Attempts to exercise in Reinforcement Learning book Chapter 3

Mengliao Wang

July 3, 2017

Exercise 3.1:

- Example 1: Chess game. The goal is to beat the opponent in a classic chess game, and the *agent* would be the player. The *state* is the current board, and the *action* is each play the player makes. There is only one reward at the end of the game, which is win, lose, or draw. The limitation is how to evaluate the current board to determine how are we doing. Another limitation is that we do not receive continuous reward, but only one reward at the end. Thus it is harder to evaluate the value of each state. Eventually We need to learn the policy to map each board status to a specific action, i.e. move that would maximize the final reward.
- Example 2: Investment in stock market. Here the *agent* would be the investor, and the *state* could be many different sources of information. The most important one of course is the current market price for each stock, but we can also utilize other information like the number of trades happened, calls made, exchange rates, etc. The *action* is the buy/sell/hold on any specific stock. Lastly, *reward* is the profit we gained/lost after the action, which we need to maximize on a long term. There are a few limitations: Firstly, the transactions happen on real-time, so the model is not discrete unless we split it into small intervals in sacrifice for real-time accuracy. Secondly, We have limited information of the states, because the stock prices do not truely represent the environment. Also the prices is the source of reward, which might introduce confusion and potentially data correlation. Lastly, the rewards are highly nonstationary without a straight forward pattern/model to follow, which might make the policy learning process extremely difficult.
- Example 3: Auto driving system. Here the *agent* of course if the driver, and the *state* would be many sensors on real word, like the speed, direction, height, other cars on the road, etc. The *action* can be speed up/down, or turning the wheel, or turn on/off signals. The *reward* is mostly about safety, which is whether we can reach the destination without any accident. But other factors can be taken into consideration as part of the reward as well, such as time taken to finish, comfort measured by numbers of sudden break, etc.

Exercise 3.2:

No. There are cases that a goal-directed learning task cannot be represented by this framework, especially for the tasks that are sophisticated to evaluate the goal. For example, when we try to make a good painting, we can define the agent, action, environment clearly. We can also define the state to be the pixels on the painting. However there is not a good way to determine the reward. We cannot decide how good or bad a painting is in a measurable way.

Exercise 3.3:

To separate the environment from the agent, the general rule we follow is that anything that cannot be changed arbitrarily by the agent is considered to be outside of it and thus part of its environment. Thus in this case anything outside of the car should be considered as environment, such as road, other vehicles, road

block, etc. Anything that can be directly touched and manipulated inside the car should be considered as part of the agent, such as the break, steering wheel, accelerator, signals, etc. Lastly, regarding the parts on the car that are indirectly controlled by the driver, they should be classified as either agent or environment depending on the possibility that the this indirect control would not be functioning as expected. For example, if we do not take things that might make wheel losing control such as driving on the ice into consideration, then the wheels should be part of the agent as well, and vice versa.

Exercise 3.4:

If we treat this pole-balancing task also as episodic but with the same reward, then given K is the number of time steps before failure the return would be $-\gamma^K$. The return looks the same as continuing formula, but after reaching failure the time step will be reset to 0, instead of continuing at K+1.

Exercise 3.5:

If we use formula 3.1 $G_t = R_{t+1} + R_{t+2} + ...R_T = 1$ as the way to calculate the return where $R_T = 1, R_i = 0, \forall i \neq T$, then we have not communicated to the program that we want to escape from the maze as soon as possible, because no matter what T is the return is alway 1. Instead we can use discounted reward expectation as the return, i.e. $G_t = R_{t+1} + \gamma R_{t+2} + ...\gamma^{T-t-1} R_T = \gamma^{T-t-1}, 0 < \gamma < 1$. In this way, the sooner robot escapes the maze, the bigger return would be (T will be less, so γ^{T-t-1} will be bigger). Another alternative is to set the rewards to be 0 from escaping the maze, and a reward of -1 for all other states.

Exercise 3.6: Broken Vision System

Yes after seeing the first scene we already have access to the Markov state of the environment. If the camera was broken and not receiving any image for a whole day, then we would not have access to the Markov state since the information is independent with the states in the history.

Exercise 3.7:

By definition of $q_\pi(s, a)$:

$$q_\pi(s,a) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a] \tag{1}$$

$$= \mathbb{E}_\pi[R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \mid S_t = s, A_t = a] \tag{2}$$

$$= r(s,a) + \gamma \sum_{s',r} p(s', r \mid s, a)[\sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \mid S_{t+1} = s', A_t = a] \tag{3}$$

$$= r(s,a) + \gamma \sum_{s',r} p(s', r \mid s, a) \sum_{a'} \pi(a' \mid s')[\sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \mid S_{t+1} = s', A_{t+1} = a'] \tag{4}$$

$$= r(s,a) + \gamma \sum_{s',r} p(s', r \mid s, a) \sum_{a'} \pi(a' \mid s') q_\pi(s', a') \tag{5}$$

Exercise 3.8:

Suppose the index starts from left-top corner from 0 to 4, and we denote value of the ith row and jth row as $v_{i,j}$, then we have:

$$v_{2,2} = \sum_a \pi(a|s_{2,2}) \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_\pi(s')] \tag{6}$$

$$= \frac{1}{4}[0 + \gamma v_{1,2}] + \frac{1}{4}[0 + \gamma v_{2,3}] + \frac{1}{4}[0 + \gamma v_{3,2}] + \frac{1}{4}[0 + \gamma v_{2,1}] \tag{7}$$

$$= 0.25 \times 0.9 \times 2.3 + 0.25 \times 0.9 \times 0.4 + 0.25 \times 0.9 \times -0.4 + 0.25 \times 0.9 \times 0.7 \tag{8}$$

$$= 0.7 \tag{9}$$

Exercise 3.9:

If we add a constanct c to reward R, then the new return G'_t would be:

$$G'_t = \sum_{k=0}^{\infty} \gamma^k (R_{t+k+1} + c) \tag{10}$$

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} + \sum_{k=0}^{\infty} c\gamma^k \tag{11}$$

$$= G_t + \sum_{k=0}^{\infty} c\gamma^k \tag{12}$$

$$= G_t + \frac{c}{1-\gamma} \tag{13}$$

So the constant $v_c = \dfrac{c}{1-\gamma}$

Exercise 3.10:

It will change the task if we add a constanct c to rewards in an episodic task. Here we have new return:

$$G'_t = \sum_{k=0}^{T-t-1} \gamma^k (R_{t+k+1} + c) \tag{14}$$

$$= \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1} + \sum_{k=0}^{T-t-1} c\gamma^k \tag{15}$$

$$= G_t + c\frac{1-\gamma^{T-t-1}}{1-\gamma} \tag{16}$$

Here T is the termination time step. Thus we can see the reward for each time step has been modified differently, and we are introducing higher reward for earlier time steps than the original reward, which changes the task goal.

Exercise 3.11:

The two equations are: $v_\pi(s) = \mathbb{E}_\pi[q_\pi(s, a) \mid S_t = s] = \sum_a \pi(a \mid s) q_\pi(s, a)$.

Exercise 3.12:

Similarily, the two equations are $q_\pi(s, a) = \mathbb{E}[R_{t+1} + \gamma v_\pi(s') \mid S_t = s, A_t = a, S_{t+1} = s'] = r(s, a, s') + \gamma \sum_{s',r} p(r, s' \mid s, a) v_\pi(s')$

Exercise 3.13:

The optimal policy would be to use putter on the green, and use driver whenever outside of the green. So the optimal value function would have -1 on the whole green, and -2 and above outside of the green, depending on the shortest distance $|S|$ from the location to the green. Formally, given the max range of a hit with driver is m, then we have the optimal value function:

$$v(s) = \begin{cases} -1, & \text{if } green \\ -[ceil(\frac{|S|}{m}) + 1], & \text{otherwise} \end{cases}$$

Exercise 3.14:

On the green $q_*(s, putter)$ will be always -1, since we can hit the target with one put. Outside of the green within the max range of putter to the green, $q_*(s, putter)$ is -2 since we can hit green after one put. Between the contour of -2 we defined, and the put range in addition to the -2 contour in Figure 3.6 for $q_*(s, driver)$, $q_*(s, putter)$ is -3. Then between the contour of -3 we defined, and the put range in addition to the -3 contour in Figure 3.6 for $q_*(s, driver)$, $q_*(s, putter)$ is -4.

Exercise 3.15:

According to the formula of optimal action function, we have:

$$\begin{aligned}
q_*(h, s) &= p(h \mid h, s)[r_s + \gamma \max_{a'} q_*(h, a')] + p(l \mid h, s)[r_s + \gamma \max_{a'} q_*(l, a')] && (17)\\
&= \alpha[r_s + \gamma \max_{a'} q_*(h, a')] + (1 - \alpha)[r_s + \gamma \max_{a'} q_*(l, a')] && (18)\\
q_*(h, w) &= p(h \mid h, w)[r_w + \gamma \max_{a'} q_*(h, a')] + p(l \mid h, w)[r_w + \gamma \max_{a'} q_*(l, a')] && (19)\\
&= r_w + \gamma \max_{a'} q_*(h, a') && (20)\\
q_*(l, s) &= p(h \mid l, s)[-3 + \gamma \max_{a'} q_*(h, a')] + p(l \mid l, s)[r_s + \gamma \max_{a'} q_*(l, a')] && (21)\\
&= (1 - \beta)[-3 + \gamma \max_{a'} q_*(h, a')] + \beta[r_s + \gamma \max_{a'} q_*(l, a')] && (22)\\
q_*(l, w) &= p(h \mid l, w)[r_w + \gamma \max_{a'} q_*(h, a')] + p(l \mid l, w)[r_w + \gamma \max_{a'} q_*(l, a')] && (23)\\
&= r_w + \gamma \max_{a'} q_*(l, a') && (24)\\
q_*(l, r) &= p(h \mid l, r)[0 + \gamma \max_{a'} q_*(h, a')] + p(l \mid l, r)[0 + \gamma \max_{a'} q_*(l, a')] && (25)\\
&= \gamma \max_{a'} q_*(h, a') && (26)\\
& && (27)
\end{aligned}$$

Exercise 3.16:

If we mark the optimal value of the cell at ith row and jth column as $v_*(i,j), i,j = 0,1,2,3,4$. Then according to formula 3.17 we have

$$\begin{aligned} v_*(A) &= r(A', A) + \gamma v_*(A') && (28)\\ &= r(A', A) + \gamma[0 + \gamma v_*(3,1)] && (29)\\ &= r(A', A) + \gamma^2[0 + \gamma v_*(2,1)] && (30)\\ &= r(A', A) + \gamma^3[0 + \gamma v_*(1,1)] && (31)\\ &= r(A', A) + \gamma^5 v_*(A) && (32)\\ &= 10 + 0.9^5 v_*(A) && (33)\\ & && (34) \end{aligned}$$

From this equation we can easily have $v_*(A) = 24.419$.