Answers to Exercises
Reinforcement Learning: Chapter 3

The first three exercises are meant simply to be thought provoking and do not have specific answers.

Exercise 3.1 Devise three example tasks of your own that fit into the MDP framework, identifying for each its states, actions, and rewards. Make the three examples as *different* from each other as possible. The framework is abstract and flexible and can be applied in many different ways. Stretch its limits in some way in at least one of your examples.

Answer: This exercise is marked primarily on the basis of your identifying the states, actions and rewards for each example. □

Exercise 3.2 Is the MDP framework adequate to usefully represent *all* goal-directed learning tasks? Can you think of any clear exceptions?

Answer: Any thoughtful answer will do here. I suppose there are exceptions, but the framework seems relevant to almost all the interesting cases. □

Exercise 3.3 Consider the problem of driving. You could define the actions in terms of the accelerator, steering wheel, and brake, that is, where your body meets the machine. Or you could define them farther out—say, where the rubber meets the road, considering your actions to be tire torques. Or you could define them farther in—say, where your brain meets your body, the actions being muscle twitches to control your limbs. Or you could go to a really high level and say that your actions are your choices of *where* to drive. What is the right level, the right place to draw the line between agent and environment? On what basis is one location of the line to be preferred over another? Is there any fundamental reason for preferring one location over another, or is it a free choice?

Answer: I would say that this is largely a free choice, dependent on what level one wishes to address the problem, what one wants to treat as completely controllable. □

Exercise 3.4 Give a table analogous to that in Example 3.3, but for $p(s', r \,|\, s, a)$. It should have columns for s, a, s', r, and $p(s', r \,|\, s, a)$, and a row for every 4-tuple for which $p(s', r \,|\, s, a) > 0$.

Answer:

s	a	s'	r	$p(s', r \,\|\, s, a)$
high	search	high	$r_{\texttt{search}}$	α
high	search	low	$r_{\texttt{search}}$	$1-\alpha$
low	search	high	-3	$1-\beta$
low	search	low	$r_{\texttt{search}}$	β
high	wait	high	$r_{\texttt{wait}}$	1
high	wait	low	$r_{\texttt{wait}}$	0
low	wait	high	$r_{\texttt{wait}}$	0
low	wait	low	$r_{\texttt{wait}}$	1
low	recharge	high	0	1
low	recharge	low	0	0

□

Exercise 3.5 The equations in Section 3.1 are for the continuing case and need to be modified (very slightly) to apply to episodic tasks. Show that you know the modifications needed by giving the modified version of (3.3).

Answer: The sum over all states $\mathcal{S}$ must be modified to a sum over all states $\mathcal{S}^+$. The same is true for (3.5). □

Exercise 3.6 Suppose you treated pole-balancing as an episodic task but also used discounting, with all rewards zero except for -1 upon failure. What then would the return be at each time? How does this return differ from that in the discounted, continuing formulation of this task?

Answer: In this case the return at each time would be $-\gamma^{k-1}$ where k is the number of steps until failure. This differs from the continuing formulation of the task in that what happens on the following episode is ignored. For example, suppose the current policy always avoided failure for exactly 10 steps from the start state. Then under the episodic formulation the return from the start state would be $-\gamma^{10}$ whereas under the continuing formulation it would be $-\gamma^{10} - \gamma^{20} - \gamma^{30} - \cdots$. □

Exercise 3.7 Imagine that you are designing a robot to run a maze. You decide to give it a reward of +1 for escaping from the maze and a reward of zero at all other times. The task seems to break down naturally into episodes—the successive runs through the maze—so you decide to treat it as an episodic task, where the goal is to maximize expected total reward (3.7). After running the learning agent for a while, you find that it is showing no improvement in escaping from the maze. What is going wrong? Have you effectively communicated to the agent what you want it to achieve?

Answer: Notice that the returns in this task are always +1 no matter how long it takes the agent to exit the maze. Thus there is no incentive in the problem formulation to find short paths: all are equally good. If you want the agent to find short paths, then this formulation does not capture this, and in that sense you have not communicated your desire to the agent. □

Exercise 3.8 Suppose $\gamma = 0.5$ and the following sequence of rewards is received $R_1 = -1$, $R_2 = 2$, $R_3 = 6$, $R_4 = 3$, and $R_5 = 2$, with $T = 5$. What are G_0, G_1, ..., G_5? Hint: Work backwards.

Answer: The trick to making this easy is to use (3.9) and to compute the returns backwards, starting with G_5, then G_4, and so on. The reward sequence is -1, 2, 6, 3, 2, so the returns are

$G_5 = 2$
$G_4 = 3 + 0.5 \cdot G_5 = 4$
$G_3 = 6 + 0.5 \cdot G_4 = 8$
$G_2 = 2 + 0.5 \cdot G_3 = 6$
$G_1 = -1 + 0.5 \cdot G_2 = 2$ □

Exercise 3.9 Suppose $\gamma = 0.9$ and the reward sequence is $R_1 = 2$ followed by an infinite sequence of 7s. What are G_1 and G_0?

Answer: Using (3.10), we get $G_1 = 70$. Using (3.9) we get $G_0 = 2 + 0.9 \cdot G_1 = 65$. □

Exercise 3.10 Prove the second equality in (3.10).

Answer: Because all rewards are 1, all the returns are equal: $G_t = G_{t+1} \doteq G$. Starting from (3.9), then, we have:

$$\begin{aligned} G_t &= R_{t+1} + \gamma G_{t+1} \qquad (3.9) \\ G &= 1 + \gamma G \\ G - \gamma G &= 1 \\ G(1-\gamma) &= 1 \\ G &= \frac{1}{1-\gamma}. \end{aligned}$$

QED. □

Exercise 3.11 If the current state is S_t, and actions are selected according to stochastic policy π, then what is the expectation of R_{t+1} in terms of π and the four-argument function p (3.2)?

Answer:

$$\begin{aligned} \mathbb{E}[R_{t+1} \mid S_t, A_t \sim \pi] &= \sum_a \pi(a|S_t)\mathbb{E}[R_{t+1} \mid S_t, A_t = a] \\ &= \sum_a \pi(a|S_t) \sum_{s',r} p(s', r \,|\, s, a) r \end{aligned}$$

□

Exercise 3.12 Give an equation for v_π in terms of q_π and π.

Answer:

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

□

Exercise 3.13 Give an equation for q_π in terms of v_π and the four-argument p.

Answer:

$$q_\pi(s, a) = \sum_{s \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \,|\, s, a) \left[r + \gamma v_\pi(s') \right]$$

□

Exercise 3.14 The Bellman equation (3.14) must hold for each state for the value function v_π shown in Figure 3.2 (right) of Example 3.5. Show numerically that this equation holds for the center state, valued at +0.7, with respect to its four neighboring states, valued at +2.3, +0.4, −0.4, and +0.7. (These numbers are accurate only to one decimal place.)

Answer:

$$\begin{aligned} v_\pi(s) &= \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \big[r + \gamma v_\pi(s') \big] \qquad (3.14) \\ &= \sum_a 0.25 \cdot \Big[0 + 0.9 \cdot (2.3 + 0.4 - 0.4 + 0.7) \Big] \\ &= 0.25 \cdot [0.9 \cdot 3.0] = 0.675 \approx 0.7 \end{aligned}$$

□

Exercise 3.15 In the gridworld example, rewards are positive for goals, negative for running into the edge of the world, and zero the rest of the time. Are the signs of these rewards important, or only the intervals between them? Prove, using (3.8), that adding a constant c to all the rewards adds a constant, v_c, to the values of all states, and thus does not affect the relative values of any states under any policies. What is v_c in terms of c and γ?

Answer: The values are the expected values of the returns, and the returns are all changed by a constant:

$$G_t = \sum_{k=0}^{\infty} \gamma^k (R_{t+k+1} + c) = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} + \sum_{k=0}^{\infty} \gamma^k c$$

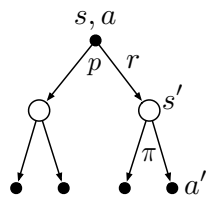
$$v_c = \sum_{k=0}^{\infty} \gamma^k c = c \sum_{k=0}^{\infty} \gamma^k = \frac{c}{1-\gamma}$$

□

Exercise 3.16 Now consider adding a constant c to all the rewards in an episodic task, such as maze running. Would this have any effect, or would it leave the task unchanged as in the continuing task above? Why or why not? Give an example.

Answer: The absolute values of the rewards do matter in the total reward case because they strongly affect the relative values of trajectories of different lengths. For example, consider a maze task where the object is to reach the terminal state in the smallest number of steps. We can arrange for this by setting the reward on each step to be -1. However, if we add the constant $+2$ to all the rewards, then the system obtains a $+1$ reward for every step; instead of finding the shortest way out it will find the longest way to stay in! □

Exercise 3.17 What is the Bellman equation for action values, that is, for q_π? It must give the action value $q_\pi(s,a)$ in terms of the action values, $q_\pi(s',a')$, of possible successors to the state–action pair (s,a). Hint: the backup diagram to the right corresponds to this equation. Show the sequence of equations analogous to (3.14), but for action values.



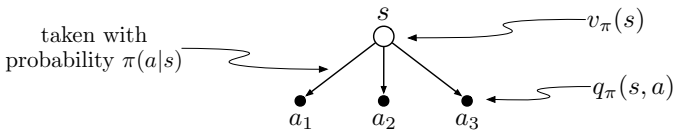
q_π backup diagram

Answer:

$$\begin{aligned}
q_\pi(s,a) &= \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] && \text{(by (3.9))} \\
&= \sum_{s'} \sum_{r} p(s', r \mid s, a) \Big[r + \gamma \mathbb{E}_\pi\big[G_{t+1} \mid S_{t+1} = s'\big] \Big] \\
&= \sum_{s',r} p(s', r \mid s, a) \Big[r + \gamma \sum_{a'} \pi(a'|s') \mathbb{E}_\pi\big[G_{t+1} \mid S_{t+1} = s', A_{t+1} = a'\big] \Big] \\
&= \sum_{s',r} p(s', r \mid s, a) \Big[r + \gamma \sum_{a'} \pi(a'|s') q_\pi(s', a') \Big]
\end{aligned}$$

□

Exercise 3.18 The value of a state depends on the values of the actions possible in that state and on how likely each action is to be taken under the current policy. We can think of this in terms of a small backup diagram rooted at the state and considering each possible action:



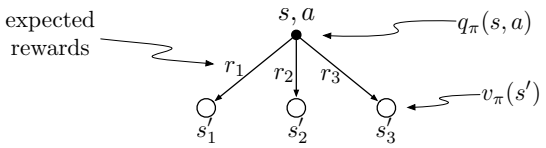
Give the equation corresponding to this intuition and diagram for the value at the root node, $v_\pi(s)$, in terms of the value at the expected leaf node, $q_\pi(s,a)$, given $S_t = s$. This equation should include an expectation conditioned on following the policy, π. Then give a second equation in which the expected value is written out explicitly in terms of $\pi(a|s)$ such that no expected value notation appears in the equation.

Answer:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[q_\pi(S_t, A_t) \mid S_t = s] \\ &= \sum_a \pi(a|s) q_\pi(s,a) \end{aligned}$$

□

Exercise 3.19 The value of an action, $q_\pi(s,a)$, depends on the expected next reward and the expected sum of the remaining rewards. Again we can think of this in terms of a small backup diagram, this one rooted at an action (state–action pair) and branching to the possible next states:



Give the equation corresponding to this intuition and diagram for the action value, $q_\pi(s,a)$, in terms of the expected next reward, R_{t+1}, and the expected next state value, $v_\pi(S_{t+1})$, given that $S_t = s$ and $A_t = a$. This equation should include an expectation but *not* one conditioned conditioned on following the policy. Then give a second equation, writing out the expected value explicitly in terms of $p(s', r \,|\, s, a)$ defined by (3.2), such that no expected value notation appears in the equation.

Answer:

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

□

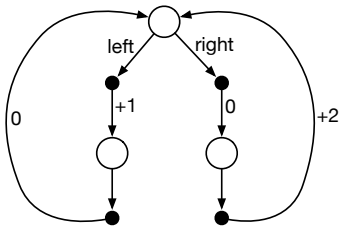
Exercise 3.20 Draw or describe the optimal state-value function for the golf example.

Answer: The optimal state-value function gives the value of each state given optimal behavior. It is -1 anywhere on the green, then -2 up to about midway down the fairway, then -3. In other words, it is exactly the function formed by taking the max of the two functions shown in Figure 3.3. □

Exercise 3.21 Draw or describe the contours of the optimal action-value function for putting, $q_*(s, \texttt{putter})$, for the golf example.

Answer: This function gives the value of each state if we putt once, then afterwards behave optimally. It is -1 anywhere on the green, -2 anywhere within reach of a putt from the green, then -3 up to anywhere from which we can putt to within the -2 contour of $q_*(s, \texttt{driver})$ shown in Figure 3.3. Similarly, it is -4 anywhere from which we can putt to within the -3 contour of $q_*(s, \texttt{driver})$ shown in Figure 3.3. The sand traps are both -3 because the initial putt does nothing and then we drive out of the trap and putt to the hole. □

Exercise 3.22 Consider the continuing MDP shown on to the right. The only decision to be made is that in the top state, where two actions are available, left and right. The numbers show the rewards that are received deterministically after each action. There are exactly two deterministic policies, π_{left} and π_{right}. What policy is optimal if $\gamma = 0$? If $\gamma = 0.9$? If $\gamma = 0.5$?



Answer: If $\gamma = 0$, then delayed rewards do not count at all. The left action has value 1 from the top state, and the right action has value 0; π_{left} is optimal.

If $\gamma = 0.9$, then delayed rewards are given substantial weight. Under π_{left}, the sequence of rewards from the top state is 1, 0, 1, 0, 1, 0, ..., and the corresponding return and value is $1 + \gamma^2 + \gamma^4 + \cdots = 1/(1-\gamma^2) \approx 5.26$. Under π_{right}, the sequence of rewards from the top state is 0, 2, 0, 2, 0, 2, ..., and the corresponding return and value is $2\gamma + 2\gamma^3 + 2\gamma^5 + \cdots = 2\gamma(1 + \gamma^2 + \gamma^4 + \cdots) = 2\gamma/(1-\gamma^2)$, which is clearly better as $2\gamma > 1$. Only π_{right} is optimal.

If $\gamma = 0.5$, we have a borderline case. The returns are exactly as computed above, but now we have $2\gamma = 1$, and thus the two actions have the same value. Both policies are optimal. □

Exercise 3.23 Give the Bellman equation for q_* for the recycling robot.

Answer: In general, the Bellman equation for q_* is

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

For the recycling robot, there are 5 possible state-action pairs, so the Bellman equation for q_* is actually 5 equations. Abbreviate the states `high` and `low`, and the actions `search`, `wait`, and `recharge` respectively by `h`, `l`, `s`, `w`, and `r`.

$$\begin{aligned}
q_*(\texttt{h},\texttt{s}) &= \alpha[r_{\texttt{search}} + \gamma \max\{q_*(\texttt{h},\texttt{s}), q_*(\texttt{h},\texttt{w})\}] \\
&\quad + (1-\alpha)[r_{\texttt{search}} + \gamma \max\{q_*(\texttt{l},\texttt{s}), q_*(\texttt{l},\texttt{w}), q_*(\texttt{l},\texttt{r})\}] \\
q_*(\texttt{l},\texttt{s}) &= (1-\beta)[-3 + \gamma \max\{q_*(\texttt{h},\texttt{s}), q_*(\texttt{h},\texttt{w})\}] \\
&\quad + \beta[r_{\texttt{search}} + \gamma \max\{q_*(\texttt{l},\texttt{s}), q_*(\texttt{l},\texttt{w}), q_*(\texttt{l},\texttt{r})\}] \\
q_*(\texttt{h},\texttt{w}) &= r_{\texttt{wait}} + \gamma \max\{q_*(\texttt{h},\texttt{s}), q_*(\texttt{h},\texttt{w})\} \\
q_*(\texttt{l},\texttt{w}) &= r_{\texttt{wait}} + \gamma \max\{q_*(\texttt{l},\texttt{s}), q_*(\texttt{l},\texttt{w}), q_*(\texttt{l},\texttt{r})\} \\
q_*(\texttt{l},\texttt{r}) &= \gamma \max\{q_*(\texttt{h},\texttt{s}), q_*(\texttt{h},\texttt{w})\}
\end{aligned}$$

□

Exercise 3.24 Figure 3.5 gives the optimal value of the best state of the gridworld as 24.4, to one decimal place. Use your knowledge of the optimal policy and (3.8) to express this value symbolically, and then to compute it to three decimal places.

Answer: Following the optimal policy from the best state the agent receives a first reward of +10 and then four rewards of 0, then the sequence repeats. The optimal value of the state then must be

$$\begin{aligned}
v_*(s) &= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \;\middle|\; S_t = s\right] \\
&= 10 + \gamma 0 + \gamma^2 0 + \gamma^3 0 + \gamma^4 0 + \gamma^5 10 + \gamma^6 0 + \gamma^7 + \gamma^8 0 + \gamma^9 0 + \gamma^{10} 10 + \cdots \\
&= \gamma^0 10 + \gamma^5 10 + \gamma^{10} 10 + \cdots \\
&= 10 \sum_{k=0}^{\infty} (\gamma^5)^k \\
&= \frac{10}{1-\gamma^5}
\end{aligned}$$

Substituting $\gamma = 0.9$, we obtain $v_*(s) = 24.419$. □

Exercise 3.25 Give an equation for v_* in terms of q_*.

Answer:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_*(s,a) \quad \text{for all } s \in \mathcal{S} \qquad \square$$

Exercise 3.26 Give an equation for q_* in terms of v_* and the four-argument p.

Answer:

$$q_*(s,a) = \sum_{s' \in \mathcal{S}^+} \sum_{r \in \mathcal{R}} p(s', r \,|\, s, a)\big(r + \gamma v_*(s')\big) \quad \text{for all } s \in \mathcal{S}, a \in \mathcal{A}(s) \qquad \square$$

Exercise 3.27 Give an equation for π_* in terms of q_*.

Answer:

$$\pi_*(s) = \operatorname*{arg\,max}_{a \in \mathcal{A}(s)} q_*(s,a) \quad \text{for all } s \in \mathcal{S} \qquad \square$$

Exercise 3.28 Give an equation for π_* in terms of v_* and the four-argument p.

Answer:

$$\pi_*(s) = \operatorname*{arg\,max}_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}^+} \sum_{r \in \mathcal{R}} p(s', r \,|\, s, a)\big(r + \gamma v_*(s')\big) \quad \text{for all } s \in \mathcal{S} \qquad \square$$

Exercise 3.29 Rewrite the four Bellman equations for the four value functions (v_π, v_*, q_π, and q_*) in terms of the three argument function p (3.4) and the two-argument function r (3.5).

Answer:

$$v_\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}^+} p(s' \,|\, s, a) v_\pi(s') \right) \quad \text{for all } s \in \mathcal{S}$$

$$v_*(s) = \max_{a \in \mathcal{A}(s)} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}^+} p(s' \,|\, s, a) v_*(s') \right) \quad \text{for all } s \in \mathcal{S}$$

$$q_\pi(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}^+} p(s' \,|\, s, a) \sum_{a' \in \mathcal{A}(s')} \pi(a'|s') q_\pi(s', a') \quad \text{for all } s \in \mathcal{S}, a \in \mathcal{A}(s)$$

$$q_*(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}^+} p(s' \,|\, s, a) \max_{a' \in \mathcal{A}(s')} q_*(s', a') \quad \text{for all } s \in \mathcal{S}, a \in \mathcal{A}(s) \qquad \square$$