Reinforcement Learning: An Introduction
Attempted Solutions
Chapter 3

Scott Brownlie & Rafael Rui

1 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.

An e-commerce sit could use reinforcement learning to control daily pricing of products. The actions would be the the prices set for each product on each day. The states might include the month of the year, the day of the week and the proximity to special days such as Christmas and Valentine's Day. The reward would be the profit at the end of each day.

The manager of a football team could use reinforcement learning to pick the 11 players to play each game. The actions would be the team selection. The states could be the opponent, whether the game is home or away and the fitness of the players. The reward would be 0, 1 or 3 depending on whether the team lost, drew or won the game.

A company could use reinforcement learning to control the air temperature in its office. The actions would be the specific settings of the heating/air-conditioning system. The states would include the current outdoor temperature and the indoor temperature in each room in the building. The reward would be the satisfaction of the employees, which could be measured by selecting 10 employees at random every hour and asking them to rate the their comfort on a scale of 1 to 10 and then averaging the ratings.

2 Exercise 3.2

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

Consider the example of a trader using reinforcement learning to buy and sell shares at the beginning of each day with the goal of maximising profit. A good trading strategy will take into account the prices of the shares over the last several days, perhaps even the last several months. Is this an example of a goal-directed learning task which the MDP framework does not usefully represent? Or can we include the historic share prices in the current state vector?

——————— Rafael:

I dont think it is a good example. The state can be anything you want, the idea of state is very abstract.

> The MDP framework is abstract and flexible and can be applied to many different problems in many different ways. For example, the time steps need not refer to fixed intervals of real time; they can refer to arbitrary successive stages of decision making and acting.

I think we need to think in cases where there are no win solutions, or when the reward is decoupled to the action. Maybe lottery. The result is completely decoupled of one's action. I dont know. I cannot think in any "no win" example

Non stationary examples too.

3 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?

The level will depend on the task. If the task has minimal external influence. One could define the actions in terms of the accelerator, steering wheel, and brake. If we think in terms of a self driving car, driving in a city with a lot of external players then maybe it is better to model everything at a much higher level.

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

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

Table 1: Transition table

5 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).

Equation 3.3 states:

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

For episode tasks there are two sets of states, non terminal states S and the terminal states S^+. The transition probability sums to one for state in S. However we need to consider the transition to the terminal state. Hence

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

6 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?

For the discounted case we have:

$$G_t = -\gamma^{T-t}.$$

In the continuing case the value is

$$\begin{aligned} G_t &= -\gamma^{T_1-t} - \gamma^{T_2-t} - \gamma^{T_3-t} - \dots, \\ &= -\sum_{\tau \in \mathcal{T}} \gamma^{\tau-t}, \end{aligned}$$

where $\mathcal{T} = \{T_1, T_2, T_3, \dots\}$ is the set of times after t at which the pole falls over.