

# Dynamic Programming

ECE/CSE RL Monsoon 2019

Sep 30, 2020

**Question 1.** Solve Exercise 3.4. Explain how you obtained the table. Your solution may be hand-written.

**Question 2.** Write code that solves the linear equations required to find  $v_\pi(s)$  and generate the values in the table in Figure 3.2. Note that the policy  $\pi$  picks all valid actions in a state with equal probability. Add comments to your code that explain all your steps.

**Question 3.** Solve Exercises 3.15 and 3.16.

**Question 4.** Write code that generates the optimal state-value function and the optimal policy for the Gridworld in Figure 3.5. You want to solve the corresponding system of non-linear equations. Explain all your steps.

**Question 5.** Given an equation for  $v_*$  in terms of  $q_*$ .

**Question 6.** Code policy iteration and value iteration (VI) to solve the Gridworld in Example 4.1. Your code must log output of each iteration. Pick up a few sample iterations to show policy evaluation and improvement at work. Similarly, show using a few obtained iterations that every iteration of VI improves the value function. Your code must include the fix to the bug mentioned in Exercise 4.4.

**Question 7.** Code exercise 4.7.

**Question 8.** When we defined a Markov Decision Process (MDP) we explicitly captured, using probability mass functions (PMF), the fact that the random variable  $R_{t+1}$  is dependent on the state  $S_t$  and the action  $A_t$ . Is the random variable  $R_{t+2}$  dependent on  $S_t$  and  $A_t$ ? Support your answer using the PMFs we used to define a MDP. [Hint: Start with the conditional PMF of  $R_{t+2}$  conditioned on  $S_t$  and  $A_t$ .]

**Question 9.** Derive the expression for  $E[R_{t+2}|S_t = s, A_t = a]$  in terms of the PMF(s) that define a MDP. [Hint: This is in a way an extension of Question 8.]

**Question 10.** We know that the state-value function  $v_\pi(s) = E[G_t|S_t = s]$ . Use this definition of  $v_\pi(s)$  to derive the Bellman equation for  $v_\pi(s)$  for all  $s \in S$ . The Bellman equation will use the PMF corresponding to the policy  $\pi$  and the PMF  $p(s', r|s, a)$  and will provide a recursive method of calculating  $v_\pi(s)$ .

**Question 11.** Suppose an agent receives the sequence of rewards  $R_1 = 2$ ,  $R_2 = -1$ ,  $R_3 = 10$ , and  $R_4 = -3$ . Calculate the  $\gamma$  discounted return/reward for each time step for  $\gamma = 0.5$ . Also show that, if an agent receives a constant reward  $c$ , at every time step, for  $\gamma < 1$ , the infinite horizon discounted return is given by

$$G_t = \frac{c}{1 - \gamma}.$$

**Question 12.** Suppose you are given the optimal state-value function  $v_*(s)$ , for all  $s \in S$ . How will you find the optimal policy? Show your steps.

**Question 13.** A server requires information from a sensor. The server would like the information to be fresh. However, there is a cost to querying information from the sensor. Specifically, the state at the server can

be either fresh or stale. The former indicates that the information at the server about the sensor is fresh and the latter indicates that it is stale. At any time, the server may choose to query or remain silent. A query makes stale information fresh with probability 0.8 and fresh information stale with probability 0.1. Staying silent keeps stale information stale with probability 1 and makes fresh information stale with probability 0.5. Also, a query when current information at the server is stale costs 8 dollars. Otherwise, it costs 4 dollars. Staying quiet has a reward of 4 dollars.

1. Draw and/or tabulate the MDP.
2. Consider a finite horizon problem in which the server gets to optimize costs over three time steps. Assume that if at the end of the three time steps the server has stale information it pays a cost of 10, else it receives a reward of 10. Also, assume that future costs are discounted with a factor  $1/2$ . Calculate the optimal costs for every starting state. Calculate the optimal policy. Show all your steps. Guesses and intuition will bring you unbounded costs.
3. Suppose the server is to optimize for ever and is looking for a stationary policy. Write down the first four iterations of value iteration that will help the server get closer to a stationary policy. Do the same for policy iteration (where one iteration includes evaluation and improvement). Clearly identify each iteration.

**Question 14.** Assume an infinite horizon discounted costs problem. Prove that the policy improvement step either improves the current policy or the current policy is optimal. Show all your steps and support them using the properties of dynamic programming.

**Question 15.** *The Stairwell:* A stairwell connects the ground floor (G) of a palace with its first floor (F). It consists of a total of  $n$  steps, indexed  $1, 2, \dots, n$ , with step 1 close to G and step  $n$  connected to F. A robot may start in any step and must learn to decide on whether it should go to F or to G. Both F and G are terminal states. The robot gets a reward of 2 with probability 0.5, and 0 otherwise, when it descends from step  $i$  to  $i - 1$ ,  $1 < i \leq n$ . It also gets a reward of 1 on going from step 1 to G. On the other hand, the robot pays a cost of 1 (reward  $-1$ ) when it moves from step  $i$  to  $i + 1$ ,  $1 \leq i < n$ . Finally, the robot gets a bumper reward of  $2n$  when it goes from step  $n$  to F. The robot can only take one step at a time.

Start with a policy in which the robot goes up or down a step in the stairwell with probability 0.5. Draw the corresponding MDP for  $n = 2$ . Use policy iteration to find the optimal policy for  $n = 2$ . Show all your iterations and explain them. How does the number of iterations increase with  $n$ ? Explain your answer. Assume a discount factor of 1.