Homework n° 1 Rienforcement Learning

Solutions by Maria Cherifa

8 novembre 2020

Question 1 :

1. In this question we are looking for r_s such that the optimal policy is the shortest path to 14. First $r_s < 0$ because a path longer than the shortest path to 14 will have a greater reward, for example we are in the state 2 looking for the optimal policy means looking for the shortest path to 14, so we need to have $3\,r_s + 10 > r_s - 10$, which gives $r_s > -10$, thus $-9 \leq r_s \leq -1$, so to have the optimal policy let's take $r_s = -1$. For this value value function is equal to $V = -L + r_g$ where L is the length of the path from the current state to 14. So we have :

5	-10	7	6
6	7	8	5
7	8	9	4
8	9	10	3

2. Consider a general MDP with rewards, and transitions. Consider a discount factor of $\gamma < 1$. Assume th horizon is infinite. A policy π in this MDP induces a value function V^π. In this case we apply an affine transformation of the reward for example $\text{Newreward}(x, \pi(x)) = a\,r(x, \pi(x)) + b$, we want to compute the new value function V^π :

$$V^\pi_{new}(x) = \mathbf{E}(\sum_{t=0}^{\infty} \gamma^t \,(a\,r(x_t, \pi(x_t)) + b))|x_0 = x; \pi)$$

Using the conditional expectation linearity we have :

$$V^\pi_{new}(x) = a\mathbf{E}(\sum_{t=0}^{\infty} \gamma^t \, r(x_t, \pi(x_t))|x_0 = x; \pi) + b\,\mathbf{E}(\sum_{t=0}^{\infty} \gamma^t |x_0 = x; \pi)$$

the second term of the previous expression is not random so it is equivalent to :

$$V^\pi_{new}(x) = a\mathbf{E}(\sum_{t=0}^{\infty} \gamma^t \, r(x_t, \pi(x_t))|x_0 = x; \pi) + b\sum_{t=0}^{\infty} \gamma^t$$

In this case $\gamma < 1$ so $\sum_{t=0}^{\infty} \gamma^t$ converges to $\frac{1}{1-\gamma}$. So the new value function is equal to :

$$V^\pi_{new}(x) = a\,V^\pi(x) + \frac{b}{1-\gamma}$$

The optimal policy is not preserved since the value function changed, so the solution of the maximization problem which gives us the optimal policy changed.

3. Consider the same setting as in question 1, here the new reward is given by $r_s = r_s + 5$. The optimal policy becomes a policy that would just wander around forever, never reaching either target. Using the previous question we see that the new value function is giving by :

$$\begin{aligned} V^{\pi}_{new}(x) &= V_{old} + \frac{5}{1-\gamma} \\ &= r_s\, L + r_g + \frac{5}{1-\gamma} \end{aligned}$$

Where L is the length of the path from the current state to 14.

Question 2 :

Consider infinite-horizon $\gamma-$discounted Markov Decision Processes with S states and A actions. Denote by Q^* the Q-function of the optimal policy π^*. We want to prove that for any function $Q(s,a)$, the following inequality holds for any s,

$$V^{\pi_Q}(s) \geq V^*(s) - \frac{2\,||Q^* - Q||_\infty}{1-\gamma}$$

Let's start the proof :

$$\begin{aligned} V^*(s) - V^{\pi_Q}(s) &= Q^*(s,\pi^*(s)) - Q^*(s,\pi_Q(s)) + Q^*(s,\pi_Q(s)) - Q^{\pi_Q}(s,\pi_Q(s)) \\ &\leq Q^*(s,\pi^*(s)) - Q(s,\pi^*(s)) + Q(s,\pi_Q(s)) - Q^*(s,\pi_Q(s)) + \gamma\, \mathbf{E}_{s'\sim P(s,\pi_Q(s))}(V^*(s') - V^{\pi_Q}(s')) \\ &\leq 2||Q - Q^*||_\infty + \gamma||V^* - V^{\pi_Q}|| \end{aligned}$$

So we find,

$$||V^* - V^{\pi_Q}||_\infty \leq \frac{2\,||Q - Q^*||_\infty}{1-\gamma}$$

which implies

$$V^{\pi_Q}(s) \geq V^*(s) - \frac{2\,||Q^* - Q||_\infty}{1-\gamma}$$

Question 3 :

Consider the average reward setting ($\gamma = 1$) and a Markov Decision Process with S states and A actions. We want to prove that

$$g^{\pi'} - g^{\pi} = \sum_s \mu^{\pi'}(s) \sum_a (\pi'(a|s) - \pi(a|s))\, Q^{\pi}(s,a)$$

using the hints given.

Let's focus on the right member of the above equation :

$$A = \sum_s \mu^{\pi'}(s) \sum_a (\pi'(a|s) - \pi(a|s))\, Q^{\pi}(s,a)$$

Let's replace $Q^{\pi}(s,a)$ by the Bellman equation giving, so we have :

$$\begin{aligned} A &= \sum_s \mu^{\pi'}(s) \sum_a (\pi'(a|s) - \pi(a|s)) \left(r(s,a) - g^{\pi} + \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') Q^{\pi}(s',a') \right) \\ &= \sum_{s,a} \mu^{\pi'}(s)(\pi'(a|s) - \pi(a|s))(r(s,a) - g^{\pi}) + \sum_{s,a,s',a'} \mu^{\pi'}(s)(\pi'(a|s) - \pi(a|s)) p(s'|s,a) \pi(a'|s') Q^{\pi}(s',a') \end{aligned}$$

Let $S_1 = \sum_{s,a} \mu^{\pi'}(s)(\pi'(a|s) - \pi(a|s))(r(s,a) - g^\pi)$ and $S_2 = \sum_{s,a,s',a'} \mu^{\pi'}(s)(\pi'(a|s) - \pi(a|s))p(s'|s,a)\pi(a'|s')Q^\pi(s',a')$ and using the fact that $g^\pi = \sum_x \pi^\pi(s) \sum_a \pi(a|s) r(s,a)$ we have :

$$\begin{aligned} S_1 &= \underbrace{\sum_{s,a} \mu^{\pi'}(s)(\pi'(a|s)r(s,a) - \pi(a|s)r(s,a) - \pi'(a|s)g^\pi + \pi(a|s)g^\pi)}_{g^{\pi'}} \\ &= g^{\pi'} - \sum_{s,a} \mu^{\pi'}(s)\pi'(a|s)g^\pi - \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)r(s,a) + \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)g^\pi \end{aligned}$$

As we know $\sum_{s,a} \mu^{\pi'}(s)\pi(a|s) = 1$ also $\sum_{s,a} \mu^{\pi'}(s)\pi'(a|s) = 1$, so we have :

$$\begin{aligned} S_1 &= g^{\pi'} - \sum_{s,a} \mu^{\pi'}(s)\pi'(a|s)g^\pi - \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)r(s,a) + \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)g^\pi \\ &= g^{\pi'} - g^\pi - \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)r(s,a) + g^\pi \\ &= g^{\pi'} - \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)r(s,a) \end{aligned}$$

Let's try to simplify S_2 :

$$\begin{aligned} S_2 &= \sum_{s,a,s',a'} \mu^{\pi'}(s)(\pi'(a|s) - \pi(a|s))p(s'|s,a)\pi(a'|s')Q^\pi(s',a') \\ &= \sum_{s',a'} \pi(a'|s')Q^\pi(s',a') \underbrace{\sum_{s,a} \mu^{\pi'}(s)(\pi'(a|s) - \pi(a|s))p(s'|s,a)}_{\mu^{\pi'}(s') - \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)p(s'|s,a)} \\ &= \sum_{s',a'} \pi(a'|s')Q^\pi(s',a')\mu^{\pi'}(s') - \sum_{s',a'} \pi(a'|s')Q^\pi(s',a') \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)p(s'|s,a) \\ &= \sum_{s',a'} \pi(a'|s')Q^\pi(s',a')\mu^{\pi'}(s') - \sum_{s,a} \mu^{\pi'}(s)\pi(a|s) \sum_{s',a'} p(s'|s,a)\pi(a'|s')Q^\pi(s',a') \end{aligned}$$

Using the fact that the bellman equation gives us $\sum_{s',a'} p(s'|s,a)\pi(a'|s')Q^\pi(s',a') = Q^\pi(s,a) - r(s,a) + g^\pi$ we have :

$$\begin{aligned} S_2 &= \sum_{s',a'} \pi(a'|s')Q^\pi(s',a')\mu^{\pi'}(s') - \sum_{s,a} \mu^{\pi'}(s)\pi(a|s) \sum_{s',a'} p(s'|s,a)\pi(a'|s')Q^\pi(s',a') \\ &= \underbrace{\sum_{s',a'} \pi(a'|s')Q^\pi(s',a')\mu^{\pi'}(s') - \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)Q^\pi(s,a)}_{=0} - \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)(g^\pi - r(s,a)) \\ &= -\sum_{s,a} \mu^{\pi'}(s)\pi(a|s)g^\pi + \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)r(s,a) \end{aligned}$$

Using the fact that $\sum_{s,a} \mu^{\pi'}(s)\pi(a|s) = 1$, we find that :

$$S_2 = -g^\pi + \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)r(s,a)$$

So we find that :

$$\begin{aligned} A &= S_1 + S_2 \\ &= g^{\pi'} - \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)r(s,a) - g^\pi + \sum_{s,a} \mu^{\pi'}(s)\pi(a|s)r(s,a) \\ &= g^{\pi'} - g^\pi \end{aligned}$$

Thus we find that :

$$g^{\pi'} - g^{\pi} = \sum_s \mu^{\pi'}(s) \sum_a (\pi'(a|s) - \pi(a|s))\, Q^{\pi}(s,a)$$

Question 4 :

In this question we want to model the elevator despatching problem using an MDP. As we know to formulate this problem in the framework of MDP, the state space, action space, cost or reward and value function need to be defined.

We consider a 6-story building with 2 elevators and the the agent can simultaneously control all the elevators.

— **The state Set :** We define $S = \{s_t\}$ as the discrete state set of two elevators control system, including the direction and current position of each elevator, calls belong to each elevator, hall calls (landing floor calls) already existing and the new hall call which has not been allocated. Each hall call has direction and the number of the floor where the button is pressed.

— **The action Set :** Let $U = \{u_i\}$ denote the action set. u_i refers to the dispatching algorithm allocating the new hall call to the elevator 1 or 2.

— **Cost and Action-value Function :** To define the cost and action-value function we take into account the waiting and journey times of passengers existing in each elevator when last decision is made and the number of stops in the future according to the existing elevator calls.

1. Waiting time : Let $T(p)$ be the waiting time of passenger p who arrives before the last decision making. The total waiting time of all passengers is defined by :

$$R_1 = \sum_p T(p) = \sum_p (t - t_p)$$

where t_p is the arriving time of passenger p.

2. Journey time : Let $T'(p')$ be the journey time of passenger p' existing in an elevator before the last decision making. The total waiting time of all passengers in elevators is defined by :

$$R_2 \sum_{p'} T'(p') = \sum_{p'} (t - t_r)$$

Where t_r denotes the time of passenger p entering the elevator.

3. Number of stops : the number of stops of the elevators is giving by :

$$R_3 = \frac{1}{2} \sum_{i=1}^{2} C_i$$

where C_i denotes the number of stops for each elevator . Thus, the total cost of the system is giving by :

$$R = R_1 + R_2 + R_3$$

So the value function of an infinite-horizon is giving by :

$$\begin{aligned} V^{\pi}(s) &= \mathbf{E}(R_{t+1} + \gamma\, R_{t+2} + \gamma^2\, R_{t+3} + ... | s_t = s, \pi) \\ &= \sum_s \pi(s,u)\, (R_s^u + \sum_{s'} \gamma\, P_{ss'}^u V^{\pi}(s')) \end{aligned}$$

where :

$$\mathbf{P}(s_{t+1} = s' | s_t = s, u_t = u) = P_{ss'}^u$$

is the probability transition between state s and s' taking the action u.

$$R(s, u, s') = R^u_{ss'}$$

is the reward giving in the transition between state s and s' taking the action u.

$$\pi(s, u)$$

is the policy that gouverns the choice of the actions.

So solving the MDP is to find :

$$V^{\pi^*} = \min_{\pi} V^{\pi} = \min_{u} R^u_s + \sum_{s'} \gamma \, P^u_{ss'} V^{\pi^*}(s')$$

Question 5 :

In this question we are going to implement value iteration and policy iteration.

1. Implementation of policy iteration.
2. Implementation of value iteration.
3. Stochasticity generally increases the number of iterations required to converge. In the stochastic environment, the number of iterations for value iteration increases. For policy iteration, depending on the implementation method, the number of iterations could remain unchanged ; or policy iteration might not even converge at all. The stochasticity would also change the optimal policy. The optimal policy of the stochastic environment is different from the one of the deterministic.

4. Here are some differences between value iteration and policy iteration algorithms :
   — **Policy iteration** includes : **policy evaluation** + **policy improvement**, and the two are repeated iteratively until policy converges.
   — **Value iteration** includes : finding optimal value function + one policy extraction. There is no repeat of the two because once the value function is optimal, then the policy out of it should also be optimal (convergence).
   — **Finding optimal value function** can also be seen as a combination of policy improvement (due to max) and truncated policy evaluation.
   — The algorithms for **policy evaluation** and **finding optimal value function** are highly similar except for a max operation.
   — Similarly, the key step to **policy improvement** and **policy extraction** are identical except the former involves a stability check.
   Value iteration : the complexity is very large to obtain an approximation of V^*.
   Policy iteration : converges in a finite number of steps, but requires, but requires an evaluation of the policy in each step.