Reinforcement Learning: An Introduction - Chapter 3

Thomas Hopkins

Exercise 3.1

1. Human-like robot*
   - States: the real-world environment and various sensor information about appendages
   - Actions: voltages applied to appendages
   - Rewards: pleasure/pain signal similar to that of a human
2. Video recommender system
   - States: database of available videos and a single user's profile
   - Actions: choose videos to display to the user
   - Rewards: user engagement (time watched, liked/disliked video, etc.)
3. Race car
   - States: distance to obstacles (from laser readings of the boundaries of the race track)
   - Actions: choose amount of acceleration, steer angle, braking, etc.
   - Rewards: -1 at each step, positive reward for passing checkpoints along the track, large negative reward for crashing

*Example of stretching the limits of this framework due to the difficult to design reward signal.

Exercise 3.2

This framework does not seem to be particularly suited to solving multi-task learning problems. Since there is only a single reward signal, $r_t \in \mathbb{R}$, the framework does not allow for a vector of rewards representing performance on a variety of tasks resulting from a single decision. Incorporating this kind of flexibility into the framework is somewhat possible by reducing the components of the reward vector into a single signal. For example, a weighted sum of reward components could work. However, this kind of method may not allow the agent to adequately *realize* the component effects of its decisions in the environment.

At the moment of writing this, I think that the reinforcement learning framework can adequately represent any goal-directed learning task with a single reward signal. It is as easy as setting the goal state to have $+1$ reward with zero everywhere else.

Exercise 3.3

The right place to draw the line would be depending on the goal of the agent. For example, deciding where to drive is a different goal than driving as safe and smooth as possible. This is the basis for determining the design of the problem in the reinforcement learning framework. The fundamental reason for preferring one location over another is in terms of the *amount of control* and *amount of feedback* that is present in the framework. Clearly, actions based on tire torques will have greater control of the car than actions based on where to go. Depending on the goal, the amount of feedback could be appropriate for either choice. However, if the goal is driving smooth then both of these representations could easily fail. Changing tire torques could have *too much* control which may cause erratic driving. Deciding where to drive (such as left, right, forward, etc.) could be *too little* control for smooth driving because rapid changes in steering angle could result from choosing to go left then right. The right choice for this task, in my (untested) opinion, would be to define actions in terms of the accelerator, steering angle, and brake. This would give the *right* amount of control in a smooth driving task since they were designed to be that way for human drivers.

Exercise 3.4

The return would always be -1 in this case. This is seen cearly from

$$R_T = r_T + \gamma r_{T-1} + \ldots + \gamma^T r_0 = r_T = -1$$

This differs from the continuing case since failures can happen repeatedly along the chain of rewards within the return. Let T an arbitrary moment of time for the continuing task, then

$$R_T = r_T + \gamma r_{T-1} + \ldots + \gamma^T r_0$$

where any $r_0, \ldots, r_T$ could be either -1 or 0. This is clearly different from the episodic task.

Exercise 3.5

The agent is having trouble finding the exit the first time. It might even be stuck trying options that it has already tried before since it is not being penalized for re-visiting locations in the maze. This reward signal is a failure to communicate what the goal is since it does not have a sense of where it has been before. Normally, the task of finding an exit to a maze involves searching possible paths through the maze *and* not searching the same path more than once. You could introduce a reward of -1 for each step to account for this as it will pressure the agent to explore other parts of the maze (eventually finding the exit).

Exercise 3.6

No, I would not have access to the Markov state of that environment because a single image alone is not sufficient to predict the next future states. For example, self-driving cars usually look back a few images in the past (or a few consecutive images at a time) in order to make accurate predictions about their future environment. If the camera were completely broken I would have access to *a* Markov state, being that the probability that the next image will not exist will be 1. This would not be the Markov state of the environment though.

Exercise 3.7

Let R denote the (finite) set of possible rewards, we have

$$\mathscr{P}_{ss'}^{a} = Pr[s_{t+1} = s'|s_t = s, a_t = a]$$

$$= \frac{Pr[s_{t+1} = s', r_{t+1} = r|s_t = s, a_t = a]}{Pr[r_{t+1} = r|s_{t+1} = s', s_t = s, a_t = a]}$$

$$\mathscr{R}_{ss'}^{a} = E[r_{t+1}|s_t = s, a_t = a, s_{t+1} = s']$$

$$= \sum_{r \in R} rPr[r_{t+1} = r|s_t = s, a_t = a, s_{t+1} = s']$$

$$= \sum_{r \in R} r \frac{Pr[s_{t+1} = s', r_{t+1} = r|s_t = s, a_t = a]}{\mathscr{P}_{ss'}^{a}}$$

This just uses Bayes' theorem to rewrite the equations using the Markov property.

Exercise 3.8

$$Q^{\pi}(s, a) = E_{\pi}[R_t|s_t = s, a_t = a]$$

$$= E_{\pi}[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|s_t = s, a_t = a]$$

$$= \sum_{s'} \mathscr{P}_{ss'}^{a}[\mathscr{R}_{ss'}^{a} + \gamma \sum_{a'} E_{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|s_{t+1} = s', a_{t+1} = a']]$$

$$= \sum_{s'} \mathscr{P}_{ss'}^{a}[\mathscr{R}_{ss'}^{a} + \gamma \sum_{a'} Q^{\pi}(s', a')\pi(s', a')]$$

Exercise 3.9

Here is the Bellman equation for $V^\pi(s)$

$$V^\pi(s) = \sum_a \pi(s,a) \sum_{s'} \mathscr{P}^a_{ss'}[\mathscr{R}^a_{ss'} + \gamma V^\pi(s')]$$

Now for the center state $s = (3,3)$, we can verify that the expected return is indeed 0.7. We have

$$V^\pi((3,3)) = 0.25(1(0 + (0.9)(0.7))) + 0.25(1(0 + (0.9)(2.3))) + 0.25(1(0 + (0.9)(0.4))) +$$

$$= 0.675$$

Rounded to one decimal place (since the $V^\pi(s')$ are only accurate to one decimal place) we have $V^\pi((3,3)) = 0.7$, as expected.

Exercise 3.10

The return, R_t, is defined as

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

Adding a constant C to each reward gives

$$R_t = \sum_{k=0}^{\infty} \gamma^k (r_{t+k+1} + C)$$

$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} + \gamma^k C$$

$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} + \frac{C}{1-\gamma}$$

$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} + K$$

that is, this amounts to adding a constant K to all of the state values where $K = \frac{C}{1-\gamma}$ (we observe that for $\gamma < 1$, the infinite sum is a geometric series which has a closed form that is finite).

Exercise 3.11

Since this is an episodic task, we now define an episode length T. So,

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1}$$

Adding a constant C to each reward gives

$$R_t = \sum_{k=0}^{T} \gamma^k (r_{t+k+1} + C)$$

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1} + \sum_{k=0}^{T} \gamma^k C$$

Since T can vary from episode to episode and each state may not be visited (or backed up) in a single episode, this constant changes the value of each state.

As an example, consider maze running with a $+1$ reward for finishing the maze and a -0.1 reward for every other step. If we added a constant $C = 5$ to this environment, then the agent would get stuck thinking actions it had taken previously were better than ones it has never tried. It would fail to explore the full maze, never find the exit, and never end the episode. This makes sense as never ending the episode would yield the largest return.

Exercise 3.12

$$V^\pi(s) = E_\pi[Q^\pi(s_t, a_t)|s_t = s]$$

$$V^\pi(s) = \sum_a \pi(s, a) Q^\pi(s, a)$$

Exercise 3.13

$$Q^\pi(s, a) = E[r_{t+1}|s_t = s, a_t = a] + E_\pi[V^\pi(s_{t+1})|s_t = s, a_t = a]$$

$$Q^\pi(s, a) = \sum_{s'} (\mathscr{P}_{ss'}^a \mathscr{R}_{ss'}^a + \mathscr{P}_{ss'}^a V^\pi(s')) = \sum_{s'} \mathscr{P}_{ss'}^a [\mathscr{R}_{ss'}^a + V^\pi(s')]$$

Exercise 3.14

This has the value of using the driver from outside the green, then using the putter when in the green. These are the best actions so these are the optimal values for the states.

Exercise 3.15

This example is more interesting, since we are forced to use the putter at each state we are actually in, this makes our immediate reward smaller when we are far out. The countours will be similar to the ones for Exercise 3.14 except they will each be one lower since we are forced to putt once. In the sand, it will be -3 since we putt once then use the driver to get to the green, the sink the ball in using the putter (3 strokes).

Exercise 3.16

$$Q^*(h,s) = \mathscr{P}^s_{hh}[\mathscr{R}^s_{hh} + \gamma \max_{a'} Q^*(h,a')] + \mathscr{P}^s_{hl}[\mathscr{R}^s_{hl} + \gamma \max_{a'} Q^*(l,a')]$$

$$= \alpha[\mathscr{R}^s + \gamma \max_{a'} Q^*(h,a')] + (1-\alpha)[\mathscr{R}^s + \gamma \max_{a'} Q^*(l,a')]$$

$$Q^*(h,w) = \mathscr{P}^w_{hh}[\mathscr{R}^w_{hh} + \gamma \max_{a'} Q^*(h,a')] = \mathscr{R}^w + \gamma \max_{a'} Q^*(h,a')$$

$$Q^*(l,s) = \mathscr{P}^s_{lh}[\mathscr{R}^s_{lh} + \gamma \max_{a'} Q^*(h,a')] + \mathscr{P}^s_{ll}[\mathscr{R}^s_{ll} + \gamma \max_{a'} Q^*(l,a')]$$

$$= (1-\beta)[-3 + \gamma \max_{a'} Q^*(h,a')] + \beta[\mathscr{R}^s + \gamma \max_{a'} Q^*(l,a')]$$

$$Q^*(l,w) = \mathscr{P}^w_{ll}[\mathscr{R}^w_{ll} + \gamma \max_{a'} Q^*(l,a')] = \mathscr{R}^w + \gamma \max_{a'} Q^*(l,a')$$

$$Q^*(l,re) = \mathscr{P}^{re}_{lh}[\mathscr{R}^{re}_{lh} + \gamma \max_{a'} Q^*(h,a')] = \gamma \max_{a'} Q^*(h,a')$$

Exercise 3.17

Since it takes 5 steps to get back to state A, we have

$$V^*(A) = \sum_{k=0}^{\infty} 10\gamma^{5k} = \frac{10}{1-\gamma^5} = \frac{10}{1-(0.9)^5} = 24.419$$