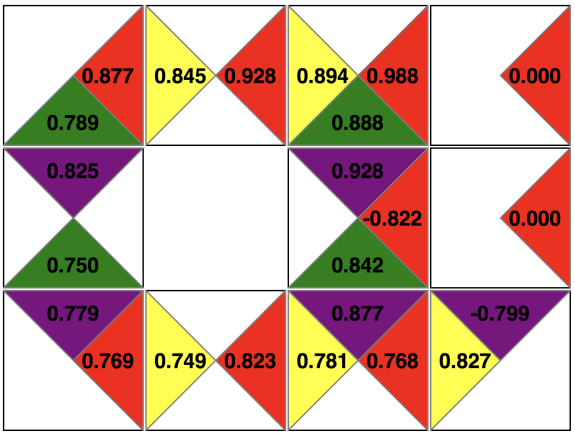
The chosen learning rate strategy $\alpha_t(x_i, a_i) = \dfrac{a}{N(x_i, a_i)}$. We sample uniformly 9 learning rates a from the interval $[0.5, 5]$ and evaluate the final result.



By considering the above experiments, we choose $\epsilon = 0.1$, $a = 2$, and $K = 100000$ and re-calculate the Q matrix. The following figure visualize the two metrics.



The learnt policy is:



Q6 The optimal policy of an MDP should not be affected by the change of the initial distribution μ_0, since the optimal policy is defined by a function that selects an action for every possible state and actions in different states are independant.