RL Book Chapter 3 Solutions

Hariharan Sezhiyan

November 2019

Exercise 3.6

The return at each time is still $-y^K$, and there is no difference between this return and that of the continuing, discounted formulation. The pole task is interesting because it can be easily formulated as both a continuous and episodic task. It is episodic in the sense that there is a terminal state (the pole being out of balance), but that terminal state is not necessarily reached (in the case of successful balancing).

Exercise 3.7

It is likely the agent has not explored enough of the state space to know that exiting the maze results in a +1 reward. This lack of exploration can be the result of the agent being stuck in a local optimum in its loss function. On a simpler level, the designer might not have introduced enough randomness into the agent to allow it to explore beyond its greedy policy. The agent might always be selecting actions based on this policy, but not exploring to find possibly better policies.

Exercise 3.8

$G_5 = 0$
$G_4 = R_5 + \gamma * G_5 = 2$
$G_3 = R_4 + \gamma * G_4 = 4$
$G_2 = R_3 + \gamma * G_3 = 8$
$G_1 = R_2 + \gamma * G_2 = 6$
$G_0 = R_1 + \gamma * G_1 = 2$

Exercise 3.9

$G_1 = \frac{7}{1-0.1} = 70$
$G_0 = 2 + \gamma * G_1 = 65$

Exercise 3.10

$s = 1 + y + y^2 + y^3 + ...$
$y * s = y + y^2 + y^3 + ...$
$s * (1 - y) = 1$
$s = \frac{1}{1-y}$

Exercise 3.11

$$E[R_{t+1}] = \sum_{r \in R} r \sum_{s \in S} \sum_{a \in A} \pi(a|s) p(s', r|s, a)$$

Exercise 3.12

$$v_\pi(s) = \sum_{a \in A} q_\pi(s, a) \pi(a|s)$$

Exercise 3.13

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

Exercise 3.14

We know that the policy is random, so every action is taken with probability 0.25. Furthermore, there is a discounting factor of $\gamma = 0.9$. Using the formulas in exercises 3.12 and 3.13, we have $0.25 * 0.9 * [2.3 + 0.7 + 0.4 - 0.4] = 0.675$.

Exercise 3.15

The Bellman equation for the value of a state can be written as:

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

If a constant c is added to each reward, then the term $r + \gamma v_\pi(s')$ becomes $r + c + \gamma v_\pi(s')$. However, this not only affects the current state s, but all successor states s', which involves a product of γ. The total change in value this becomes:

$$v_c = c + \gamma * c + \gamma^2 * c + \gamma^3 * c... = \frac{c}{1 - \gamma}$$

Exercise 3.16

If the episodic task does not have a discounted reward system, the sum of the rewards will go to infinity.

Exercise 3.17

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

Exercise 3.18

The value of a state following a policy can be determined by the expected state-action value. The value of q(s,a) already takes into account the reward of the subsequent step, so there is no need to include an expected reward.

$$v(s) = E[q_\pi(s,a)|\pi] = \sum_{a \in A} q_\pi(s,a)\pi(a|s)$$

Exercise 3.19

The state-action value is simply the sum of the expected reward and the expected value of the subsequent state: $q_\pi(s,a) = E[R_{t+1} + V_\pi(S_{t+1}]$. Taking into account the dynamics of the environment, this is equivalent to exercise 3.13:

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

Exercise 3.20

First, it is important to identify the optimal policy: two drives followed by a single putt into the hole. Taking this into account, $v*$ would identical to the diagram for $q*(s,a)$. The tee would have a value of -3, and a single drive from this position would land the golf ball at an intermediate location, with state value of -2. From here, the optimal action would be to again drive into the green, landing at a location with -1 value. From there, a single putt would end the game.

Exercise 3.21

For this situation, the optimal policy (after the initial putt action) will be 2 drives followed by a single putt. That's a total of 4 actions to terminate the episode. The outermost contour will have state-value -4, with the 2 subsequent having state-value -3 and -2. The innermost contour (within the green section), will have state-value -1.

Exercise 3.22

For $\gamma = 0$, the π_{left} is optimal because π_{right} will lead to 0 rewards. After the initial reward of 0, all actions will be discounted to 0.

For $\gamma = 0.99$, π_{right} is optimal:

$$E[\pi_{left}] = 1 + 0.99^2 + 0.99^4 + ... = \frac{1}{1 - 0.99^2} = \ 50.25$$

$$E[\pi_{right}] = 2 * 0.99 + 2 * 0.99^3 + ... = \frac{2 * 0.99}{1 - 0.99^2} = \ 90.60$$

For $\gamma = 0.5$, both policies give equal returns and are both optimal:

$$E[\pi_{left}] = 1 + 0.5^2 + 0.5^4 + ... = \frac{1}{1 - 0.5^2} = \ 1.33$$

$$E[\pi_{right}] = 2 * 0.5 + 2 * 0.5^3 + ... = \frac{2 * 0.5}{1 - 0.5^2} = \ 1.33$$

3.23

I'll provide the value for $q * (h, w)$, but the same logic applies to the other 3 combinations.

$$q*(h,w) = max([p(h|h,w)[r(h,w,h)+\gamma q*(h,w)], [p(h|h,s)[r(h,w,s)+\gamma q*(h,s)])$$

$$max([(1)(r_{wait} + \gamma * q * (h, w)], [\alpha * (r_{search} + \gamma q * (h, s))])$$

3.24

The optimal policy involves choosing any action in state A (which will lead to state A'), and then always choosing the "up" action until the agent again reaches state A. The first reward will be +10, but every subsequent action will be discounted ($\gamma = 0.9$). The next time the agent reaches state A, the action will be discounted by 0.9 with a factor of 5. The value of state A can be written as:

$$v_(A) = 10 + 10 * 0.9^5 + 10 * 0.9^{10} + ... = \frac{10}{1 - 0.9^5} = 24.412$$

3.25

$$v_*(s) = max[q_*(s, a)]$$

3.26

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

3.27

Instead of the optimal state-action value itself, we are looking for the action that will yield this optimal state-action value. Hence, we use an argmax.

$$\pi_*(s) = argmax_a[q_*(s,a)]$$

3.28

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