MVA: Reinforcement Learning (2020/2021) **Homework 1**

Dynamic Programming

Lecturers: *A. Lazaric, M. Pirotta* *(November 8, 2020)*

Solution by AMEKOE Kodjo Mawuena

Instructions

- The deadline is **November 8, 2020. 23h00**
- By doing this homework you agree to the *late day policy, collaboration and misconduct rules* reported on Piazza.
- **Mysterious or unsupported answers will not receive full credit**. A correct answer, unsupported by calculations, explanation, or algebraic work will receive no credit; an incorrect answer supported by substantially correct calculations and explanations might still receive partial credit.
- Answers should be provided in **English**.

1 Question

Consider the following grid environment. The agent can move up, down, left and right. Transitions are deterministic. Attempts to move in the direction of the wall will result in staying in the same position. There are two absorbing states: 1 and 14. Taking any action in 1 (resp 14) leads to a reward r_r (resp. r_g) ($r(1,a) = r_r, r(14,a) = r_g, \forall a$) and *ends the episode*. Everywhere else the reward is r_s. Assume discount factor $\gamma = 1$, $r_g = 10$ and $r_r = -10$, unless otherwise specified.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

1. Define r_s such that the optimal policy is the shortest path to state 14. Using the chosen r_s, report the value function of the optimal policy for each state.
   There is a simple solution that doesn't require complex computation. You can copy the image an replace the id of the state with the value function.

2. Consider a general MDP with rewards, and transitions. Consider a discount factor of $\gamma < 1$. For this case assume that the horizon is infinite (so there is no termination). A policy π in this MDP induces a value function V^π. Suppose an affine transformation is applied to the reward, what is the new value function? Is the optimal policy preserved?

3. Consider the same setting as in question 1. Assume we modify the reward function with an additive term $c = 5$ (i.e., $r_s = r_s + c$). How does the optimal policy change (just give a one or two sentence description)? What is the new value function?

Answers

1. Definition of r_s such that the optimal policy is the shortest path:
   The value function at the state s is. $V(s) = E\left[\sum_{t=0}^{T} r_t\right]$. As the transitions are deterministic, we have $V(s) = \sum_{t=0}^{T} r_t$ where r_t is the reward at the state s_t.
   The optimal policy is the one which maximizes the cumulative reward towards the state 14. We have to solve: $\min_l\{lr_s + r_g > kr_s + r_r\}$, where l is length of the path to the state 14, k the length to the state 1, $r_g = 10$ and $r_r = -1$. When $l = 1$, state (state 13 and 10), we have $r_s < 10$, which is less informative. Because for example $r_s = 1$, the agent may prefer to move around the grid indefinitely. For $l = 2$, any value of r_s is acceptable so is not the solution. Finally for $l = 3$ (state 5 or 2) we have $\underline{r_s > -10}$. Hence for the rest of the exercice, we will use $\underline{r_s = -1}$.
   Value function at each state (for $r_s = -1$):

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

2. Now let's consider the a general MDP with a discount factor $\gamma < 1$ in infinite horizon case. V^π the value fonction induce by the policy π. Let assume that an affine transformation $(r \to mr + c)$ is applied to reward reward r. Then the new valus fonction is :

$$\begin{aligned} V^\pi_{aff} &= E\left[\sum_{t=0}^{\infty} \gamma^t (mr_t + c)\right] \\ &= mE\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] + cE\left[\sum_{t=0}^{\infty} \gamma^t\right] \\ V^\pi_{aff} &= mV^\pi + \frac{c}{1-\gamma}. \end{aligned} \tag{1}$$

   In this case:

   - if $m \geq 0$, $x \to mx + c$ is nondecreasing, so if the $V^{\pi^\star}$ the optimal value fonction, then $mV^{\pi^\star} + \frac{c}{1-\gamma}$ is the optimal value function associeted to the reward $mr + c$. Thus the optimal policy is preserved.
   - if $m < 0$, the optimal policy is not preserved.

3. We have $\gamma = 1$ and $c = 5$, then $r_s = r_s + 5 = 4$. The optimal policy is the one which allows the agent to move indefinitely in the grid and avoid states 1 , 14. The new value function is $V^\pi_5 = E\left[\sum_{t=0}^{H}(mr_t + 5)\right] = E\left[\sum_{t=0}^{H} mr_t\right] + 5(H+1) \to \infty$ as $H \to \infty$. H is the horizon.

2 Question

Consider infinite-horizon γ-discounted Markov Decision Processes with S states and A actions. Denote by $Q^\star$ the Q-function of the optimal policy $\pi^\star$. Prove that, for any function $Q(s, a)$, the following inequality holds for any s

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

where $e = (1, \ldots, 1)$, $\|Q^\star - Q\|_\infty = \max_{s,a} |Q^\star(s,a) - Q(s,a)|$ and $\pi_Q(s) = \arg\max_a Q(s,a)$. Thus $\pi^\star(s) = \arg\max_a Q^\star(s,a)$.

Answers

Let's define $L_Q(s) = Q^\star(s, \pi^\star(s)) - Q(s, \pi_Q(s))$ (The loss). Let s' be the state that achieves the max of loss. Then $L_Q(s') \geq L_Q(s), \forall s \in S$. Let's also $a_1 = \pi^\star(s')$ and $a_2 = \pi_Q(s^{'})$ and $\epsilon = \|Q^\star - Q\|_\infty$.
As $\pi_Q(s) = \arg\max_a Q(s,a)$ (greedy policy),

$$\underline{Q(s,a_1) \leq Q(s,a_2)}. \tag{2}$$

Since $|Q^\star(s,a) - Q(s,a)| \leq \epsilon = \|Q^\star - Q\|_\infty$, for $a \in A$, $s \in S$,
$Q^\star(s',a_1) - \epsilon \leq Q(s',a_1) \leq Q^\star(s',a_1) + \epsilon$ and $Q^\star(s',a_2) - \epsilon \leq Q(s',a_2) \leq Q^\star(s',a_2) + \epsilon$.
Thus we have, $Q^\star(s',a_1) - \epsilon \leq Q(s',a_1) \leq Q(s',a_2) \leq Q^\star(s',a_2) + \epsilon$.

$$\begin{aligned}
&\implies && Q^\star(s',a_1) - \epsilon \leq Q^\star(s',a_2) - \epsilon \\
&\implies && r(s',a_1) + \gamma \sum_y p(y|s',a_1) V^\star(y) - \epsilon \leq r(s',a_2) + \gamma \sum_y p(y|s',a_2) V^\star(y) - \epsilon \\
&\implies && \underline{r(s',a_1) - r(s',a_2) \leq 2\epsilon + \gamma \sum_y V^\star(y) \left[-p(y|s',a_1) + p(y|s',a_2)\right]}
\end{aligned} \tag{3}$$

Let's define $L_V(s') = V^\star(s') - V^{\pi_Q}(s')$. Then :

$$\begin{aligned}
L_V(s') &= r(s',a_1) + \gamma \sum_y p(y|s',a_1) V^\star(y) - r(s',a_2) - \gamma \sum_y p(y|s',a_2) V^{\pi_Q}(y) \\
&= \underline{r(s',a_1) - r(s',a_2)} + \gamma \sum_y p(y|s',a_1) V^\star(y) - \gamma \sum_y p(y|s',a_2) V^{\pi_Q}(y)
\end{aligned}$$

By the 3, we have:

$$\begin{aligned}
L_V(s') &\leq 2\epsilon + \gamma \sum_y V^\star(y) \left[-p(y|s',a_1) + p(y|s',a_2)\right] + \gamma \sum_y p(y|s',a_1) V^\star(y) - \\
&\gamma \sum_y p(y|s',a_2) V^{\pi_Q}(y) \\
&= 2\epsilon + \gamma \sum_y V^\star(y) \left[p(y|s',a_1) + p(y|s',a_2) - p(y|s',a_1)\right] - \gamma \sum_y p(y|s',a_2) V^{\pi_Q}(y) \\
&= 2\epsilon + \gamma \sum_y p(y|s',a_2) \left[V^\star(y) - V^{\pi_Q}(y)\right] \\
&= 2\epsilon + \gamma \sum_y p(y|s',a_2) L_V(y) \\
&\leq 2\epsilon + \gamma L_V(s') \sum_y p(y|s',a_2), \text{since the state } s' \text{ achieves the max of loss}
\end{aligned}$$

So we have,

$$\begin{aligned}
L_V(s') \leq 2\epsilon + \gamma L_V(s') &\implies (1-\gamma) L_V(s') \leq 2\epsilon \\
&\implies L_V(s') \leq \frac{2\epsilon}{1-\gamma} \\
&\implies -L_V(s) \geq -L_V(s') \geq -\frac{2\epsilon}{1-\gamma} \\
&\implies V^{\pi_Q}(s) - V^\star(s) \geq -\frac{2\epsilon}{1-\gamma}
\end{aligned}$$

Finally, we have the inequality:

$$\underline{V^{\pi_Q}(s) \geq V^\star(s) - \frac{2\epsilon}{1-\gamma}} \tag{4}$$

3 Question

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

$$g^{\pi'} - g^{\pi} = \sum_s \mu^{\pi'}(s) \sum_a (\pi'(a|s) - \pi(a|s)) Q^{\pi}(s,a) \tag{5}$$

using the fact that in average reward the Bellman equation is

$$Q^{\pi}(s,a) = r(s,a) - g^{\pi} + \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') Q^{\pi}(s',a'), \quad \forall s,a,\pi$$

and μ^{π} is the **stationary distribution** of policy π. Note also that $g^{\pi} = \sum_x \mu^{\pi}(s) \sum_a \pi(a|s) r(s,a)$.

Note: All the information provided to prove Eq. 5 are mentioned in the question. Start from the definition of Q and use the property of stationary distribution.

Answers

Prove of Eq. 5: In average reward the Bellman equation is :

$$Q^{\pi}(s,a) = r(s,a) - g^{\pi} + \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') Q^{\pi}(s',a'), \quad \forall s,a,\pi$$

$$\implies r(s,a) = Q^{\pi}(s,a) - \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') Q^{\pi}(s',a') + g^{\pi} \tag{6}$$

We have also for a policy π' : $g^{\pi'} = \sum_s \mu^{\pi'}(s) \sum_a \pi'(a|s) r(s,a)$. The by using Eq.6, we have:

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

because $\sum_s \mu^{\pi'}(s) \sum_a \pi'(a|s) = 1$.
Since $\sum_{s,a} p(s'|s,a) \pi'(a|s) \mu^{\pi'}(s) = \mu^{\pi'}(s')$ (stationary distribution property), we have :

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

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

4 Question

Provide an MDP modeling, specifying all its defining elements, of the following process:

- Elevator dispatching. The elevator controllers assign elevators to service passenger requests in real-time while optimizing the overall service quality, e.g. by minimizing the waiting time and/or the energy consumption. The agent can simultaneously control all the elevators. In order to model this problem, consider a 6-story building with 2 elevators. Explain the choices made for modeling this problem.

Answers

Using MDP to model this problem, our goal is to minimize the waiting time, so the MDP is tuple (S, A, p, r) where :

- $S = \{S_t\}$: discrete state of elevator system (agent). This state includes the direction, the current position of the car (elevator) etc...
- As we have 2 elevators (cars), the set of action $A = \{a_1, a_2\}$, with $a_i, i \in \{1, 2\}$ the action of allocating the call (new) of the passenger Pa to the car i.
- $p(s'|s, a)$ the transition probability (is suppose unknown).
- $r(s, a, s')$ the reward when taking the action a from the state s to s'. In our modeling, we define this reward with respect to the waiting time T_{Pa} of the passenger Pa who arrives before the last action (decision) of the agent. So the reward should be the sum of waiting time of all passengers : $r = \sum_{Pa} T_{Pa}$

5 Question

Implement value iteration and policy iteration. We have provided a custom environment in the starter code.

1. (coding) Consider the provided code `vipi.py` and implement `policy_iteration`. Use γ as provided by `env.gamma` and the terminal condition seen in class. Return the optimal value function and the optimal policy.
2. (coding) Implement `value_iteration` in `vipi.py`. The terminal condition is based on $\|V_{new} - V_{old}\|_\infty$ and the tollerance is 10^{-5}. Return the optimal value function and the optimal policy.
3. (written) Run the methods on the deterministic and stochastic version of the environment. How does stochasticity affect the number of iterations required, and the resulting policy?
   You can render the policy using `env.render_policy(policy)`
4. (written) Compare value iteration and policy iteration. Highlight pros and cons of each method.

Answers

1. Coding part
2. Coding part
3. The more we are in a stochastic environment (by decreasing the probability of success which controls the stochasticity of the environment), the more the number of iterations of value_iteration algorithm decreases. So, I go from 1137 (in the deterministic case) to 939 iterations when the probability of success is 0.9.

For policy_iteration, the I noticed that the numbers of iterations varies around 5 by modifying the stochasticity of the environment.

The optimal policy given by the two algorithms remains equal, regardless of the modification of the stochasticity of the environment. However the policy (optimal) obtained by using each of these two methods may change depending on the stochasticity of the environment

4. The number of iterations using value_iteration greatly exceeds that of policy_iteration. In the deterministic environment example, I have 1137 iterations against 5.

   However, looking at the execution time versus the number of iterations, we found that the iterations of policy_iterations are more expensive : 0.35s for 1137 iterations vers 0.099s for 5 iterations in the deterministic case (Average on 10 measures using python package time).
   So according to the dimension of the environment, of the space of actions, a compromise in the choice of the algorithms is necessary.

Conclusion

This first homework assignment allowed me to review the concepts of MDP discussed during the course on the theoretical and experimental aspects. Unless there is a programming error, the practical results obtained agree with those of the course. I am expecting the reinforcement learning one very soon.
It is important to note that the experimental values obtained come from a personnal computer (6 cores)