Reinforcement Learning TP1 : Dynamic Programming and Reinforcement Learning

Plassier Vincent

1 Dynamic Programming

1.1

We can guess that $[1, 1, 2]$ is the optimal policy.

1.2

Thanks to the Puterman's theorem with $\varepsilon = \frac{1-\gamma}{200\gamma}$, once can deduce that :

$$\|v^{k+1} - v^k\|_\infty \Longrightarrow \|v^{\pi_{k+1}} - v^\star\|_\infty \leq 0.01$$

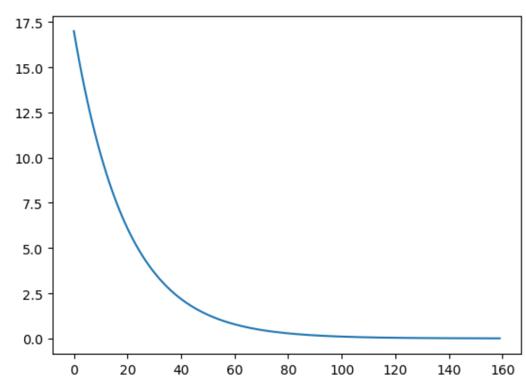


FIGURE 1: $\|v^{\pi_k} - v^\star\|_\infty$ in function of k

1.3

Now, we display the iterations obtained by the algorithm Policy iteration (PI) :

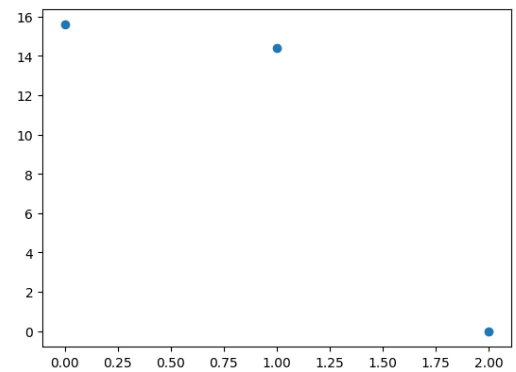


FIGURE 2: $\|v^{\pi_k} - v^{\star}\|_{\infty}$ in function of k

We find that the Political Iteration (PI) algorithm converges in only 3 iterations whereas the algorithm Value iteration (VI) need more run. However, the PI algorithm solves a linear system. So PI has a complexity about $\mathcal{O}(n\_states^3)$, whereas VI has a complexity about $\mathcal{O}(n\_action^2 \times n\_states)$. Thus, it may be advantageous to choose the method based on the ratio n_action/n_states.

2 Reinforcement Learning

2.1

In this question, we consider the quantity $\sum_{s \in S} \mu_0(s) \left[V_n(s) - V^{\pi}(s)\right]$ above n. We plot the first iterations :

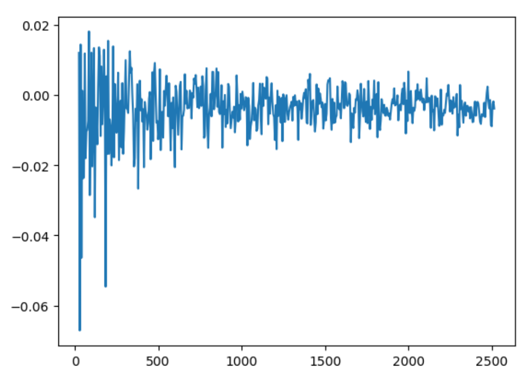


FIGURE 3: $sum_{s \in S} \mu_0(s) \left[V_n(s) - V^{\pi}(s)\right]$ in function of n

The error this decay to 0 when n increase.

2.2

From now on, we are interested by the study of a Q-learning algorithm. In the algortihme described in the subject, we can change the exploration-exploitation's rapport by playing on the parameters ε and $\alpha_i(x, a)$. The first figure is obtained by considering $\varepsilon_t = (0.99)^t$. Thus, as the number of iterations increase, our learning rate decreases. Therefore, the algorithm will more and more privilege exploitation.

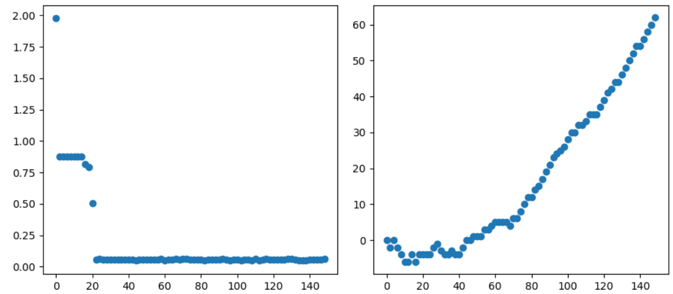
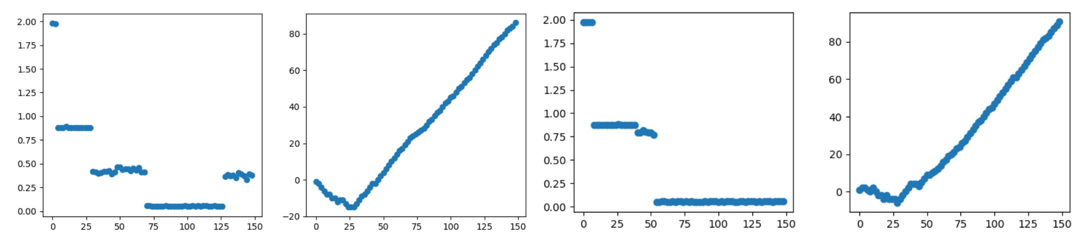


FIGURE 4: Error and reward evolutions in function of the epoch

We also represent the evolution of $\|v^\star - v_{\pi_n}\|_\infty$ during learning as well as the sum of cumulative rewards. Considering this time $\varepsilon \in \{0.01, 0.1\}$. We get :



(a) $\varepsilon = 0.01$ (b) $\varepsilon = 0.1$

FIGURE 5: Representation des approximations obtenues

For the value $\varepsilon = 0.01$, we see that the algorithm still has not converged after 150 iterations. Indeed, there has not been enough exploration.
Remark : in order to determine the optimal hyperparameters, it is possible to use the python fmin function.

2.3

Let's denote $V^\star$ the unique fixed point of Gamma and $pi^\star$ the optimal policy.
According to the course,

$$\pi^\star(x) = \arg\max_a \left[r(x,a) + \sum_y p(y|x,a) V^\star(y) \right]$$

The previous equation is independant of the choice of the initial policy μ_0. Thus, we conclude that $\pi^\star$ is independant of μ_0.

Numerically, we can verify that the choice of another initial distribution does not affect the result. For that, we draw the curve obtained by replacing the initial distribution :

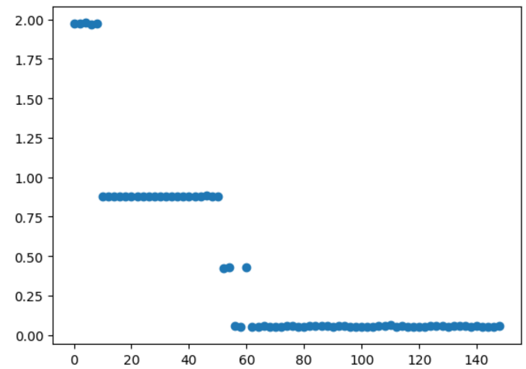


FIGURE 6: Evolution of the error with an other initial distribution

We constate that the curve obtained 6 is very similar to the previous one in 5. We deduce that μ_0 input distribution does not influence the optimal policy.