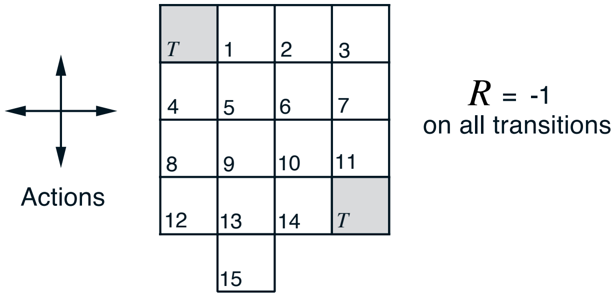
Answers to Exercises
Reinforcement Learning: Chapter 4

Exercise 4.1 In Example 4.1, if π is the equiprobable random policy, what is $q_\pi(11, \texttt{down})$? What is $q_\pi(7, \texttt{down})$?

Answer: $q_\pi(11, \texttt{down}) = -1$. $q_\pi(7, \texttt{down}) = -15$. □

Exercise 4.2 In Example 4.1, suppose a new state 15 is added to the gridworld just below state 13, and its actions, `left`, `up`, `right`, and `down`, take the agent to states 12, 13, 14, and 15, respectively. Assume that the transitions *from* the original states are unchanged. What, then, is $v_\pi(15)$ for the equiprobable random policy? Now suppose the dynamics of state 13 are also changed, such that action `down` from state 13 takes the agent to the new state 15. What is $v_\pi(15)$ for the equiprobable random policy in this case?



T	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	T
	-20.		

Answer: In the case where none of the other states have their outgoing transitions changed, then the new state's value under the random policy is

$$\begin{aligned} v_\pi(15) &= \mathbb{E}_\pi[R_{t+1} + v_\pi(S_{t+1}) \mid S_t = s] \\ &= -1 + \frac{1}{4}v_\pi(12) + \frac{1}{4}v_\pi(13) + \frac{1}{4}v_\pi(14) + \frac{1}{4}v_\pi(15) \end{aligned}$$

Plugging in the asymptotic values for $v_\infty = v_\pi$ for states 12, 13, and 14 from Figure 4.1 (and above, right) and solving for $v_\pi(15)$ yields

$$v_\pi(15) = -1 - \frac{1}{4}22 - \frac{1}{4}20 - \frac{1}{4}14 - \frac{1}{4}v_\pi(15)$$

$$v_\pi(15)\left(1 - \frac{1}{4}\right) = -15$$

$$v_\pi(15) = -20$$

If the dynamics of state 13 also change, then it turns out that the answer is the same! This can be most easily seen by hypothesizing that $v_\pi(15) = -20$ and then checking that all states still satisfy the Bellman equation for v_π. □

Exercise 4.3 What are the equations analogous to (4.3), (4.4), and (4.5) for the action-value function q_π and its successive approximation by a sequence of functions $q_0, q_1, q_2, \ldots$?

Answer:

$$\begin{aligned} q_\pi(s,a) &= \mathbb{E}_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a] && (4.3) \\ &= \sum_{s',r} p(s',r \,|\, s,a) \left[r + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a') \right] && (4.4) \end{aligned}$$

$$\begin{aligned} q_{k+1}(s,a) &= \mathbb{E}_\pi[R_{t+1} + \gamma q_k(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a] \\ &= \sum_{s',r} p(s',r \,|\, s,a) \left[r + \gamma \sum_{a'} \pi(a'|s') q_k(s',a') \right] && (4.5) \end{aligned}$$

□

Exercise 4.4 The policy iteration algorithm on page 80 has a subtle bug in that it may never terminate if the policy continually switches between two or more policies that are equally good. This is ok for pedagogy, but not for actual use. Modify the pseudocode so that convergence is guaranteed.

Answer: TBD. □

Exercise 4.5 How would policy iteration be defined for action values? Give a complete algorithm for computing q_*, analogous to that on page 80 for computing v_*. Please pay special attention to this exercise, because the ideas involved will be used throughout the rest of the book.

Answer: Just as for state values, we would have an alternation of policy improvement and policy evalution steps, only this time in q rather than in v:

$$\pi_0 \xrightarrow{\text{PE}} q_{\pi_0} \xrightarrow{\text{PI}} \pi_1 \xrightarrow{\text{PE}} q_{\pi_1} \xrightarrow{\text{PI}} \pi_2 \xrightarrow{\text{PE}} \cdots \xrightarrow{\text{PI}} \pi_* \xrightarrow{\text{PE}} q_*$$

Each policy evaluation step, $\pi_i \xrightarrow{\text{PE}} q_{\pi_i}$, would involve multiple iterations of equation (4.5) above, until convergence, or some other way of computing q_{π_i}. Each policy improvement step, $q_{\pi_i} \xrightarrow{\text{PI}} \pi_{i+1}$, would be a greedification with respect to q_{π_i}, i.e.:

$$\pi_{i+1}(s) = \arg\max_a q_{\pi_i}(s,a).$$

A boxed algorithm for policy iteration to q_* is:

Policy iteration for action values (using iterative policy evaluation)

```
1. Initialization
   $Q(s,a) \in \mathbb{R}$ arbitrarily for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$
   $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Repeat
       $\Delta \leftarrow 0$
       For each $s \in \mathcal{S}$ and $a \in \mathcal{A}$:
           $q \leftarrow Q(s,a)$
           $Q(s,a) \leftarrow \sum_{s',r} p(s',r \mid s,a)\Big[r + \gamma Q(s',\pi(s'))\Big]$
           $\Delta \leftarrow \max(\Delta, |q - Q(s,a)|)$
   until $\Delta < \theta$  (a small positive number)

3. Policy Improvement
   *policy-stable* $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
       $a \leftarrow \pi(s)$
       $\pi(s) \leftarrow \arg\max_a Q(s,a)$
       If $a \neq \pi(s)$, then *policy-stable* $\leftarrow false$
   If *policy-stable*, then stop and return $Q$ and $\pi$; else go to 2
```

In the "arg max" step, it is important that ties be broken in a consistent order. □

Exercise 4.6 Suppose you are restricted to considering only policies that are *ε-soft*, meaning that the probability of selecting each action in each state, s, is at least $\varepsilon/|\mathcal{A}(s)|$. Describe qualitatively the changes that would be required in each of the steps 3, 2, and 1, in that order, of the policy iteration algorithm for v_* (page 80).

Answer: Step 3, the policy improvement step, would have to be changed such that the new policy is not the deterministic greedy policy, but the closest ϵ-soft policy. That is, all non-greedy actions would be given the minimal probability, $\frac{\epsilon}{|\mathcal{A}(s)|}$, and all the rest of the probability would go to the greedy action. The check for termination would also need to be changed. Somehow we would have to check for a change in the action with the bulk of the probability.

Step 2, policy evalution, would need to be generalized to accomodate stochastic policies. A new equation analogous to (4.5) would be needed.

Step 1, initialization, would need be changed only to permit the initial policy to be stochastic. □

Exercise 4.8 Why does the optimal policy for the gambler's problem have such a curious form? In particular, for capital of 50 it bets it all on one flip, but for capital of 51 it does not. Why is this a good policy?

Answer: In this problem, with $p = 0.4$, the coin is biased against the gambler. Because of this, the gambler want to minimize his number of flips. If he makes many small bets he is likely to lose. Thus, with a stake of 50 he can bet it all and have a 0.4 probability of winning. On the other hand, with stake of 51 he can do slightly better. If he bets 1, then even if he loses he still has 50 and thus a 0.4 chance of winning. And if he wins he ends up with 52. With 52 he can bet 2 and maybe end up with 54 etc. In these cases there is a chance he can get up to 75 without ever risking it all on one bet, yet he can always fall back (if he loses) on one big bet. And if he gets to 75 he can safely bet 25, possibly winning in one, while still being able to fall back to 50. It is this sort of logic which causes such big changes in the policy with small changes in stake, particularly at multiples of the negative powers of two. □

Exercise 4.10 What is the analog of the value iteration update (4.10) for action values, $q_{k+1}(s,a)$?

Answer: Value iteration in action values is defined by

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

for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$. For arbitrary q_0, the sequence $\{q_k\}$ converges to q_*. □