

Reinforcement Learning: An Introduction

Attempted Solutions

Chapter 4

Scott Brownlie & Rafael Rui

1 Exercise 4.1

In Example 4.1, if π is the equiprobable random policy, what is $q_\pi(11, \text{down})$. What is $q_\pi(7, \text{down})$?

As moving downwards from 11 results in the terminal state, $q_\pi(11, \text{down}) = -1$. Moving down from 7 leaves the agent in state 11, so

$$q_\pi(7, \text{down}) = -1 + v_\pi(11) = -1 - 14 = -15.$$

2 Exercise 4.2

In Example 4.1, suppose a new state 15 is added to the gridworld just below state 13, and its actions, left, up, right, and down, take the agent to states 12, 13, 14, and 15, respectively. Assume that the transitions from the original states are unchanged. What, then, is $v_\pi(15)$ for the equiprobable random policy? Now suppose the dynamics of state 13 are also changed, such that action down from state 13 takes the agent to the new state 15. What is $v_\pi(15)$ for the equiprobable random policy in this case?

When the transitions from the original states are unchanged we have

$$\begin{aligned} v_\pi(15) &= -1 + 0.25(v_\pi(12) + v_\pi(13) + v_\pi(14) + v_\pi(15)) \\ &= -1 + 0.25(-22 - 20 - 14) + 0.25 \cdot v_\pi(15) \\ \iff 0.75 \cdot v_\pi(15) &= -15 \\ \iff v_\pi(15) &= -20. \end{aligned}$$

Now suppose that the dynamics of state 13 are changed such that action down from state 13 takes the agent to the new state 15. Since $v_\pi(15) = -20 = v_\pi(13)$ in the case of unchanged dynamics, $v_\pi(15)$ should remain -20 . We can formally prove this:

$$\begin{aligned} v_\pi(15) &= -1 + 0.25(v_\pi(12) + v_\pi(13) + v_\pi(14) + v_\pi(15)) \\ &= -1 + 0.25(-22 - 14) + 0.25 \cdot v_\pi(13) + 0.25 \cdot v_\pi(15) \\ &= -10 + 0.25 \cdot v_\pi(13) + 0.25 \cdot v_\pi(15), \end{aligned}$$

where

$$\begin{aligned} v_\pi(13) &= -1 + 0.25(v_\pi(9) + v_\pi(12) + v_\pi(14) + v_\pi(15)) \\ &= -1 + 0.25(-20 - 22 - 14) + 0.25 \cdot v_\pi(15) \\ &= -15 + 0.25 \cdot v_\pi(15). \end{aligned}$$

Hence,

$$\begin{aligned}
v_\pi(15) &= -10 + 0.25(-15 + 0.25 \cdot v_\pi(15)) + 0.25 \cdot v_\pi(15) \\
&= -13.75 + 0.3125 \cdot v_\pi(15) \\
\iff 0.6875 \cdot v_\pi(15) &= -13.75 \\
\iff v_\pi(15) &= -20.
\end{aligned}$$

3 Exercise 4.3

What are the equations analogous to (4.3), (4.4), and (4.5) for the action-value function q_π and its successive approximation by a sequence of functions q_0, q_1, q_2, \dots ?

We have

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

and

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

4 Exercise 4.4

The policy iteration algorithm on page 80 has a subtle bug in that it may never terminate if the policy continually switches between two or more policies that are equally good. This is ok for pedagogy, but not for actual use. Modify the pseudocode so that convergence is guaranteed.

In part 3 of the algorithm, instead of checking if the policy is stable, we should check if the policy has improved as follows:

```

policy-improved  $\leftarrow$  false
For each  $s \in \mathcal{S}$ 
   $v \leftarrow V(s)$ 
   $\pi(s) \leftarrow \operatorname{argmax}_a \sum_{s', r} p(s', r | s, a) [r + \gamma V(s')]$ 
   $v_{\text{new}} \leftarrow \sum_{s', r} p(s', r | s, \pi(s)) [r + \gamma V(s')]$ 
  If  $v_{\text{new}} > v$ , then policy-improved  $\leftarrow$  true
If policy-improved, then go to 2, else stop and return  $V \approx v_*, \pi \approx \pi_*$ .

```

5 Exercise 4.5

How would policy iteration be defined for action values? Give a complete algorithm for computing q_* , analogous to that on page 80 for computing v_* . Please pay special attention to this exercise, because the ideas involved will be used throughout the rest of the book.

The algorithm is

1. Initialisation

$Q(s, a) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$

2. Policy Evaluation

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in \mathcal{S}$:

Loop for each $a \in \mathcal{A}(s)$:

$q \leftarrow Q(s, a)$

$Q(s, a) \leftarrow \sum_{s', r} p(s', r | s, a) [r + \gamma Q(s', \pi(s'))]$

$\Delta \leftarrow \max(\Delta, |q - Q(s, a)|)$

until $\Delta < \theta$

3. Policy Improvement

policy-stable \leftarrow *true*

For each $s \in \mathcal{S}$:

old-action $\leftarrow \pi(s)$

$\pi(s) \leftarrow \operatorname{argmax}_a \sum_{s', r} p(s', r | s, a) [r + \gamma Q(s', \pi(s'))]$

If *old-action* $\neq \pi(s)$, then *policy-stable* \leftarrow *false*

If *policy-stable*, then stop and return $Q \approx q_*, \pi \approx \pi_*$, else go to 2.

6 Exercise 4.6

Suppose you are restricted to considering only policies that are ϵ -soft, meaning that the probability of selecting each action in each state, s , is at least $\epsilon/|\mathcal{A}(s)|$. Describe qualitatively the changes that would be required in each of the steps 3, 2, and 1, in that order, of the policy iteration algorithm for v_* on page 80.

As the policy is now non-deterministic, for each state $s \in \mathcal{S}$ we must define a probability distribution over the actions $a \in \mathcal{A}(s)$ such that $\pi(s, a) \geq \epsilon/|\mathcal{A}(s)|$ for all $a \in \mathcal{A}(s)$.

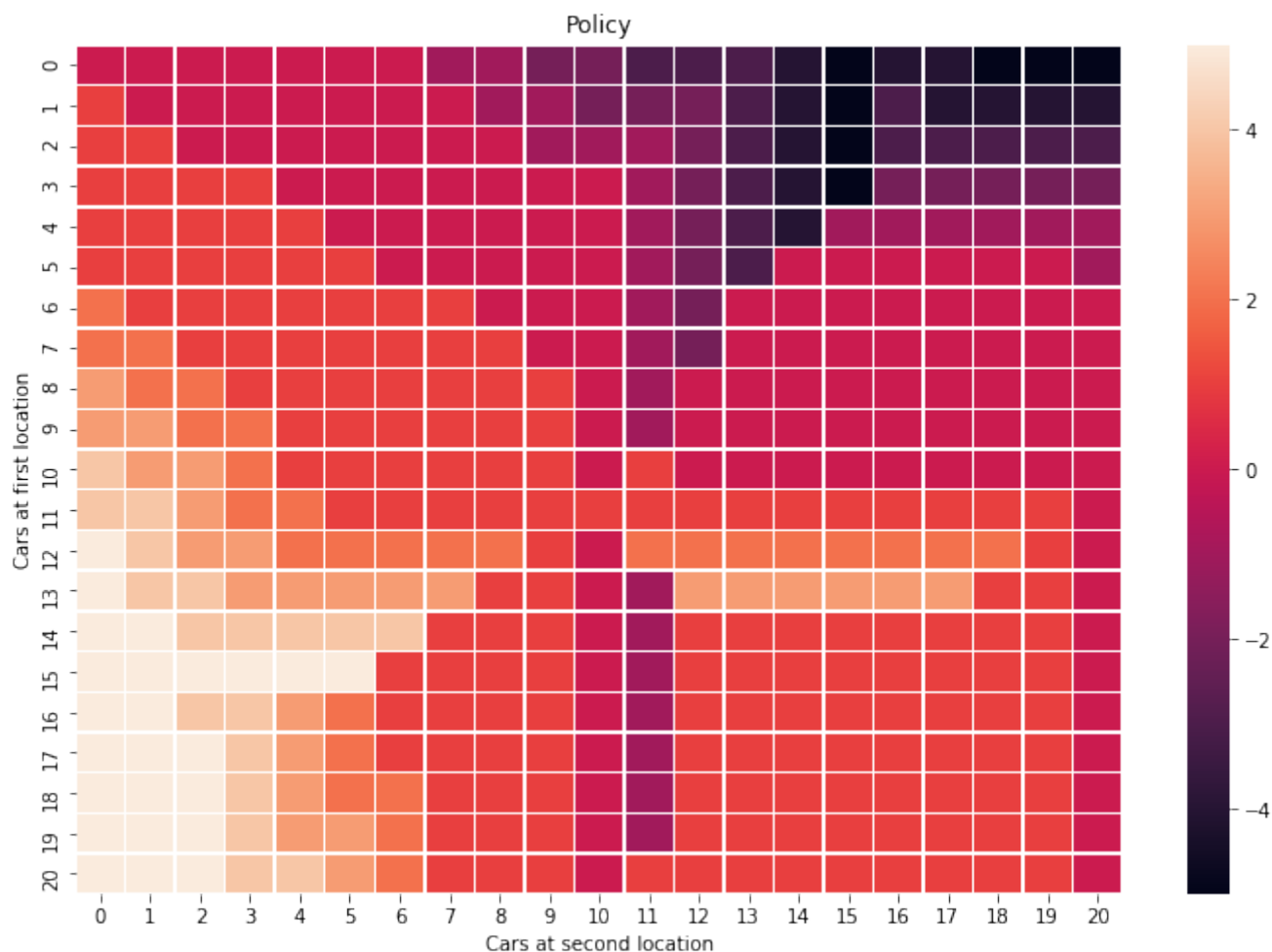
In step 3, *old-action* now becomes a distribution as opposed to a single action, and the policy is improved by finding the distribution which maximises the expected return with respect to the current value function, where the expectation is taken over the actions. This can be solved using linear programming. The policy is stable if the distribution does not change for any $s \in \mathcal{S}$, subject to some small tolerance.

In step 2, when estimating the value of each state $s \in \mathcal{S}$ we need to compute its expectation over all possible actions given the current probability distribution over $a \in \mathcal{A}(s)$. Finally, in step 1 we need to initialise $\pi(s, a) \geq \epsilon/|\mathcal{A}(s)|$ for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$.

7 Exercise 4.7

The optimal policy and value function for this problem are shown in Figures 1 and 2 respectively.

Figure 1: Optimal policy for Exercise 4.7



8 Exercise 4.8

Why does the optimal policy for the gambler's problem have such a curious form? In particular, for capital of 50 it bets it all on one flip, but for capital of 51 it does not. Why is this a good policy?

Whenever the gambler's capital is at least 50, if he bets it all then he either reaches his goal with probability p_h or loses everything with probability $1 - p_h$. Thus, if he were to follow this policy then there would be no advantage to having a capital of 99 as opposed to 50. Clearly he can do better.

When he has 51, if he bets 1 then with probability p_h his new capital is 52 and with probability $1 - p_h$ his new capital is 50. Compared to the policy of always betting the full capital, this policy of only betting 1 when he has capital of 51 can be seen as a *free hit* at moving closer to his goal, because he will either increase his capital by 1 or, in the worst case, he is down to 50 and he can then stake everything on the next bet. The policy is clearly more sensible than betting the full 51.

9 Exercise 4.9

10 Exercise 4.10

Figure 2: Optimal value function for Exercise 4.7

