

CME 241: Assignment 4: Problem 1

Let us determine the optimal deterministic policy with the value iteration algorithm.

We have $V_0 \in \mathbb{R}^2$ and $V_0 = \begin{pmatrix} 10 \\ 1 \end{pmatrix}$

k=1: let us compute $q_1(s_1, a_1)$, $q_1(s_1, a_2)$, $q_1(s_2, a_1)$ and $q_1(s_2, a_2)$ for $\gamma=1$.

$$\begin{aligned} \text{We have } q_1(s_1, a_1) &= R(s_1, a_1) + P(s_1, a_1, s_1) V_0(s_1) + P(s_1, a_1, s_2) V_0(s_2) \\ &= 8 + 0.2 \times 10 + 0.6 \times 1 = 10.6. \end{aligned}$$

$$\text{Similarly } q_1(s_1, a_2) = 10 + 0.1 \times 10 + 0.2 \times 1 = 11.2.$$

$$\text{As } V_1(s_1) = \max_{a \in A} \{q_1(s_1, a)\}, \text{ we get } V_1(s_1) = 11.2 \text{ and } \pi_1(s_1) = a_2$$

$$\begin{aligned} \text{Besides: } q_1(s_2, a_1) &= 1 + 0.3 \times 10 + 0.3 \times 1 = 4.3 \\ q_1(s_2, a_2) &= -1 + 0.5 \times 10 + 0.3 \times 1 = 4.3 \end{aligned} \quad \begin{array}{l} \text{this gives us } V_1(s_2) = 4.3 \text{ and the optimal} \\ \text{action to take is either } a_1 \text{ or } a_2 \text{ in this case.} \end{array}$$

k=2: We have $V_1 = \begin{pmatrix} 11.2 \\ 4.3 \end{pmatrix}$

$$q_2(s_1, a_1) = 8 + 0.2 \times 11.2 + 0.6 \times 4.3 = 12.82$$

$$q_2(s_1, a_2) = 10 + 0.1 \times 11.2 + 0.2 \times 4.3 = 11.98$$

$$q_2(s_2, a_1) = 1 + 0.3 \times 11.2 + 0.3 \times 4.3 = 5.65$$

$$q_2(s_2, a_2) = -1 + 0.5 \times 11.2 + 0.3 \times 4.3 = 5.89$$

This gives us that
 $\rightarrow V_2(s_1) = 12.82$ and $V_2(s_2) = 5.89$
 and $\pi_2(s_1) = a_1$ and $\pi_2(s_2) = a_2$

Let us show that $\pi_k(\cdot)$ for $k \geq 2$ will be the same as π_2

$$\text{For } k \geq 2, \text{ we indeed have: } q_k(s_1, a_1) = 8 + 0.2 V_{k-1}(s_1) + 0.6 V_{k-1}(s_2)$$

$$q_k(s_1, a_2) = 10 + 0.1 V_{k-1}(s_1) + 0.2 V_{k-1}(s_2)$$

$$q_k(s_2, a_1) = 1 + 0.3 V_{k-1}(s_1) + 0.3 V_{k-1}(s_2)$$

$$q_k(s_2, a_2) = -1 + 0.5 V_{k-1}(s_1) + 0.3 V_{k-1}(s_2)$$

$$\text{In particular: } q_k(s_1, a_1) - q_k(s_1, a_2) = -2 + 0.1 V_{k-1}(s_1) + 0.4 V_{k-1}(s_2)$$

$$\text{As } V_k(s_1) = \max_{a \in A} q_k(s_1, a), \text{ we have } V_k(s_1) \geq 8 + 0.2 V_{k-1}(s_1) + 0.6 V_{k-1}(s_2)$$

$$\text{Similarly: } V_k(s_2) \geq -1 + 0.5 V_{k-1}(s_1) + 0.3 V_{k-1}(s_2)$$

$$\text{In particular for } k=3: V_3(s_1) \geq 8 + 0.2 \times 12.82 + 0.6 \times 5.89 = 14.098$$

$$V_3(s_2) \geq -1 + 0.5 \times 12.82 + 0.3 \times 5.89 = 7.177$$

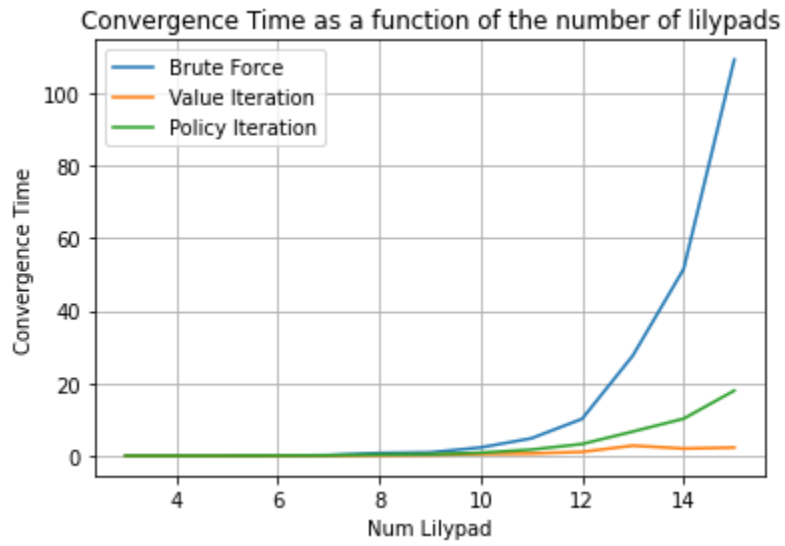
By induction, we can easily get: $\forall k \geq 2: V_k(s_1) \geq 10$ and $V_k(s_2) \geq 5$.

$$\text{Hence } q_k(s_1, a_1) - q_k(s_1, a_2) \geq -2 + 0.1 \times 10 + 0.3 \times 5 \geq 0$$

$$\text{Similarly } q_k(s_2, a_2) - q_k(s_2, a_1) \geq -2 + 0.2 V_{k-1}(s_1) = -2 + 0.2 \times 10 = 0$$

This gives us: $\forall k \geq 2: \pi_k(s_1) = a_1$
 and $\pi_k(s_2) = a_2$

We plotted the graph of the convergence speed for different values of n the number of lilypads using the code in `assignment4_code.py`. This gave us the following plot:



Problem 3:

Let us write the spaces and functions for this problem.

Here a person is characterized by its employment state and the job offered or employed which can be referred to by an index.

Hence $S = \{E, U\} \times \{i \mid 1 \leq i \leq n\}$ where i is the index of the job, E stands for employed and U for unemployed.

When someone is employed, it has no choice to make, and when someone is unemployed it has two actions to choose: Accept (A) or Decline (D).

The action space is $A = \{A, D\}$ but D is impossible when $s = (E, i)$ for any i .

Let us write the transition function. Let $s = (J, i)$ be the current state and $s' = (J', i')$ be the transition state:

$$P(s' | s, A) = \begin{cases} 1 - \alpha & \text{when } i' = i, J \in \{U, E\} \text{ and } J' = E \\ \alpha p_i' & \text{for each } 1 \leq i' \leq n, J \in \{U, E\} \text{ and } J' = U \text{ (situation where you lose your job but receive offer } i' \text{ after that)} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{And: } P(s' | s, D) = \begin{cases} p_i' & \text{for each } 1 \leq i' \leq n \text{ and } J' = U \\ 0 & \text{otherwise} \end{cases} \quad \text{in this case } s \text{ is of the form } (U, i): \text{ you can only choose to decline when you are already unemployed.}$$

Let us define the reward function: it is the function that defines the expected reward when taking action $a \in A$ in state $s \in S$. We call it $R(s, a) = R((J, i), a)$.

$$\text{We have: } R(s, A) = U(w_i) = \log(w_i) \quad \text{for each } s \text{ of the form } (J, i) \text{ where } 1 \leq i \leq n \text{ and } J \in \{U, E\}$$

$$R(s, D) = U(w_0) = \log(w_0) \quad \text{for each } s \text{ of the form } (U, i) \text{ where } 1 \leq i \leq n.$$

Let us write the Bellman Optimality Equation customized for this MDP.

When you are employed, you can only take one action which is: remain employed.

$$\text{Hence: } V^*((E, i)) = \frac{R((E, i), A)}{U(w_i)} + \gamma [(1 - \alpha) V^*((E, i)) + \alpha \sum_{i'=1}^n p_i' V^*((U, i'))]$$

When you are unemployed, you can choose to accept your offer or decline it.

$$\text{Hence: } V^*((U, i)) = \max \left(V^*((E, i)), U(w_0) + \gamma \sum_{i'=1}^n p_i' V^*((U, i')) \right)$$

This means that job offer i will be accepted only if: $V^*((E, i)) \geq U(w_0) + \gamma \sum_{i'=1}^n p_i' V^*((U, i'))$

We can simplify this problem. For the moment $V \in \mathbb{R}^{2n}$, but note that if we let $V_0 = \sum_{i'=1}^n p_i' V^*((U, i'))$, we get: $V_0 = \sum_{i'=1}^n p_i' \max(U(w_0) + \gamma V_0, V_i)$ and $\forall i \in \{1, n\} : V_i = U(w_i) + \gamma (\alpha V_0 + (1 - \alpha) V_i)$ where $V_i = V^*((E, i))$

The code to solve this Bellman Optimality Equation with a numerical iterative algorithm has been written in the `assignment4_code.py` file.

In our expressions of the value function, we used the simplification derived above (our value function is thus of dimension $n + 1$ instead of $2 \times n$, and the optimal policy only involves n choices to make: the choices you need to make when you are unemployed and receive a job offer of one of the n jobs.

The code written to solve this problem has also been written in the file `assignment4_code.py`