Master M2 MVA 2018/2019
Reinforcement Learning - TP1

Souhaib ATTAIKI

November 10, 2018

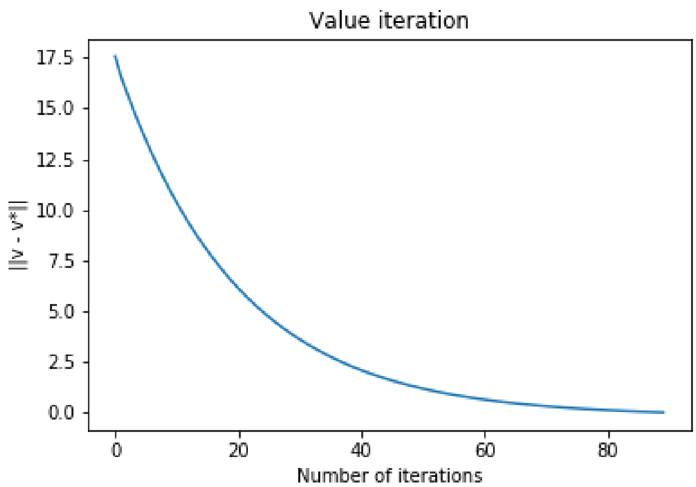
1 Dynamic Programming

1.1 Q1 : Optimal policy

The MPD is implemented in *exo1.py*.
The guessed optimal policy is $\pi^* = [a_1, a_1, a_2]$.

1.2 Q2 : Implementation of value iteration

Value iteration is implemented in *tp1_exo12.py*. In the following figure is plotted $\| v^k - v^* \|$ as a function of iterations.



The optimal policy returned by the value iteration algorithm is $\pi^* = [a_1, a_1, a_2]$ which is conform to our guess in **Q1**.
By implementing the policy evaluation, we found that the $v^* = [15.39, 16.54, 18.]$

1.3 Q3 : Exact policy iteration

By implementing the exact policy iteration algorithm, we found that $\pi^* = [a_1, a_1, a_2]$.

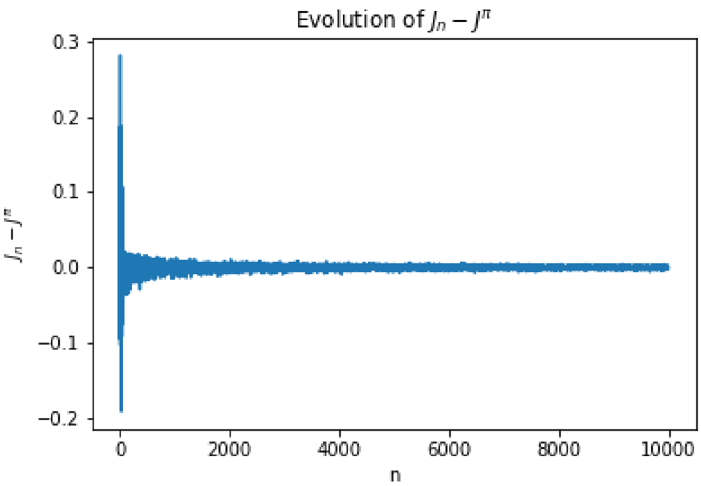
It can be seen that **PI** converges faster than **VI** in terms of iterations (4 versus 89), however, the latter's iterations are not itchy in terms of calculation unlike **PI**.

2 Reinforcement Learning

2.1 Q4 : Policy evaluation

The code for computing J_n is provided in the notebook *visualisation.ipynb* or the generated pdf *visualtions.pdf.*

The plot of $J_n - J^\pi$ is shown in the next figure.



2.2 Q5 : Policy optimization

See *visualisation.ipynb* or the generated pdf *visualtions.pdf.*

2.3 Q6 : Effect of μ_0

No, the optimal policy is not affected by by the distribution of μ_0. In fact, if we often start with states that give a good reward, then the decisions that will choose these states will be privileged, and if not, if we often start with states that give a bad reward, then the decisions that will choose these states will not be privileged.