3 Dynamic Programming

22. Exercise 4.1 In Example 4.1, if π is the equiprobable random policy, what is q_π(11, down)?

As the terminal state's action value is 0, q_π(11, down) can be calculated as:

$$\begin{aligned} q_\pi(11, down) &= p(s', r \mid s, a)[r] \\ &= 1 \cdot [-1] = -1 \end{aligned}$$

23. 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?

Using the general formula

$$v_\pi(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_\pi(s')]$$

$v_\pi(15)$ can be calculated as

$$\begin{aligned} v_\pi(15) = &\frac{1}{4} p(12, -1 \mid 15, left)[-1 + \gamma v_\pi(12)] + \\ &\frac{1}{4} p(13, -1 \mid 15, up)[-1 + \gamma v_\pi(13)] + \\ &\frac{1}{4} p(14, -1 \mid 15, right)[-1 + \gamma v_\pi(14)] + \\ &\frac{1}{4} p(15, -1 \mid 15, down)[-1 + \gamma v_\pi(15)] \\ &= -1 + \frac{1}{4} v_\pi(12) + \frac{1}{4} v_\pi(13) + \frac{1}{4} v_\pi(14) + \frac{1}{4} v_\pi(15) \end{aligned}$$

25. 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 ,... ? (Textbook p. 74)

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

$$q_{k+1}(s, a) = \sum_{s',r} p(s', r \mid s, a)[r + \gamma q_k(s', a')]$$

26. 6. 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.

$$\pi_0 \xrightarrow{pe} q_{\pi_0} \xrightarrow{pi} \pi_1 \xrightarrow{pe} q_{\pi_1} \cdots \xrightarrow{pi} \pi_* \xrightarrow{pe} q_*$$

Algorithm 1: Policy iteration

1. Initialization
$\pi \leftarrow$ arbitrary deterministic policy
$Q \leftarrow$ arbitrary
$\theta \leftarrow$ small positive number

2. Policy evaluation
while $\Delta < \theta$ **do**
  for *each* $s \in \mathcal{S}$ *and* $a \in \mathcal{A}(s)$ **do**
    $q \leftarrow Q(s,a)$
    $q(s,a) \leftarrow \sum_{s',r} p(s', r \mid s, a)[r + \gamma q_\pi(s', a)]$
    $\Delta \leftarrow max(\Delta, \mid q - Q(s,a) \mid)$
  end
end

3. Policy improvement
$improved \leftarrow false$
for *each* $s \in \mathcal{S}$ **do**
  $b \leftarrow \pi_s$
  $\pi_s \leftarrow argmax_a Q(s,a)$
  if $b \neq \pi(s)$ **then**
    $improved \leftarrow true$
  end
end
if *improved* **then**
  Goto *Policy evaluation*
end

27. 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 $\epsilon/|A(s)|$. 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_* (Textbook p. 80).

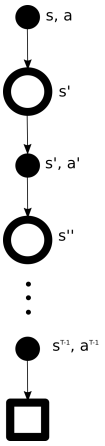
Step 3:
The update of the policy would have to be changed, such that with a probability of $\epsilon/|A(s)|$ a non-greedy action is taken and the greedy action is taken with the remaining probability.
The check for stability of the policy, i.e. the comparison to *old-action*, would have to adapted, such that for one does compare the actions with the highest probabilities.
Step 2:
In the policy evaluation, the update to the state-value function would have to be changed, s.t. it is calculated in a way similar to equation 4.5 in Sutton and Barto [2018].
Step 1: The initialization of the policy would have to be changed to create an ϵ-soft policy.

28. Exercise 4.10 What is the analog of the value iteration update (4.10) for action values, $q_{k+1}(s, a)$?

$$\begin{aligned} q_{k+1}(s,a) &= \mathbb{E}[r_{t+1} + \gamma max_{a'} q_k(s_{t+1}, a) \mid s_t = s, a_t = a] \\ &= \sum_{s',r} p(s', r \mid s, a)[r + \gamma max_{a'} q_k(s', a')] \end{aligned}$$

4 Monte Carlo Methods

29. Exercise 5.3 What is the backup diagram for Monte Carlo estimation of q_π?



30. Exercise 5.9 Modify the algorithm for first-visit MC policy evaluation (Section 5.1) to use the incremental implementation for sample averages described in Section 2.4.

Algorithm 2: First-visit MC prediction with incremental update

Input: a policy π to be evaluated
Initialize:
$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
$N(s) \leftarrow 0$, for all $s \in \mathcal{S}$
for *ever (for each episode)* **do**
  Generate an episode following π : $S_0, A_0, R_1, S_1, A_1, R_2, ..., S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, ..., 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless S_t appears in $S_0, S_1, ..., S_{t-1}$:
      $N(S_t) \leftarrow N(S_t) + 1$
      $V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}[G - V(S_t)]$
end

31. Exercise 5.10 Derive the weighted-average update rule (5.8) from (5.7). Follow the pattern of the derivation of the unweighted rule (2.3). (Textbook p. 109)

$$
\begin{aligned}
V_n &= \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \qquad n \to n+1 \\
V_{n+1} &= \frac{\sum_{k=1}^{n} W_k G_k}{\sum_{k=1}^{n} W_k} \\
&= \frac{1}{\sum_{k=1}^{n} W_k} \left[\sum_{k=1}^{n} W_k G_k \right] \\
&= \frac{1}{\sum_{k=1}^{n} W_k} \left[W_n G_n + \sum_{k=1}^{n-1} W_k G_k \right] \\
&= \frac{1}{\sum_{k=1}^{n} W_k} \left[W_n G_n + \frac{\sum_{k=1}^{n-1} W_k}{\sum_{k=1}^{n-1} W_k} \sum_{k=1}^{n-1} W_k G_k \right] \\
&= \frac{1}{\sum_{k=1}^{n} W_k} \left[W_n G_n + V_n \sum_{k=1}^{n-1} W_k \right] \\
&= \frac{1}{\sum_{k=1}^{n} W_k} \left[W_n G_n + V_n \sum_{k=1}^{n} W_k - V_n W_n \right] \\
&= V_n + \frac{W_n G_n}{\sum_{k=1}^{n} W_k} - \frac{V_n W_n}{\sum_{k=1}^{n} W_k} \\
&= V_n + \frac{W_n}{\sum_{k=1}^{n} W_k} \left[G_n - W_n \right] \\
&= V_n + \frac{W_n}{C_n} \left[G_n - W_n \right], \qquad \text{with } C_n = \sum_{k=1}^{n} W_k
\end{aligned}
$$

33. Exercise 5.13 Show the steps to derive (5.14) from (5.12). (Textbook p. 114)

34. Exercise 5.14 Modify the algorithm for off-policy Monte Carlo control (page 110) to use the idea of the truncated weighted-average estimator (5.10). Note that you will first need to convert this equation to action values.

References

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL `http://incompleteideas.net/book/the-book-2nd.html`.