CS 4180/5180: Reinforcement Learning and Sequential Decision Making (Fall 2021) Robert Platt
Northeastern University Due October 1, 2021

Exercise 2: Markov Decision Processes

Please remember the following policies:

- Exercises are due at **11:59 PM** Boston time (EDT/EST).
- Submissions should be made electronically on Canvas. Please ensure that your solutions for both the written and programming parts are present. You can upload multiple files in a single submission, or you can zip them into a single file. You can make as many submissions as you wish, but only the latest one will be considered, and late days will be computed based on the latest submission.
- If you are unable to access Canvas, you may submit via the submission link on Piazza. In this case, please zip your submission into a single file, and follow the naming convention listed on the form:
  `Ex[Num] - [FirstName] [LastName] [Version number].zip`
- Solutions may be handwritten or typeset. For the former, please ensure handwriting is legible. If you write your answers on paper and submit images of them, that is fine, but please put and order them correctly in a single .pdf file. One way to do this is putting them in a Word document and saving as a PDF file.
- You are welcome to discuss these problems with other students in the class, but you must understand and write up the solution **and code** yourself, *and* indicate who you discussed with (if any).
- Some questions are intended for CS 5180 students only. CS 4180 students may complete these for extra credit.
- Each exercise may be handed in up to two days late (24-hour period), penalized by 5% per day. Submissions later than this will not be accepted. There is no limit on the total number of late days used over the course of the semester.
- Contact the teaching staff if there are *extenuating* circumstances.

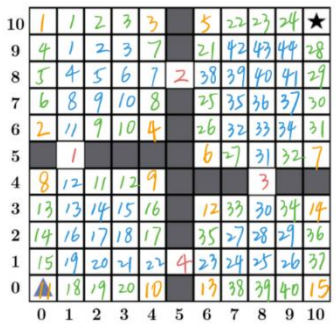
1. **1 point.** *Formulating an MDP.*
   Written: It is instructive to formally define an MDP at least once, in particular, the dynamics function. Consider the four-rooms domain from Ex0.

   (a) What are the state and action spaces S, A?

S (state spaces) are a set of positions in this 11 by 11 matrix (from point (0, 0) to (10, 10)) except the walls.

A (action spaces) are actions selectable from each location, including left, right, up and down.

(b) Consider the dynamics function $p(s', r|s, a)$. Approximately how many non-zero rows are in this conditional probability table?



From the graph showing in the left, there are 44 blue blocks, 40 green blocks, 15 orange blocks and 4 red blocks. They represent number of directions that each state could move in that block to avoid being hit by the walls. (B=4 directions, G=3, O=2 and R=2). (10, 10), the star position is not being considered as it arrives at the final position then the agent would go back to the original point.

$p(s', r|s, a)$ as non-zero rows could be considered as the possible actions from each s to s' as r is always 0 expect the reward is 1 for agent arrive at the final position.

Therefore, for all s in B region, possible actions could be 4*3(4 directions * 3 selections/direction), 44*4*3= 528. For all s in G region, possible actions could be 4*3(including stay when hitting the walls), 40*3*4 = 480. For all s in O region, possible action could be 2*3(not hitting walls)+2*2(hitting walls), 15*(2*3+2*2) = 150. For all s in R region, possible action could be 4*3, 4*4*3=48. So the total estimation for the conditional probability table is 528+480+150+48=1206. Approximately, 1200 non-zero rows exist.

2. **1 point.** (RL2e 3.6, 3.7) *The RL objective.*
Written:

(a) 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? (Derive expressions for both the episodic and continuing cases.)

Followed by Equation 3.7

An episodic task with discounting is:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$$
$$= \sum_{i=0}^{T-t-1} \gamma^i R_{t+i+1}$$

reward is -1 for failure, $R_T = -1$

so the update return is $-\gamma^{T-t-1}$

An continuing task with discounting is:

the update return is $-\gamma^K$ as K is the time step before failure

(b) Imagine that you are designing a robot to run a maze. You decide to give it a reward of $+1$ for escaping from the maze and a reward of zero at all other times. The task seems to break down naturally into episodes – the successive runs through the maze – so you decide to treat it as an episodic task, where the goal is to maximize expected total reward (Equation 3.7). There is no discounting, i.e. $\gamma = 1$. After running the learning agent for a while, you find that it is showing no improvement in escaping from the maze. What is going wrong? Have you effectively communicated to the agent what you want it to achieve?

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T, \tag{3.7}$$

Without using γ to implement the discount, the maximum return is always +1 whatever time step from the maze. To effectively communicate to the agent, we need to add punishment or cost with each time step before escape.

3. **1 point.** (RL2e 3.8, 3.9) *Discounted return.*
Written:

(a) Suppose $\gamma = 0.5$ and the following sequence of rewards is received:
$R_1 = -1, R_2 = 2, R_3 = 6, R_4 = 3, R_5 = 2$
with $T = 5$. What are $G_0, G_1, \ldots, G_5$? Hint: Work backwards.

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \\ &= R_{t+1} + \gamma\left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \cdots\right) \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned} \tag{3.9}$$

$G_5 = 0$

$G_4 = R_5 + \gamma G_5 = 2 + 0.5(0) = 2$

$G_3 = R_4 + \gamma G_4 = 3 + 0.5(2) = 4$

$G_2 = R_3 + \gamma G_3 = 6 + 0.5(4) = 8$

$G_1 = R_2 + \gamma G_2 = 2 + 0.5(8) = 6$

$G_0 = R_1 + \gamma G_1 = -1 + 0.5(6) = 2$

(b) Suppose $\gamma = 0.9$ and the reward sequence is $R_1 = 2$ followed by an infinite sequence of '7's. What are G_1 and G_0?

$$G_1 = \sum_{k=0}^{\infty} \gamma^k * R = R\frac{1}{1-\gamma} = 7 * \frac{1}{1-0.9} = 7 * 10 = 70$$

$G_0 = R_1 + \gamma G_1 = 2 + 0.9(70) = 2 + 63 = 65$

4. **1 point.** (Artificial Intelligence: A Modern Approach 17.9.) *Discount factor.*

r	-1	+10
-1	-1	-1
-1	-1	-1

(a)

+50	-1	-1	-1	...	-1	-1	-1	-1
Start				...				
-50	+1	+1	+1	...	+1	+1	+1	+1

(b)

Written: Consider the 101 × 3 world shown above (right; the left sub-figure will not be used). In the start state the agent has a choice of two deterministic actions, `Up` or `Down`, but in the other states the agent has one deterministic action, `Right`. Express the value of each action as a function of the discount factor γ. For what values of the discount factor γ should the agent choose `Up` and for which `Down`? It is fine to leave your answer as an unsolved expression, although you can solve for the threshold value of γ numerically using tools such as WolframAlpha.
(This simple example reflects real-world situations in which one must weigh the value of an immediate action versus the potential continual long-term consequences, such as choosing to dump pollutants into a lake.)

If $\gamma = 1$, we should choose "down" action.

If $\gamma \neq 1$

$$G_{up} = 50 + \sum_{i=1}^{100} \gamma^i (-1) = 50 - \sum_{i=1}^{100} \gamma^i$$

$$G_{down} = -50 + \sum_{i=1}^{100} \gamma^i$$

$$G_{up} - G_{down} = (50 - \sum_{i=1}^{100} \gamma^i) - (-50 + \sum_{i=1}^{100} \gamma^i)$$

$$= 50 - \sum_{i=1}^{100} \gamma^i + 50 - \sum_{i=1}^{100} \gamma^i$$

$$= 100 - 2\sum_{i=1}^{100} \gamma^i > 0$$

If $100 - 2\sum_{i=1}^{100} \gamma^i > 0$

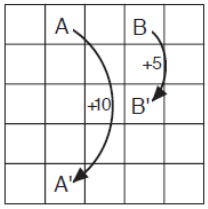
$$\sum_{i=1}^{100} \gamma^i > 50$$

$$\frac{\gamma^{100} - 1}{\gamma - 1} > 50$$

when $\frac{\gamma^{100} - 1}{\gamma - 1} > 50$, we should choose "up" action.

5. **1 point.** (RL2e 3.15, 3.16) *Modifying the reward function.*
Written:

(a) In the gridworld example (Figure 3.2 in RL2e), rewards are positive for goals, negative for running into the edge of the world, and zero the rest of the time. Are the signs of these rewards important, or only the intervals between them? Prove, using Equation 3.8, that adding a constant c to all the rewards adds a constant, v_c, to the values of all states, and thus does not affect the relative values of any states under any policies. What is v_c in terms of c and γ?




Actions

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

Figure 3.2: Gridworld example: exceptional reward dynamics (left) and state-value function for the equiprobable random policy (right).

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \tag{3.8}$$

$$V_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\Big]$$

$$V_{\pi c}(s) = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1+c} \mid S_t = s\Big]$$

$$= \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\Big] + \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k c \mid S_t = s\Big]$$

$$= V_\pi(s) + \sum_{k=0}^{\infty} \gamma^k c$$

$$= V_\pi(s) + c\frac{1}{1-\gamma}$$

$$V_c(s) = V_{\pi c}(s) - V_\pi(s) = \frac{c}{1-\gamma}$$

(b) Now consider adding a constant c to all the rewards in an episodic task, such as maze running. Would this have any effect, or would it leave the task unchanged as in the continuing task above? Why or why not? Give an example.

Reward is important in an episodic task. By taking negative reward, the agent would finish the task faster. Thus, adding a constant C would change the task, especially when the constant changes the sign of the equation. If the negative reward is still negative but smaller, it may change the direction of an agent. But in some circumstance, the task may be unchanged.

6. **1 point.** (RL2e 3.14) *Bellman equation.*
Written:

(a) The Bellman equation (Equation 3.14) must hold for each state for the value function v_π shown in Figure 3.2 (right) of Example 3.5. Show numerically that this equation holds for the center state, valued at $+0.7$, with respect to its four neighboring states, valued at $+2.3, +0.4, -0.4, +0.7$. (These numbers are accurate only to one decimal place.) Note that Figure 3.2 (right) is the value function for the equiprobable random policy.

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] && \text{(by (3.9))} \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r \mid s, a) \Big[r + \gamma \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s'] \Big] \\
&= \sum_a \pi(a|s) \sum_{s', r} p(s', r \mid s, a) \Big[r + \gamma v_\pi(s') \Big], \quad \text{for all } s \in \mathcal{S}, && (3.14)
\end{aligned}
$$

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

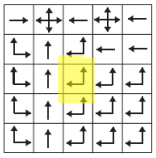
$$v_\pi(s) = \frac{1}{4} * 1 * (0 + 0.9 * 2.3) + \frac{1}{4} * 1 * (0 + 0.9 * 0.7) + \frac{1}{4} * 1 * (0 + 0.9 * 0.4) + \frac{1}{4} * 1 * (0 + 0.9 * (-0.4)) = 0.675 \approx 0.7$$

(b) The Bellman equation holds for *all* policies, including optimal policies. Consider v_* and π_* shown in Figure 3.5 (middle, right respectively). Similar to the previous part, show numerically that the Bellman equation holds for the center state, valued at $+17.8$, with respect to its four neighboring states, for the optimal policy π_* shown in Figure 3.5 (right).

If the Bellman equation is unclear to you, you should practice this question again for other states.

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

v_*



π_*

$$v_\pi(s) = \frac{1}{2} * 1 * (0 + 0.9 * 19.8) + \frac{1}{2} * 1 * (0 + 0.9 * 19.8) = 17.82 \approx 17.8$$

7. **1 point.** *Guessing and verifying value functions.*

Written: We have not discussed algorithms for finding the value function yet, but it is possible to guess them for simple problems and verify that they are consistent with the Bellman equation. A purported value function $V(s)$ is consistent (according to the Bellman equation) for all states under the policy π *if and only if* $V = v_\pi$, the unique value function for π.

(a) Consider the simple 3-state MDP shown above (left). All episodes start in the center state, A, then proceed either left or right by one state on each step, with equal probability. Episodes terminate either on the extreme left (L) or the extreme right (R). When an episode terminates on the right, a reward of $+1$ occurs; all other rewards are zero. This is an undiscounted MDP ($\gamma = 1$). Guess the value function for this MDP (for the equiprobable random policy), and verify its consistency using the Bellman equation.

Assume all actions happened with equal probability,

$V_\pi(s) = \frac{1}{2} \times 1 + \frac{1}{2} \times 0 = \frac{1}{2}$

According to Bellman Equation,

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

$$= \frac{1}{2} \times 1 \times 1 + \frac{1}{2} \times 1 \times 0 = \frac{1}{2}$$

The consistency is verified.

(b) Now consider an extension of the MDP to contain 7 states total (right). Guess the value function for this MDP (for the equiprobable random policy), and verify its consistency using the Bellman equation.
Hint: The new value function has a simple form. Use part (a) to inform your thinking.

The value function I guess is $V_\pi(A) = \frac{1}{6}$, $V_\pi(B) = \frac{2}{6}$, $V_\pi(C) = \frac{3}{6}$, $V_\pi(D) = \frac{4}{6}$, $V_\pi(E) = \frac{5}{6}$.

According to Bellman Equation:

$V_\pi(A) = \frac{1}{2} \times 1 \times [0 + 1 \times V_\pi(L)] + \frac{1}{2} \times 1 \times [0 + 1 \times V_\pi(B)] = \frac{1}{2} V_\pi(B)$

$V_\pi(B) = \frac{1}{2} \times 1 \times [0 + 1 \times V_\pi(A)] + \frac{1}{2} \times 1 \times [0 + 1 \times V_\pi(C)] = \frac{1}{2} V_\pi(A) + \frac{1}{2} V_\pi(C)$

$V_\pi(C) = \frac{1}{2} \times 1 \times [0 + 1 \times V_\pi(B)] + \frac{1}{2} \times 1 \times [0 + 1 \times V_\pi(D)] = \frac{1}{2} V_\pi(B) + \frac{1}{2} V_\pi(D)$

$V_\pi(D) = \frac{1}{2} \times 1 \times [0 + 1 \times V_\pi(C)] + \frac{1}{2} \times 1 \times [0 + 1 \times V_\pi(E)] = \frac{1}{2} V_\pi(C) + \frac{1}{2} V_\pi(E)$

$V_\pi(E) = \frac{1}{2} \times 1 \times [0 + 1 \times V_\pi(D)] + \frac{1}{2} \times 1 \times [0 + 1 \times V_\pi(R)] = \frac{1}{2} V_\pi(D) + \frac{1}{2}$

$$\Rightarrow \begin{cases} V_\pi(A) = \frac{1}{6} \\ V_\pi(B) = \frac{2}{6} = \frac{1}{3} \\ V_\pi(C) = \frac{3}{6} = \frac{1}{2} \\ V_\pi(D) = \frac{4}{6} = \frac{2}{3} \\ V_\pi(E) = \frac{5}{6} \end{cases}$$

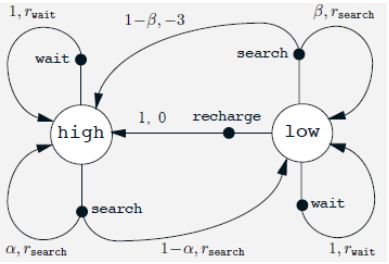
The consistency is verified.

(c) What do you think the value function is for arbitrary an arbitrary number of states n?

The value function is $v_\pi(S_i) = \frac{i-1}{n-1}$ $(i \geq 2)$ as n is the number of state (e.g. L, A, B, …, R) and i is position of the state.

8. **2 points.** *Solving for the value function.*
Written: Another way to solve for the value function is to write out the Bellman equation for each state, view it as a system of linear equations, and then solve for the unknowns (the value of each state). Consider a particularly simple MDP, the 2-state recycling robot in Example 3.3.

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



(a) Expand the Bellman equation for the 2 states in the recycling robot, for an arbitrary policy $\pi(a|s)$, discount factor γ, and domain parameters $\alpha, \beta, r_{\texttt{search}}, r_{\texttt{wait}}$ as described in the example.

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] && \text{(by (3.9))} \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r \mid s, a) \Big[r + \gamma \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s'] \Big] \\
&= \sum_a \pi(a|s) \sum_{s', r} p(s', r \mid s, a) \Big[r + \gamma v_\pi(s') \Big], \quad \text{for all } s \in \mathcal{S}, && (3.14)
\end{aligned}
$$

Follow by equation from 3.14.

$$V(\text{high}) = \pi(\text{search} \mid \text{high}) [\alpha (r_{\text{search}} + \gamma V(\text{high})) + (1-\alpha)(r_{\text{search}} + \gamma V(\text{low}))] + \pi(\text{wait} \mid \text{high}) [1 (r_{\text{wait}} + \gamma V(\text{high}))]$$

$$V(\text{low}) = \pi(\text{search} \mid \text{low}) [(1-\beta)(r_{\text{search}} + \gamma V(\text{high})) + \beta (r_{\text{search}} + \gamma V(\text{low}))] + \pi(\text{wait} \mid \text{low}) [1 (r_{\text{wait}} + \gamma V(\text{low}))] + \pi(\text{recharge} \mid \text{low}) [1 (0 + \gamma V(\text{high}))]$$

(b) You should now have two linear equations involving two unknowns, $v(\texttt{high})$ and $v(\texttt{low})$, as well as involving the policy $\pi(a|s)$, γ, and the domain parameters. Let $\alpha = 0.8, \beta = 0.6, \gamma = 0.9, r_{\texttt{search}} = 10, r_{\texttt{wait}} = 3$.
Consider the policy $\pi(\texttt{search} \mid \texttt{high}) = 1$, $\pi(\texttt{wait} \mid \texttt{low}) = 0.5$, and $\pi(\texttt{recharge} \mid \texttt{low}) = 0.5$.
Find the value function for this policy, i.e., solve the equations for the values of $v(\texttt{high})$ and $v(\texttt{low})$.
Check that your solution satisfies the Bellman equation.

$$V(\text{high}) = 1 [0.8 (10 + 0.9 V(\text{high})) + (1-0.8)(10 + 0.9 V(\text{low}))] + (1 - \pi(\text{search} \mid \text{high})) [1 (3 + 0.9 V(\text{high}))]$$

$$= 8 + 0.72 V(\text{high}) + 2 + 0.18 V(\text{low}) + 0$$

$$0.28 V(\text{high}) - 0.18 V(\text{low}) - 10 = 0 \quad \cdots (1)$$

$$V(\text{low}) = (1 - \pi(\text{wait} \mid \text{low}) - \pi(\text{recharge} \mid \text{low}))[(1-0.6)(10 + 0.9 V(\text{high})) + 0.6 (10 + 0.9 V(\text{low}))] + 0.5 [1 (3 + 0.9 V(\text{low})] + 0.5 [0.9 V(\text{high})]$$

$$= 1.5 + 0.45 V(\text{low}) + 0.45 V(\text{high})$$

$$0.45 V(\text{high}) - 0.55 V(\text{low}) + 1.5 = 0 \quad \cdots (2)$$

From equation (1) & (2): $\begin{cases} V(\text{high}) \approx 79 \\ V(\text{low}) \approx 67 \end{cases}$

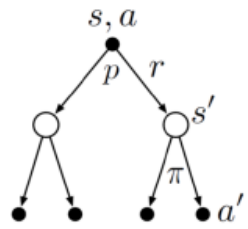
(c) (**Bonus, optional**) Suppose you can modify the policy in the `low` state, i.e., set $\pi(\texttt{wait} \mid \texttt{low}) = \theta$, and $\pi(\texttt{recharge} \mid \texttt{low}) = 1 - \theta$. What θ should you set it to, and what is the value function for that θ?

$$V(high) = 1[0.8(10 + 0.9V(high)) + (1 - 0.8)(10 + 0.9V(low))]$$
$$= 8 + 0.72V(high) + 2 + 0.18V(low)$$
$$V(low) = \theta[1(3 + 0.9V(low))] + (1 - \theta)[1(10 + 0.9V(high))]$$
$$= \theta[3 + 0.9V(low)] + (1 - \theta)[0.9V(high)]$$
$$= 3\theta + 0.9\theta V(low) + 0.9(1 - \theta)V(high)$$
$$0.28V(high) - 0.18V(low) - 10 = 0 \quad \cdots (1)$$
$$0.9(1 - \theta)V(high) + (0.9\theta - 1)V(low) + 3\theta = 0 \quad \cdots (2)$$

According to equation (1) and (2), we could get $v_{(low)} = \frac{9.0 - 8.2\theta}{0.11 - 0.09\theta}$.

In order to get the maximum value of $v_{(low)}$, $v_{(low)} \approx 76$, $\theta = 0$ and $v_{(high)} \approx 85$.

9. **1 point.** (RL2e 3.12, 3.13, 3.17) *Action-value function.*



q_π backup diagram

Written:

(a) Give an equation for v_π in terms of q_π and π.

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a)$$

(b) Give an equation for q_π in terms of v_π and the four-argument p.

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

(c) What is the Bellman equation for action values, that is, for q_π? It must give the action value $q_\pi(s,a)$ in terms of the action values, $q_\pi(s',a')$, of possible successors to the state–action pair (s,a).
Hint: The backup diagram above corresponds to this equation. Show the sequence of equations analogous to Equation 3.14, but for action values.

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