RL Book Chapter 4 Solutions

Hariharan Sezhiyan

November 2019

Exercise 4.1

From chapter 3, we know

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

We also know the equiprobable policy is deterministic, so $p(s',r|s,a) = 1, \forall s,a$ and that $r(s,a) = -1, \forall s,a$. Thus:

$$q_\pi(11, down) = 1 * (-1 + 0) = -1$$
$$q_\pi(7, down) = 1 * (-1 + (-14)) = -15$$

Exercise 4.2

For the first case:

$$v(s_{15}) = \frac{1}{4}[q(s_{15}, left) + q(s_{15}, right) + q(s_{15}, up) + q(s_{15}, down)]$$

.
We know each state transition is deterministic and $q(s_{15}, left)$ = -1-22, $q(s_{15}, right)$ = -1-14, etc, so this becomes:

$$v(s_{15}) = \frac{1}{4}[-1 - 22 - 1 - 20 - 1 - 14 - 1 + v(s_{15})]$$
$$3v(s_{15}) = -60; v(s_{15}) = -20$$

For the second case, the value of the state 15 remains at -20. In the previous problem, we calculated the value of state 15 to be -20, the same as the value of state 13. The value of state 13 doesn't change because the successor state for taking the down action (state 15) has the same value as before (state 13). Since the value of state 13 stays the same, all successor state values for state 15 remain the same, so it's value doesn't change.

Exercise 4.3

Analogy to eq 4.3: $q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1})|S_t = s, A_t = a]$

Analogy to eq 4.4: $q_\pi(s,a) = \sum_{s',r} p(s',r|s,a) * (r + \sum_{a'} \pi(a'|s') q_\pi(s',a'))$

Analogy to eq 4.6: $q_{k+1}(s,a) = \sum_{s',r} p(s',r|s,a) * (r + \sum_{a'} \pi(a'|s') q_k(s',a'))$

Exercise 4.4

The bug is in the argmax over the action indices of the sum of the expected returns of successor states starting with an action and following policy (line 4 in the policy improvement section). Instead of blindly using an argmax (which only returns the index of the first occurrence of the largest value in a list or dict), we need to return indices of all occurrences of the maximum. The following numpy code can be used:

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

action-maxes = np.argwhere(return-maxes == np.amax(return-maxes))

if old-action not in action-maxes, then policy-stable = false

Exercise 4.5

The change to account for states and actions will occur in all three parts: the initialization, the policy evaluation, and the policy improvement sections. First, in the initialization step, we need to initialize q(s,a) for all $s \in S$ and $a \in A$. In the policy evaluation section, we will now need to iterate over all states and the actions within them. We will need to obtain the current $q(s,a)$ estimate as well as the new one from the formula $q_\pi(s,a) = \sum_{s',r} p(s',r|s,a) * (r + \sum_{a'} \pi(a'|s') q_\pi(s',a'))$, and estimate the error of estimation until it converges.

Exercise 4.6

The initialization step (1) will need to be modified such that the policy is randomly initialized from the new minimum to the origin maximum value.
In the policy evaluation step, one would need to consider the situation in which 2 or more actions can be selected from the policy (since both have equal probability of being selected). In this case, the value could be the average of the evaluation of both these actions.
In the policy improvement step, like in the evaluation step, the policy could output 2 actions. To ensure you don't run into the bug of the agent outputting two equally good actions, you need to check for both actions when calculating the old-action (in the pseudo code in the book). If the new action is contained in one of the old-actions, the algorithm should terminate.

Exercise 4.8

Since there is no discounted reward, the agent has no incentive to finish the game as soon as possible. Instead, there are certain points (namely 50 and 75) where agent goes all in to win. When the agent has 51 capital, it bets a single dollar because winning will slightly increase returns (and value), but losing will put it at 50 capital and a position to go all in. The agent is thus able to go to two good states regardless of a win or lose at position 51.

Exercise 4.9

The code for this exercise is found in the file exercise-4.9.py. The associated images are found in the /img folder.

Exercise 4.10

$$q_{k+1}(s,a) = \max_{a'} \sum_{s',r} p(s',r|s,a) * (r + \gamma q_k(s',a'))$$