Chapter 3: Markov Decision Processes

Nisheet Patel
Answers to textbook problems

September 6, 2019

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.

1. Chess

   - States: Configuration of the board
   - Actions: Given the state, all legal moves would be in the action space
   - Reward: +1 for winning, -1 for losing, 0 for drawing the game and all intermediate moves.

   Issues: We may never experience states again because of the enormous state-space. Given that there are roughly between 20-30 legal actions available at each state (plus the same number for the opponent), it is impossible to simulate a game to completion since computing all the actions will be 40^m, where m is the number of moves until reward, i.e. game completion.

2. Navigation

   - States: We can discretize the map of the world in a (say square) grid. A state will be the current position of the agent.
   - Actions: Except the edges or obstacles, agents will have 4 actions available (left, right, up/forward, down/backward), which will take them to the corresponding state. For the edges and obstacles, we can either define the next state for actions that bump into walls or run out of the map to remain the same as the current one, or that the loops on the other side of the map, or disallow these actions (remove them from the actions space) altogether.
   - Some random states can be initialised with rewards.

3. Prawn-farming

   - States: Current sensor readings of concentrations of chemicals, food, salt, water temperature, outside temperature, weight/size of prawns.

- Actions: Change one or more parameters up or down and by how much. Action space can be discretized.
- Rewards: +1 if prawns grow (define criterion, e.g. weight), 0 if they don't, and -1,000,000 if they die.

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

No. It requires complete knowledge of all actions that are possible in the state, and of the state itself. Moreover, the Markov property may not be satisfied, for instance, in a weather prediction problem. Since we do not know the current state of the world completely, we rely not only on the current state of the weather, but the history.

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 outsay, where the rubber meets the road, considering your actions to be tire torques. Or you could define them farther insay, 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?*

It depends on the task at hand, but the boundary should be put at the limit of the driver's control. I would certainly not define them as farther out as where the rubber meets the road and where actions would be tires torques - since I control these only indirectly, neither would I define them as far in as where the brain meets the body and the actions are the muscle twitches to control limbs - since these are not relevant for driving either and must have been learnt previously in life.

I would think of this as a hierarchical problem, and depending on the goal, put the boundaries at the relevant level. For navigation, if 99% of the sub-goals are successful, then it may be easy to convert them into actions, and the agent only has to worry about getting the actual goal (destination) with the actions being choices of where to drive (left, right, straight, etc.). Whereas for learning to drive, the actions may be better defined in terms of accelerator, steering wheel, and brakes.

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 $p(s', r|s, a) > 0$.*

Note that all the quadruples that aren't listed in the table below have zero probability.

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

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).

For all non-terminal states, the equation remains the same. However, if the agent hits a terminal state, they would transition to a standard starting state, s_0, with $p(s' = s_0, r = 0|s = S^+) = 1$, or to a sample from a standard distribution of starting states.

$$\sum_{s' \in S} \sum_{r \in R} p(s', r|s, a) = 1, \qquad \forall s \in S - \{S^+\},\ a \in A(s)$$

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?

The return at each time would be $-\gamma^K$, if K is the number of time-steps until failure from the current configuration of the pole and cart. The discounted version of the task would have a slightly different return, $-\gamma^K - \gamma^{K+T} - \gamma^{K+2T} - ... = -\gamma^K \sum_{n=0}^{\infty} \gamma^{nT}$, where T is the expected number of time-steps until failure from the starting configuration of the pole.

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?

The agent is being rewarded only for escaping the maze. As long it does escape the maze at some point, the expected return is +1 regardless of how long it takes to do so (since there is no discounting). If instead, the rewards were changed to be -1 for every step that it is in the maze, and 0 for escaping, it would try to find ways to escape the maze quicker without plateauing. Alternatively, we can keep the reward formulation the same but introduce discounting, which will force the agent to find quicker routes to escape the maze.

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 $G_0,\ G_1,\ ...,\ G_5$?*

We know that $G_T = 0$, and $G_t = R_{t+1} + \gamma G_{t+1}$. Working backwards, we get:

$$\begin{aligned}
G_5 &= 0 && = 0\\
G_4 &= 2 && = 2\\
G_3 &= 3 + 0.5 * 2 && = 4\\
G_2 &= 6 + 0.5 * 3 + 0.25 * 2 && = 8\\
G_1 &= 2 + 0.5 * 6 + 0.25 * 3 + 0.125 * 2 && = 6\\
G_0 &= 1 + 0.5 * 2 + 0.25 * 6 + 0.125 * 3 + 0.0625 * 2 && = 4
\end{aligned}$$

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 ?*

$$\begin{aligned}
G_1 &= 7 \times \sum_{i=0}^{\infty} \gamma^i = \frac{7}{1-\gamma} = \frac{7}{1-0.9} && = 70\\
G_0 &= 2 + 0.9 * 70 && = 63
\end{aligned}$$

Exercise 3.10. *Prove the second equality in (3.10)*

$$\begin{aligned}
\sum_{k=0}^{\infty} \gamma^k &= 1 + \gamma + \gamma^2 + ...\\
\therefore (1-\gamma)\sum_{k=0}^{\infty} \gamma^k &= (1-\gamma)(1 + \gamma + \gamma^2 + ...)\\
&= 1 - \cancel{\gamma} + \cancel{\gamma} - \cancel{\gamma^2} + \cancel{\gamma^2} - ... - \cancel{\gamma^n} + \cancel{\gamma^n} - ... - \cancel{\gamma^\infty}\ 0\ (\because \gamma < 1)\\
&= 1
\end{aligned}$$

Thus, $\sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$.

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)?*

$$\langle R_{t+1}\rangle = \sum_a \pi(a|S_t) \sum_{s',r} r \cdot p(s',r|S_t,a)$$

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

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

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

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

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.)

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

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?

Only the relative rewards are important. If we add a constant c to all rewards in (3.8), we get the modified G'_t:

$$\begin{aligned} G_{t,new} &= R_{t+1} + c + \gamma(R_{t+2} + c) + \gamma^2(R_{t+3} + c) + ... \\ &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... + c(1 + \gamma + \gamma^2 + ...) \\ &= G_t + \frac{c}{1-\gamma} \end{aligned}$$

Thus, $v_c = \frac{c}{1-\gamma}$, as shown in the equations below.

$$\begin{aligned} v_{\pi,new}(s) &= \mathbb{E}[G'_t|s] \\ &= \mathbb{E}[G_t + \frac{c}{1-\gamma}|s] \\ &= \mathbb{E}[G_t|s] + \frac{c}{1-\gamma} \\ &= v_\pi(s) + v_c \end{aligned}$$

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.

Yes, it would have an effect in an episodic task. Consider the example in Exercise 3.7 (without discounting). If the terminal state has reward +1, and all other states have reward 0, then the agent is being rewarded to escape the maze, but not based on how quickly they escape the maze, since they are rewarded in exactly the same way (+1 upon escape) as long they escape the maze at some point. Now if we add $c = -1$ to each state, then the terminal state has reward 0, whereas all other states have reward -1. Now the agent would learn how to escape the maze as quickly as possible in order to maximise their reward.

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 stateaction pair (s, a). Show the sequence of equations analogous to (3.14), but for action values.

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

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. (Refer to the figure in the book). 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.*

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

Exercise 3.19 *The value of an action, q(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 (stateaction pair) and branching to the possible next states: (Refer to the figure in the book). 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, $v(S_{t+1})$, given that $S_t = s$ and $A_t = a$. This equation should include an expectation but not one 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.*

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

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

The optimal state value function for the golf example would have the -3 and -2 contours as in Figure 3.3 (lower, ref:book), *i.e.* the one for $q_*(s, driver)$, and the -1 contour in the green part as in Figure 3.3 (upper, ref:book). This corresponds to the first two shots being taken with the driver, and the last shot being taken by the putter.

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

They would be the same as Figure 3.3 (upper, ref:book) or v_{putt}.

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 $\gamma = 0.5$?*

- If $\gamma = 0$, π_{left} is optimal:
  - $G_t(left) = 1 + 0 * 0 = 1$, and
  - $G_t(right) = 0 + 0 * 2 = 0$.
- If $\gamma = 0.9$, π_{right} is optimal:
  - $G_t(left) = 1 + 0.9 * 0 + 0.9^2 * G_t(left)$, thus $G_t(left) = \frac{1}{1-0.81} = 5.26$, and
  - $G_t(right) = 0+0.9*2+0.9^2*G_t(right)$, thus $G_t(right) = \frac{1.8}{1-0.81} = 9.47 > G_t(left)$
- If $\gamma = 0.5$, $\pi_{left} = \pi_{right}$ (both are optimal):
  - $G_t(left) = 1 + 0.5 * 0 + 0.5^2 * G_t(left)$, thus $G_t(left) = \frac{1}{1-0.25} = 1.33$, and
  - $G_t(right) = 0+0.5*2+0.5^2*G_t(right)$, thus $G_t(right) = \frac{1}{1-0.25} = 1.33 = G_t(left)$.

Exercise 3.23 Give the Bellman equation for q_* for the recycling robot. The Bellman optimality equation for $q_*(s, a)$ is given by:

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

To make things compact, we will abbreviate states *high* and *low*, and actions *search*, *wait*, and *recharge* with h, l, s, w, and re. There are 6 combinations of $q_*(s, a)$. Since the rewards are a deterministic function of the previous state, I will use $p(s'|s, a)$ instead of the four-argument p.

$$\begin{aligned}
q_*(h,s) &= p(h|h,s)[r(h,s) + \gamma \max_{a'} q_*(h,a')] + p(l|h,s)[r(h,s) + \gamma \max_{a'} q_*(l,a')] \\
&= \alpha[r_s + \gamma \max_{a'} q_*(h,a')] + (1-\alpha)[r_s + \gamma \max_{a'} q_*(l,a')] \\
&= r_s + \gamma \max_{a'}[\alpha q_*(h,a') + (1-\alpha) q_*(l,a')] \\
&= r_s + \gamma \max \left\{ \begin{matrix} \alpha q_*(h,s) + (1-\alpha) q_*(l,s), \\ \alpha q_*(h,w) + (1-\alpha) q_*(l,w), \\ \alpha q_*(h,re) + (1-\alpha) q_*(l,re) \end{matrix} \right\}
\end{aligned}$$

Similarly, we can get equations for all the other combinations (s, a) for $q_*(s, a)$:

$$q_*(h,w) = r_{wait} + \gamma \max \left\{ \begin{matrix} q_*(h,s), \\ q_*(h,w), \\ q_*(h,re) \end{matrix} \right\}$$

$$q_*(h,re) = \gamma \max \left\{ \begin{matrix} q_*(h,s), \\ q_*(h,w), \\ q_*(h,re) \end{matrix} \right\}$$

$$q_*(l,s) = \beta r_{search} - 3(1-\beta) + \gamma \max \left\{ \begin{matrix} (1-\beta)q_*(h,s) + \beta q_*(l,s), \\ (1-\beta)q_*(h,w) + \beta q_*(l,w), \\ (1-\beta)q_*(h,re) + \beta q_*(l,re) \end{matrix} \right\}$$

$$q_*(l,w) = r_{wait} + \gamma \max \left\{ \begin{matrix} q_*(l,s), \\ q_*(l,w), \\ q_*(l,re) \end{matrix} \right\}$$

$$q_*(l,re) = \gamma \max \left\{ \begin{matrix} q_*(h,s), \\ q_*(h,w), \\ q_*(h,re) \end{matrix} \right\}$$

Thus, we get 6 equations in 6 variables, $q_*(s,a)$, where $s \in \{h,l\}$, $a \in \{s,w,re\}$. For any choice of r_s, r_w, α, β, and γ, with $0 \leq \gamma < 1$, $0 \leq \alpha, \beta \leq 1$, there is exactly one sextuplet (6-tuple) of numbers that satisfy these 6 nonlinear equations.

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.*

The optimal policy (Figure 3.5 (right)) says that once we are in state A, we get sent to state A', and should only go up all the way until we are in state A again, and repeat that procedure forever. From Equation 3.8, the value of state A for following this optimal policy would be given by:

$$\begin{aligned} v_*(A) &= 10 + 0 + 0 + 0 + 0 + \gamma^5 * 10 + ... + \gamma^{10} * 10 + ... \\ &= 10(1 + \gamma^5 + \gamma^{10} + ...) \\ &= \frac{10}{1-\gamma^5} \\ &= \frac{10}{1-0.9^5} \\ &= 24.419 \end{aligned}$$

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

$$v_*(s) = \max_{a \in A(s)} q_*(s,a)$$

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

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

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

$$\pi_*(a|s) = \arg\max_{a \in A(s)} q_*(s,a)$$

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

$$\pi_*(a|s) = \operatorname*{arg\,max}_{a \in A(s)} \sum_{r,s'} p(r,s'|s,a)[r + \gamma v_*(s')]$$

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).

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

$$\begin{aligned}
v_*(s) &= \max_\pi v_\pi(s) \\
&= \mathbb{E}_{\pi^*}[G_t|S_t = s] \\
&= \mathbb{E}_{\pi^*}[R_{t+1} + \gamma G_{t+1}|S_t = s] \\
&= \max_a \sum_{s'} p(s'|s,a)\big[r(s,a) + \gamma\, \mathbb{E}_{\pi^*}[G_{t+1}|S_{t+1} = s']\big] \\
&= \max_a \sum_{s'} p(s'|s,a)[r(s,a) + \gamma v_*(s')]
\end{aligned}$$

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

$$\begin{aligned}
q_*(s,a) &= \max_\pi q_\pi(s,a) \\
&= \sum_{s'} p(s'|s,a)\big[r(s,a) + \gamma \max_{a' \in A(s')} q_*(s',a')\big]
\end{aligned}$$