Róbert Leó Jónsson (robertt20@ru.is))

Written Exercises: Chapter 4

Exercise 4.1

We are not using any discount, so we have

$$\begin{aligned} q_\pi(11, down) &= r + v_\pi(T) && (1)\\ &= -1 + 0 = -1 && (2)\\ q_\pi(7, down) &= r + v_\pi(11) && (3)\\ &= -1 + -14 = -15 && (4) \end{aligned}$$

Exercise 4.2

Adding the new state, we get

$$\begin{aligned} v_\pi(15) &= \sum_a \pi(a \mid 15)(-1 + v_\pi(s')) && (5)\\ &= 0.25 \cdot (-1 + v_\pi(13)) + 0.25 \cdot (-1 + v_\pi(12)) + 0.25 \cdot (-1 + v_\pi(14)) + 0.25 \cdot (-1 v_\pi(15)) && (6)\\ &= 0.25(-4 + v_\pi(13) + v_\pi(12) + v_\pi(14) + v_\pi(15) && (7)\\ &= -1 + 0.25(-20 - 22 - 14 + v_\pi(15)) && (8)\\ &= -1 - 14 + 0.25 v_\pi(15) && (9)\\ \implies 0.75 v_\pi(15) &= -15 && (10)\\ \implies v_\pi(15) &= \frac{-15}{0.75} = -20 && (11) \end{aligned}$$

Now, if we change the dynamics such that moving down in state 13 will move us to 15, the value function will not change. This is because $v_\pi(13) = v_\pi(15)$, so $q_\pi(13, down) = -20$ hold regardless of whether the move will take us to 15 or not.

Exercise 4.3

The equations analogous to (4.3) and (4.4) are given in Exercise 3.17.

The update equation for $q_\pi(s, a)$ is given by

$$\begin{aligned} q_{k+1}(s, a) &= \mathbb{E}_\pi \left[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a\right] && (12)\\ &= \sum_{s', r} p(s', r \mid s, a) \left(r + \gamma \sum_{a'} \pi(a' \mid s) q_k(a', s) \right) && (13) \end{aligned}$$

Exercise 4.4

We need to loop through the computed value function to check whether they are equal; two optimal policies will have the same value function.

Exercise 4.5

The process is very similar - we use the Bellman equation for q_π to evaluate the policy.

To improve the policy, we again create the greedy policy π' with respect to q_π. That is,

```
(1) Initialization
      $q(s,a) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$.
(2) Policy evaluation
      Loop:
      • $\Delta \leftarrow 0$
      • Loop for each $s \in \mathcal{S}$ and each $a \in \mathcal{A}(s)$ :
          – $q \leftarrow q(s,a)$
          – $q(s,a) \leftarrow \sum_{s',r} p(s', r \mid s, a)\,(r + \gamma q(s', \pi(s')))$
          – $\Delta \leftarrow \max(\Delta, |q - q(s,a)|$
      • until $\Delta < \theta$
(3) Policy Improvement
      • $\textit{policy-stable} \leftarrow \textit{true}$
      • For each $s \in \mathcal{S}$:
          – $\textit{old-action} \leftarrow \pi(s)$
          – $\pi(s) \leftarrow \operatorname{argmax}_a \sum_{s',r} q(s,a)$
          – If $\textit{old-action} \neq \pi(s)$ then $\textit{policy-stable} \leftarrow \textit{false}$.
      • If $\textit{policy-stable}$, then stop and return $q \approx q_*$ and $\pi \approx \pi_*$, else go to 2.
```

Exercise 4.6

We assume that there would be a $1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(s)|}$ probability on the optimal action, and the remaining $\epsilon - \frac{\epsilon}{|\mathcal{A}(s)|}$ probability would be evenly spread over the remaining actions.

Now, assume that we store this action in $A(s)$. Then,

$$\pi_A(a \mid s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(s)|} & \text{if } a = A(s) \\ \epsilon - \frac{\epsilon}{|\mathcal{A}(s)|} & \text{otherwise} \end{cases} \tag{14}$$

Then, step 3 would be

```
• $\textit{policy-stable} \leftarrow \textit{true}$
• For each $s \in \mathcal{S}$ :
    – $\textit{old-action} \leftarrow A(s)$
    – $A(s) \leftarrow \operatorname{argmax}_{a'} \sum_a \pi'_a(a \mid s) \sum_{s',r} p(s', r \mid s, a)\,[r + \gamma V(s')]$
    – $\pi(s) \leftarrow \pi_A(s)$
    – If $\textit{old-action} \neq A(s)$ then $\textit{policy-stable} \leftarrow \textit{false}$
• $\cdots$
```

For step two, the value update would simply be

$$V(s) \leftarrow \sum_a \pi_A(s)(a \mid s) \sum_{s',r} p(s', r \mid s, a') \left[r + \gamma V(s')\right] \tag{15}$$

where the rest of the step remains the same.

Step one would consist of letting $V(s) \in \mathbb{R}$ arbitrarily, and setting $A(s)$ for $s \in \mathcal{S}$ arbitrarily, with $\pi(a \mid s) = \pi_{A(s)}(a \mid s)$ as defined above.

Exercise 4.7

See the code directory.

Exercise 4.8

This is a tough one. Making a lot of small bets seems to be less likely to win you the game as making a few large ones, as the game is against you with $p_h = 0.4$. So to win the game, it might be a better strategy to make large bets. When you have 50 in capital, it is wise to bet it all, as it is more likely that you will reach a 100 with that strategy than by making many small bets.

However, if you have 51, it does not pay to bet it all, as you can make a bet of 1, and you might win that bet. If you lose the bet, you might still win in the all-in strategy.

The pyramids indicate that we are willing to bet exactly the amount that, if we lose, will bounce us back to 50.

The general pattern seems to be that we want to recursively split the capital line into halves, and make jumping points.

The guess is that any optimal policy will have a jumping point in the middle, and then each split will either have a pyramid, or a recursion, with a jumping point in the middle.

Exercise 4.9

See code.

Exercise 4.10

It is

$$\begin{aligned} q_{k+1}(s,a) &= \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q_k(S_{t+1}, a) \mid S_t = s, A_t = a\right] && (16) \\ &= \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma \max_{a'} q_k(s', a')\right] && (17) \end{aligned}$$