Solutions to Reinforcement Learning by Sutton Chapter 3

Yifan Wang

April 2019

Exercise 3.1

We have to start by reviewing the definition of MDP. MDP consists of agent, state, actions, reward and return. But the most important feature is called the *Markov property.Markov property* is that each state must provide ALL information with respect to the past agent-environment interactions. Under such optimized assumption, we could define a probability function $p(s', r|s, a)$ as follows, given $s \in S$ is the state, $s' \in S$ is the next state resulted by action $a \in R$.

$$p(s', r|s, a) \doteq \texttt{Pr}\{S_t = s', R_t = r|S_{t-1} = s, A_{t-s} = a\}$$

Under such p, we can calculate many variations of functions to start our journey. Going back to the question, I will give the following 3 examples. First, an AI to escape the maze of grid.Second, an AI that is trying to learn which arm to use to grasp a glass. Third, an AI that is going to decide how hot its heater should be to make water 60 degrees hot. ■

Exercise 3.2

One clear exceptions when we do not have enough calculation power to find each s and r, for example the game Go, which has to be solved by deep learning framework. Another clear exception is when you cannot put state clear. For example, playing an online game first time, how could an AI build models ahead of knowing nothing about the game? Even the goal is to win, the state is hard to define. Even

playing so many times, simple rules can have mathematically impossible number of states. Finally, the exception could be found to violate the *Markov Property*. That is, any former actions have no observable result in the current state. For example, thinking of playing shooting game, the agent has no direct information about other players due to the sight restriction but the state will be influenced by your teammate and the opponent, making the agent impossible to figure out what is the effect of your former action on to the current situation, which makes is not a MDP. ■

Exercise 3.3

This problem is asking the proper line to define the environment and the agent. To my understanding, the line should be divided such that the effect of agent's action a on state s could be observed in some way. For instance, in the auto-driving problem, if we draw the line where we only consider where to go, how would our actions affect the state in a clear way? It is not observable and nearly abstract to me unless we have a series of agents that could decompose the job. That is indeed the truth when the modern auto-driving system has many sub-systems, for example, detection of trees. In general, I think the line should be drawn by what we can do in nature and based on what sub-agent we are able to build. ■

Exercise 3.4

The resulted table is the following:

s	a	s'	r	$p(s', r\|s, a)$
high	search	high	r_{search}	α
high	search	low	r_{search}	$1-\alpha$
low	search	high	-3	$1-\beta$
low	search	low	r_{search}	β
high	wait	high	r_{wait}	1
high	wait	low	-	0
low	wait	high	-	0
low	wait	low	r_{wait}	1
low	recharge	high	0	1
low	recharge	low	-	0

■

Exercise 3.5

$$(original) \sum_{s' \in S} \sum_{r \in R} p(s', r|s, a) = 1, \text{ for all } s \in S, a \in A(s).$$

$$(modified) \sum_{s' \in S^+} \sum_{r \in R} p(s', r|s, a) = 1, \text{for all } s \in S^+, a \in A(s), \text{ and } S^+ \doteq \{\text{Non-terminal States}\}$$

■

Exercise 3.6

First, review the G_t for an episodic task:

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T$$

If we use discounting, this will be:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... + \gamma^{T-t-1} R_T = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}$$

The reward for success is set to 0 and for failure, R_T, is set to -1. Thus we will have an updated return as

$$-\gamma^{T-t-1}$$

This actually is the same return as the continuing setting where we have return as $-\gamma^K$ where K is the time step before the failure. ■

Exercise 3.7

If you do not use γ to implement the discount, the maximum return is always 1 regardless the time the agent spends. The correct way to communicate the agent is to add -1 punishment to each time step before the escape or adding the discount. ■

Exercise 3.8

$$\begin{aligned}
G_5 &= 0 && \text{(terminal)}\\
G_4 &= 2\\
G_3 &= 0.5\,G_4 + 3 = 4\\
G_2 &= 0.5\,G_3 + 6 = 2 + 6 = 8\\
G_1 &= 0.5\,G_2 + 2 = 4 + 2 = 6\\
G_0 &= 0.5\,G_1 - 1 = 3 - 1 = 2
\end{aligned}$$

■

Exercise 3.9

$$G_0 = R_1 + \gamma G_1 = 2 + \gamma \sum_{k=0}^{\infty} \gamma^k = 2 + \frac{0.9 * 7}{1 - 0.9} = 65$$

■

Exercise 3.10

Proof:

$$(\sum_{k=0}^{\infty} \gamma^k)(1-\gamma) = \sum_{k=0}^{\infty} \gamma^k (1-\gamma) = \sum_{k=0}^{\infty} (\gamma^k - \gamma^{k+1}) = 1 - \lim_{k\to\infty} \gamma^{k+1} = 1 - 0 = 1$$

Thus:

$$\sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$$

■

Exercise 3.11

$$\mathbb{E}[R_{t+1}|S_t = s] = \sum_a \pi(a|S_t) \sum_{s',r} p(s', r|s, a)r$$

■

Exercise 3.12

$$v_\pi(s) \doteq \sum_a \pi(a|s) q_\pi(s, a)$$

■

Exercise 3.13

$$q_\pi(s, a) = \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')]$$

■

Exercise 3.14

Bellman equation is known as the following:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')]$$

, for all $s \in S$. Thus we have

$$v_{center} = \frac{0.9 \times (2.3 + 0.7 - 0.4 + 0.4)}{4} = 0.675 \approx 0.7$$

■

Exercise 3.15

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

by adding a constant C

$$G_t^* \doteq G_t + \sum_{k=0}^{\infty} \gamma^k C = G_t + \frac{C}{1-\gamma}$$

$$v_\pi^*(s) \doteq \mathbb{E}\Big[G_t^* | S_t = s\Big] = \mathbb{E}\Big[G_t + \frac{C}{1-\gamma} | S_t = s\Big] = \mathbb{E}\Big[G_t | S_t = s\Big] + \frac{C}{1-\gamma}$$

The last step uses the linearity of expectation and it is trivial to conclude that the new $v^*(s)$ **does not affect the relative difference among states.** ■

Exercise 3.16

It is a similar question as one before. The sign of reward is critical in episodic task because episodic task uses negative reward to accelerate the agent finishing the task. Thus, adding a constant C, if changing the sign, would have an impact on how agent moves. Furthermore, if the negative reward remains negative but the value of it shrinks too much, it will give a wrong signal to the agent that the time of completing the job is not that important. ■

Exercise 3.17

$$\begin{aligned} q_\pi(s,a) &\doteq \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] \\ &= \sum_{s',r} p(s', r | s, a)[r + \gamma \sum_{a'} \pi(a'|s') q_\pi(s', a')] \end{aligned}$$

■

Exercise 3.18

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[q_\pi(s,a)] \\ &= \sum_a \pi(a|s) q_\pi(s,a) \end{aligned}$$

■

Exercise 3.19

$$\begin{aligned} q_\pi(s,a) &= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s, A_t = a] \\ &= \sum_{s',r} p(s', r|s,a)[r + \gamma v_\pi(s')] \end{aligned}$$

■

Exercise 3.20

It is a combination with v_{putt} and $q_*(s, driver)$ where we have outside of green part plus the sand (since we cannot escape using the putt)the q_* and the rest for v_{putt}.

■

Exercise 3.21

****Almost same as v_{putt} but avoid the part of sand? This question is vague since the rule of golf is not clearly stated.**

■

Exercise 3.22

$$G_{\pi_{\text{left}}} = \sum_{i=0}^{\infty} \gamma^{2i} = \frac{1}{1-\gamma^2}$$
$$G_{\pi_{\text{right}}} = \sum_{i=0}^{\infty} 2\gamma^{1+2i} = \frac{2\gamma}{1-\gamma^2}$$

Based on the above return formulas for each policy, $\gamma = 0.5$ seems to be the borderline. If $\gamma > 0.5$, right is optimal; if $\gamma < 0.5$, left is optimal. If $\gamma = 0.5$, both are optimal.

■

Exercise 3.23

Bellman optimality equation for q_* is:

$$\begin{aligned} q_*(s,a) &= \mathbb{E}\Big[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \,\Big|\, S_t = s, A_t = a\Big] \\ &= \sum_{s',r} p(s', r|s, a)\Big[r + \gamma \max_{a'} q_*(s', a')\Big] \end{aligned}$$

If s = high, a = wait:

$$q_*(high, wait) = r_{wait} + \gamma \max(q_*(high, wait), q_*(high, search))$$

If s = high, a = search:

$$\begin{aligned} q_*(high, search) = &\alpha\left[r_{search} + \gamma \max_a(q_*(high, wait), q_*(high, search))\right] \quad + \\ &(1-\alpha)[r_{search} + \gamma \max_a(q_*(low, recharge), q_*(low, wait), q_*(low, search))] \end{aligned}$$

Similar equations can be made for the rest, it is trivial to do more here.

■

Exercise 3.24

The best solution after reaching A is to quickly go back A after moving to A. That takes 5 time steps. So we will have

$$v_*(A) = \sum_{t=0}^{\infty} 10\,\gamma^{5t}$$

Theoretical answer is $\frac{10}{1-\gamma^5}$ By writing a little function in python, (looping 100 times is enough), or use a calculator, we get the answer 24.419428096993954. Cutting it to 3 digits and we are done at 24.419.

■

Exercise 3.25

$$v_*(s) = \max_a(q_*(s,a))$$

■

Exercise 3.26

$$q_*(s,a) = \sum_{s',r} p(s',r|s,a)[r + \gamma v_*(s')]$$

■

Exercise 3.27

$$a_* \doteq \arg\pi_*(a_*|s) = \arg\max_a q_*(s,a)$$

Policies that map only these a_* to their arbitrary possibilities would be the π_*.

■

Exercise 3.28

$$a_* \doteq \arg\pi_*(a_*|s) = \arg\max_a \sum_{s',r} p(s',r|s,a)[r + v_*(s')]$$

■

Exercise 3.29

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi\Big[G_t|S_t = s\Big] \\
&= \sum_a \big[r(s,a) + \gamma \sum_{s'} p(s'|s,a) v_\pi(s')\big]\pi(s,a)
\end{aligned}
$$

$$
\begin{aligned}
v_*(s) &\doteq \mathbb{E}_\pi\Big[G_t|S_t = s\Big] \\
&= \sum_a \big[r(s,a) + \gamma \sum_{s'} p(s'|s,a) v_*(s')\big]\pi_*(s,a)
\end{aligned}
$$

$$
\begin{aligned}
q_\pi(s,a) &\doteq \mathbb{E}_\pi\Big[G_t|S_t = s, A_t = a\Big] \\
&= \mathbb{E}_\pi\Big[R_{t+1} + \gamma G_{t+1}|S_{t+1} = s', A_t = a\Big] \\
&= r(s,a) + \gamma \sum_{s'} p(s'|s,a) \sum_{a'} q_\pi(a',s')\pi(a'|s')
\end{aligned}
$$

$$
\begin{aligned}
q_*(s,a) &\doteq \mathbb{E}_{\pi_*}\Big[G_t|S_t = s, A_t = a\Big] \\
&= \mathbb{E}_{\pi_*}\Big[R_{t+1} + \gamma G_{t+1}|S_{t+1} = s', A_t = a\Big] \\
&= r(s,a) + \gamma \sum_{s'} p(s'|s,a) \sum_{a'} q_*(a',s')\pi_*(a'|s')
\end{aligned}
$$

■