Exercises - MDPs and Dynamic Programming

Reinforcement Learning

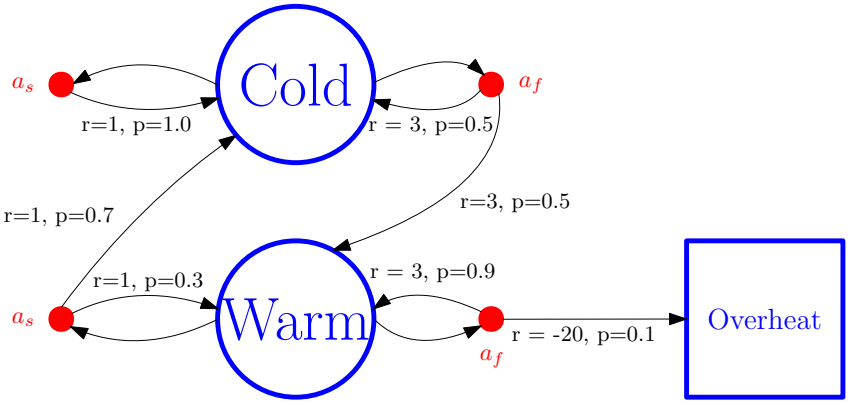
1. Do Exercise 3.1 - 3.3 in the textbook.

2. Exercise 3.8 in the textbook: 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 $R_5 = 2$ and then the episode terminates. What is G_0, G_1, G_2, G_3, G_4 and G_5?

3. Consider DC-motor modeled as
$$\theta_{t+1} = b\theta_t + c\theta_{t-1} + da_t$$
where θ_t is the angle of rotation at time t, a_t is the input voltage at time t. Here c_1, c_2 and b_1 are constant parameters. Determine a state s for the system. What is the state space?

4. Consider throwing a coin over and over again until it lands on heads. What will the average number of times you throw it be? (Note that you always throw it at least once).

   We can model this as an MDP in the following way. We have only one action (throw the coin), so the policy is to always throw the coin. We have two states, `continue` (which we will be in as long as the coin shows tails) or `stop` (a terminating state we reach when the coin shows heads). When we are in `continue` we will throw the coin, receive a reward of +1 and then the probability is 0.5 to end up in either `continue` or `stop`. This is an undiscounted task, so $\gamma = 1$. The state value $v($`continue`$)$ is the average expected total reward, i.e., the average number of times the coin will be thrown before we see heads. What is the value of the state `continue` in the MDP described above?



Figur 5.1: The racing agent MDP. Here r is immediate reward and p the transition probability.

5. Consider the MDP in Figure 5.1. It describes a racing agent that at each time step chooses between to either go fast (a_f) or slow (a_s). During operation the engine can be either warm (w) or cold (c). When the racer goes fast, there is some probability that the engine gets warm, and it can even overheat in which case the race is over (overheat is a terminal state). The racer gets higher reward for going fast than slow, but also a negative reward for overheating. We consider a discounted process with a discount factor $\gamma = 0.9$.
   (a) Use Figure 5.1 to find $p(s', r|s, a)$ for $s = w$ (warm) and $a = a_f$.
   (b) What is the expected immediate reward $r(s, a)$ of choosing action $a = a_f$ in state $s = w$?
   (c) Let $\pi(s|a) = 0.5$ for all s and a. Use the Bellman equations to confirm that $v_\pi(c) = 14.13$ and $v_\pi(w) = 11.53$ (rounded to two decimals).

(d) In order to find a better policy, we do policy improvement by acting greedily with respect to the value function in (a). Which action will you then choose in state w?

(e) State the Bellman optimality equation for the optimal state values $v_*(s)$ for all s.

(f) Perform two sweeps of value iteration with synchronous updates (do not use in-place version) for all states. Start from the initial value function $V_0(c) = V_0(w) = 0$.

(g) Confirm that $v_*(c) = 22.37$ and $v_*(w) = 20.68$ (rounded two two decimals). Determine what the optimal action in each state is. Also re-do part (b) but with the optimal policy, and confirm that the optimal state values satisfies the Bellman equation for the optimal policy.

6. Do Exercise 4.2 in the textbook.

   Hint: In the first part of the exercise the transition probabilities *from* all states except 15 are unchanged, meaning that you cannot reach state 15 from any other state.

7. (*) In this exercise we will study if the optimal policy of an MDP can change when we add a constant to all rewards.

   (a) Consider a continuing task (no terminal state). Suppose that we are given the state-value function $v_\pi(s)$ for the policy π. We now change the MDP by adding a constant c to *all* rewards. What is the state-value function for π in the new MDP? Will the new MDP have the same optimal policies as the original one?

   (b) Now consider adding a constant c to all rewards in an episodic task with terminal states. Can this change the optimal policy of the MDP? (*Note:* In the episodic task all rewards received after reaching a terminal state are 0, and the constant added will not affect this).

8. (*) We are in an environment with finite state an action spaces and a fixed (but arbitrary) transition function

$$p(s'|s, a)$$

   However, we are considering three different reward functions $r_1(s, a), r_2(s, a)$ and

$$r_3(s, a) = r_1(s, a) + r_2(s, a).$$

   Note: The reward function is defined in equation (3.5) in the textbook and is

$$r(s, a) = \sum_{r,s'} rp(s', r|s, a).$$

   This means that $p(s', r|s, a)$ is different for $r_1(s, a)$, $r_2(s, a)$ and $r_3(s, a)$. However,

$$p(s'|s, a) = \sum_{r} p(s', r|s, a).$$

   is the same for all three cases.

   (a) We want to compute the action-value functions for some fixed policy. Suppose that you are given the action values $q_1^\pi(s, a)$ and $q_2^\pi(s, a)$ corresponding to reward functions r_1 and r_2.
   Is it true that the action-values with reward function r_3 are given by $q_3^\pi(s, a) = q_1^\pi(s, a)+q_2^\pi(s, a)$?

   (b) Now suppose that you are given the optimal action-values $q_1^*(s, a)$ and $q_2^*(s, a)$ corresponding to r_1 and r_2 respectively.
   Are the optimal action values with reward function r_3 given by $q_3^*(s, a) = q_1^*(s, a) + q_2^*(s, a)$?

   (c) Explain why the result in part (b) does not contradict the result in part (a).

Solutions

1. These exercises can be solved in many different ways, and the important part is that you think about possible solutions. You can discuss your ideas with other students and/or the teachers during exercise sessions.

2. The discounted return can be written as

$$G_t = R_{t+1} + \gamma G_{t+1}$$

for $t < T$ and $G_T = 0$. We start from the end:

$$\begin{aligned}
G_5 &= G_T = 0 \\
G_4 &= R_5 + \gamma G_5 = 2 \\
G_3 &= R_4 + \gamma G_4 = 3 + 1 = 4 \\
G_2 &= R_3 + \gamma G_3 = 6 + 2 = 8 \\
G_1 &= R_2 + \gamma G_2 = 2 + 4 = 6 \\
G_0 &= R_1 + \gamma G_1 = -1 + 3 = 2
\end{aligned}$$

3. The state must be such that the state s_t and action a_t at time t contain all relevant information for predicting s_{t+1}.

We can see that θ_t by itself is *not* a state, since θ_{t+1} also depends on θ_{t-1} which cannot be computed from θ_t and a_t alone. However, if we know θ_t and θ_{t-1} together with the action a_t, then we have enough information to determine θ_{t+1}. Hence we could use

$$s_t = \begin{bmatrix} \theta_t \\ \theta_{t-1} \end{bmatrix}$$

as a state. With this we get $\theta_t = \begin{bmatrix} 1 & 0 \end{bmatrix} s_t$ and (according to the equation in the problem)

$$s_{t+1} = \begin{bmatrix} \theta_{t+1} \\ \theta_t \end{bmatrix} = \begin{bmatrix} b & c \\ 1 & 0 \end{bmatrix} s_t + \begin{bmatrix} d \\ 0 \end{bmatrix} a_t.$$

Hence, s_t together with future actions contain all information relevant for predicting future values of θ_t. The state space in this case is thus $\mathcal{S} \subset \mathbb{R}^2$. More specifically $\mathcal{S}$ contains the subset of $\mathbb{R}^2$ for which both elements lie between e.g. 0 and 2π (since θ_t is an angle).

Note: There are infinitely many ways to choose a state for the environment, so this is not the only correct answer. For example

$$s_t = \begin{bmatrix} \theta_t \\ \theta_t - \theta_{t-1} \end{bmatrix}$$

where the first element is the angle and the second element is the change of the angle (some kind of estimate of the angular velocity) could also have been used as a state. In this case

$$s_{t+1} = \begin{bmatrix} b + c & -c \\ 1 & 0 \end{bmatrix} s_t + \begin{bmatrix} d \\ 0 \end{bmatrix} a_t$$

4. The Bellman equation tells us that the value of a state is the immediate expected reward + the expected discounted value of the next state. In this case the immediate reward is always +1, and the next state is either `continue` or `stop`. Hence

$$v(\texttt{continue}) = 1 + \gamma(0.5v(\texttt{continue}) + 0.5v(\texttt{stop})) = 1 + 0.5\gamma v(\texttt{continue}),$$

since the terminating state `stop` has value $v(\texttt{stop}) = 0$. Here we have $\gamma = 1$, so

$$v(\texttt{continue}) = \frac{1}{1 - 0.5\gamma} = 2.$$

5.

(a) From the figure we can see that

$$p(w, 3|w, a_f) = 0.9$$
$$p(o, -20|w, a_f) = 0.1$$

and $p(s', r|w, a_f) = 0$ for all other s' and r.

(b) The expected immediate reward is given by

$$r(s, a) = \mathrm{E}[R_t|S_{t-1} = s|A_{t-1}a] = \sum_{r,s'} rp(s', r|s, a).$$

For this problem it is given by

$$r(w, a_f) = 3 \times 0.9 + (-20 \times 0.1) = 0.7.$$

(c) The Bellman equation for each state is given by

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

We start by computing the action-values

$$q_\pi(c, a_s) = 1 + \gamma v_\pi(c) = 13.72$$
$$q_\pi(c, a_f) = 0.5(3 + \gamma v_\pi(c)) + 0.5(3 + \gamma v_\pi(w)) = 14.55$$
$$q_\pi(w, a_s) = 0.7(1 + \gamma v_\pi(c)) + 0.3(1 + \gamma v_\pi(w)) = 13.02$$
$$q_\pi(w, a_f) = 0.9(3 + \gamma v_\pi(w)) + 0.1(-20 + 0\gamma) = 10.04$$

since overheat is a terminating state with value 0. The Bellman equation is then

$$v_\pi(c) = 0.5q_\pi(c, a_s) + 0.5q_\pi(c, a_f) = 14.13$$
$$v_\pi(w) = 0.5q_\pi(w, a_s) + 0.5q_\pi(w, a_f) = 11.53$$

Since the given $v_\pi(s)$ solves the Bellman equations, it means that it is indeed the value function for policy π.

(d) The action values for the policy π in part (a) in state w are given by

$$q_\pi(w, a_s) = 13.02$$
$$q_\pi(w, a_f) = 10.04$$

Hence the greedy policy with respect to $v_\pi(s)$ in w is

$$\arg\max_a q_\pi(w, a) = a_s.$$

(e) The Bellman optimality equation for each state is

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

For the given MDP this becomes

$$v_*(c) = \max \begin{cases} 1 + \gamma v_*(c) & (\text{if } a = a_s) \\ 0.5[3 + \gamma v_*(c)] + 0.5[3 + \gamma v_*(w)] & (\text{if } a = a_f) \end{cases}$$

$$v_*(w) = \max \begin{cases} 0.7[1 + \gamma v_*(c)] + 0.3[1 + \gamma v_*(w)] & (\text{if } a = a_s) \\ 0.9[3 + \gamma v_*(w)] + 0.1[-20 + \gamma \times 0] & (\text{if } a = a_f) \end{cases}$$

(f) Here we just insert the current estimate into the right-hand side of the Bellman optimality equation. Initially we have $V_0(c) = V_0(w) = 0$. Hence

$$V_1(c) = \max \begin{cases} 1 + 0\gamma \\ 0.5[3 + 0\gamma] + 0.5[3 + 0\gamma] \end{cases} = \max\{1, 3\} = 3$$

$$V_1(w) = \max \begin{cases} 0.7[1 + 0\gamma] + 0.3[1 + 0\gamma] \\ 0.9[3 + 0\gamma] + 0.1[-20 + 0\gamma] \end{cases} = \max\{1, 0.7\} = 1$$

and with one more sweep we get

$$V_2(c) = \max \begin{cases} 1 + 3\gamma \\ 0.5[3 + 3\gamma] + 0.5[3 + 1\gamma] \end{cases} = \max\{3.7, 4.8\} = 4.8$$

$$V_2(w) = \max \begin{cases} 0.7[1 + 3\gamma] + 0.3[1 + 1\gamma] \\ 0.9[3 + 1\gamma] + 0.1[-20 + 0\gamma] \end{cases} = \max\{3.16, 1.51\} = 3.16$$

(g) To confirm that the values are correct, just insert these values in the right-hand side of the equations in part (d) to see that they solve the Bellman optimality equation. The greedy policy with respect to $q_*(s, a)$ is optimal, i.e.

$$\pi_*(s) = \arg\max_a q_*(s, a)$$

where

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

In our case we get

$$q_*(c, a_s) = 21.14$$
$$q_*(c, a_f) = 22.37$$

so $\pi_*(c) = a_f$. For $s = w$ we get

$$q_*(w, a_s) = 20.68$$
$$q_*(w, a_f) = 17.45$$

so $\pi_*(w) = a_s$.

If you want to write the policy as a distribution we thus have

$$\pi(a_s|c) = 0, \quad \pi(a_f|c) = 1.0, \quad \pi(a_s|w) = 1.0, \quad \pi(a_f|w) = 0.0.$$

Inserting these into the Bellman equation

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a)$$

we get

$$v_\pi(c) = \pi(a_f|c) q_\pi(c, a_f) = 22.37 = v_*(c)$$
$$v_\pi(w) = \pi(a_s|w) q_\pi(w, a_s) = 20.68 = v_*(c)$$

6. We start with the case when we only add state 15, and keep the dynamics for all other states (this means that we cannot reach state 15 from the other states, so the values of these states will not be affected).

The Bellman equation for state 15 is ($\gamma = 1$ and we have $v_\pi(s)$ for $s = 11, 12, 13, 14$ given in Figure 4.1)

$$\begin{aligned} v_\pi(15) &= 0.25(-1 + \gamma v_\pi(12)) + 0.25(-1 + \gamma v_\pi(13)) + 0.25(-1 + \gamma v_\pi(14)) + 0.25(-1 + \gamma v_\pi(15)) \\ &= -1 + 0.25(v_\pi(12) + v_\pi(13) + v_\pi(14) + v_\pi(15)) = -15 + 0.25 v_\pi(15) \end{aligned}$$

hence

$$0.75v_\pi(15) = -15 \iff v_\pi(15) = -20$$

We now change the dynamics so `down` in state 13 takes us to $s = 15$ instead of $s = 13$. To evaluate the value function in this new environment we start with an initial guess $V_0(s)$ equal to Figure 4.1 and $V_0(15) = -20$. Note that the dynamics has not changed for any state except 13, so we see directly that with synchronous policy evaluation $V_1(s) = V_0(s)$ for all $s \neq 13$. Furthermore

$$V_1(13) = -1 + 0.25(V_0(9) + V_0(12) + V_0(14) + V_0(15)) = -20 = V_0(13).$$

Hence $V_1(s) = V_0(s)$ for all s, and thus $V_0(s)$ much be a solution to the Bellman equation. Hence $v_\pi(15) = -20$ also with these new dynamics.

7.

(a) The state value function is given by

$$v_\pi(s) = \mathrm{E}[G_t | S_t = s],$$

where

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

If we add a constant to all rewards to get $\tilde{R}_t = R_t + c$ then the new returns are (assuming that $\gamma < 1$)

$$\tilde{G}_t = \sum_{k=0}^{\infty} \gamma^k (R_t + c) = \sum_{k=0}^{\infty} \gamma^k R_t + \sum_{k=0}^{\infty} \gamma^k c = G_t + \frac{1}{1-\gamma} c$$

and thus

$$\tilde{v}_\pi(s) = v_\pi(s) + \frac{1}{1-\gamma} c.$$

Hence, the relative value of policies will not change, so if policy π was better than policy π' in the original MDP the same will be true for the new MDP. Hence the optimal policy will not change.

(b) Now we instead look at an episodic task. This means that the policy will also influence the number of steps before the episode terminate, and thus the reasoning above does not hold. In fact, in this case it is possible to find examples when adding a constant to all rewards changes the policy.

For example, consider the GridWorlds in Tinkering Notebook 2. Here each action gave a reward of -1, and thus the reward was maximized by finding a terminating state as fast as possible. If we add a constant 2 to all rewards, then the agent instead receives a reward +1 for each action and thus tries to avoid the terminating states instead.

8. To avoid confusion, let us denote $p_i(s', r|s, a)$ as the transition probabilities when we use reward function i.

(a) Note that for reward function r_i we have

$$\begin{aligned} q_i^\pi(s,a) &= \sum_{r,s'} p_i(s', r|s, a) \left(r + \gamma \sum_{a'} \pi(a'|s') q_i^\pi(s', a') \right) \\ &= \sum_{r,s'} p_i(s', r|s, a) r + \gamma \sum_{r,s'} p_i(s', r|s, a) \sum_{a'} \pi(a'|s') q_i^\pi(s, a) \\ &= r_i(s,a) + \gamma \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') q_i^\pi(s,a). \end{aligned}$$

It follows that

$$q_1^\pi(s,a) + q_2^\pi(s,a) = \underbrace{r_1(s,a) + r_2(s,a)}_{r_3(s,a)} + \gamma \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') (q_1^\pi(s,a) + q_2^\pi(s,a))$$

We thus see that $q_3^\pi(s,a) = q_1^\pi(s,a) + q_2^\pi(s,a)$ solves the Bellman equation when $r_3(s,a)$ is used, and thus this gives the correct action-values.

(b) We can see that

$$\begin{aligned} q_i^*(s,a) &= \sum_{r,s'} p_i(s',r|s,a)\left(r + \gamma \max_{a'} q_i^*(s',a')\right) \\ &= r_i(s,a) + \gamma \sum_{s'} p(s'|s,a) \max_{a'} q_i^*(s',a') \end{aligned}$$

Hence

$$q_1^*(s,a) + q_2^*(s,a) = r_3(s,a) + \gamma \sum_{s'} p(s'|s,a)\left(\max_{a'} q_1^*(s',a') + \max_{a'} q_2^*(s',a')\right)$$

However, this means that $q_1^*(s,a) + q_2^*(s,a)$ solves the Bellman optimality equation for $q_3^*(s,a)$ if and only if $\max_{a'} q_1^*(s',a') + \max_{a'} q_2^*(s',a') = \max_{a'}(q_1^*(s',a') + q_2^*(s',a'))$ for all s' and a'. This does typically not hold, so in general $q_3^*(s,a) \neq q_1^*(s,a) + q_2^*(s,a)$.

(c) In part (a) we look at a fixed policy π, and it holds that $q_3^\pi(s,a) = q_1^\pi(s,a) + q_2^\pi(s,a)$. This is true for any fixed policy, so why does it not hold for the optimal policy?

The reason is that the optimal policy for $r_1(s,a)$ may not be the same as the optimal policy for $r_2(s,a)$. Hence, $q_i^*(s,a)$ consider the action-values for different policies.

A simple example: We look at a simple environment with two action x and y. The environment always terminate after one action. Assume that $r_1(s,x) = 1$ and $r_1(s,y) = 0$. Also assume that $r_2(s,x) = -1$ and $r_2(s,y) = 1$. Then the optimal action for r_1 is x while the optimal action for r_2 is y. Hence they have different optimal policies, and therefore the result in part (a) does not imply that the equality holds for part (b).