*A*ssignment 4 in Data mining for networks

"Reinforcement Learning : The Q-learning algorithm"

Lynda Attouche, Lenny Klump

February 2022

1 Escape Game

Reward matrix:

$$R = \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix}$$

Discount factor: $\gamma = 0.8$

1.1 Initializing Q-table:

Q-table is a matrix of size: 6x6 (number of states, number of actions).

We initialize it with zeros since we do not have any information at the beginning.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

1.2 First episode:

Initial state: room 1

At this state, there are 2 possible actions: go to state 3 with reward 0 or go to state 5 with reward 100.

We select to go to state 5. Looking to the reward matrix, in this new state we will have 3 actions: to go to 1 with reward 0, to go to 4 with reward 0 and to go to 5 (stay at the same room) with reward 100.

1.2.1 Computing the new value of $Q(1,5)$

$$\begin{aligned} Q(1,5) &= R(1,5) + \gamma \cdot max\big(Q(5,1), Q(5,4), Q(5,5)\big) \\ Q(1,5) &= 100 + 0.8 \cdot max(0,0,0) \\ Q(1,5) &= 100 \end{aligned}$$

1.2.2 Updating Q-table

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

1.3 Second episode:

Initial state: room 3

We have 3 actions: to go to 1 with reward 0, to go to 2 with reward 0 and to go to 4 with reward 0.
We select to go to state 1, the next actions will be: to go to state 3 with reward 0 or go to state 5 with reward 100.

We Compute the new value of $Q(3,1)$

$$\begin{aligned} Q(3,1) &= R(3,1) + \gamma \cdot max\big(Q(1,3), Q(1,5)\big) \\ Q(3,1) &= 0 + 0.8 \cdot max(0,100) \\ Q(3,1) &= 80 \end{aligned}$$

Updating the Q-table

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Our current state is now 1. By repeating the inner loop and selecting again 5, we will have 3 actions: to go to 1 or 4 or 5. If we select one of them, the update should be:

$$Q(1,5) = R(1,5) + \gamma \times max\big(Q(5,1), Q(5,4), Q(5,5)\big)$$

But since we didn't compute the values: $Q(5,1)$,$Q(5,4)$,$Q(5,5)$, the value $Q(1,5)$ won't change and thus the Q-table remain unchanged.

1.4 Exploring more episodes

By coding the Q_learning function in python (see the notebook), and taking a number of 400 episodes (after testing a good number of them, we found that 400 was the number needed for the algorithm to converge, as after this number the Q-table remains unchanged).

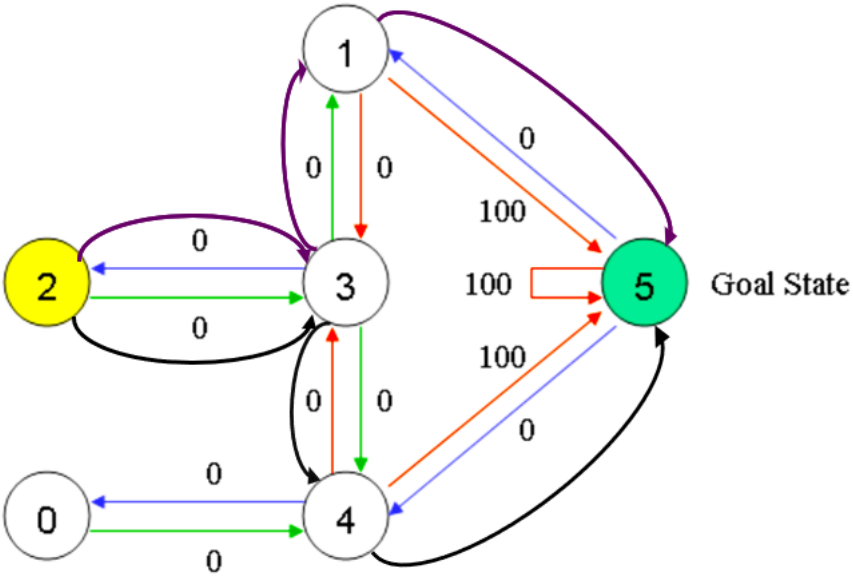
At the end, we obtained the following matrix:

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 400 & 0 \\ 0 & 0 & 0 & 320 & 0 & 500 \\ 0 & 0 & 0 & 320 & 0 & 0 \\ 0 & 400 & 256 & 0 & 400 & 0 \\ 320 & 0 & 0 & 320 & 0 & 500 \\ 0 & 400 & 0 & 0 & 400 & 500 \end{bmatrix}$$

1.5 Best sequence

Assuming that the initial state is room 2 and considering that we choose at each step the action that has the maximum Q-value, we move from state 2 to state 3 with a reward of 0. Then, from state 3 we go to state 1 with a reward of 0 or to state 4 with the same reward. Then from 1 (or 4), we go to state 5 with a reward of 100. At state 5, we remain there since it is our goal state. So, we have two sequences that are equal in terms of rewards (100) and that are the best.

We represent these two sequences in the graph as follows:



Sequence in purple: $2 \longrightarrow 3 \longrightarrow 1 \longrightarrow 5$
Sequence in black: $2 \longrightarrow 3 \longrightarrow 4 \longrightarrow 5$

2 Taxi game

Notebook