STAT 184, PROBLEM SET 2

Matthew Qu
Due: October 20, 2022
matthewqu@college.harvard.edu

0. (**Collaborators and Acknowledgements**)
Collaborators: Kevin Huang, Katherine Zhou
Acknowledgements:

1. (**Markov Decision Process Modelling**)

Proof.

1. Creating a self-steering drone that lands in a target area (i.e. on top of a cliff, on an island, etc.).

2. A typical drone has 4 propellers that can be set at different rotation speeds to allow the drone to fly in any direction and also to turn. We could parametrize each propeller in terms of revolutions per minute (rpm), with clockwise being positive and counter-clockwise being negative. For simplicity, our actions could be to slow down, stay constant, or speed up any combination of the 4 propellers by 1 rpm per unit of time, which would be a total of 81 actions, but a more complex system could make this continuous and not a constant change for each propeller.

3. Given that the goal is to land the drone in a target area, the observations needed would be the drone's position (with respect to the target), velocity, and acceleration at any point in time. The states would include this information along with the speed of each of the 4 propellers. Other possibly useful observations could include total flight time (as discussed in part 5).

4. The transition function would initially be determined by how changing the rotations of the propellers would affect the position, velocity, and acceleration of the drone. Without any external factors, this transition is deterministic, as we can calculate how each action affects these factors. However, we can assign a probability distribution to the subsequent state due to external factors such as wind or malfunction in the drone. These factors would affect the next state of the drone, so we can account for them by having some small probability of being "near" the calculated next state.

5. The largest reward would come from successfully landing the drone at the target. This means that if the observations for position, velocity, and acceleration are all 0, and the states of the propellers are also 0, then the landing reward would be received. To help guide the drone using rewards, we could reward actions that take the drone closer to the target, while also penalizing reckless flying, such as when velocity or acceleration are too large. In addition, we can use the total time in flight as a way to scale the rewards, as we ideally would like the time in flight to be minimized, so actions with a lower total time will have larger rewards.

□

2. (**Bellman Optimality**)

Proof.

1. We know that Q^* still satisfies

$$Q^*(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} V^*(s')$$

for all s and a. Therefore, it suffices to show that $V^*(s') = \max_{a'} Q^*(s',a')$ for all $s' \in S$. Let π^* be a deterministic policy such that $V^{\pi^*} = V^*$. Then, we have $V^*(s) = Q^*(s, \pi^*(s))$ for all s. But we know that $\pi^*(s) = \operatorname{argmax}_a Q^*(s,a)$. Therefore, we have

$$\begin{aligned} V^*(s) &= Q^*(s, \pi^*(s)) \\ &= Q^*(s, \operatorname*{argmax}_a Q^*(s,a)) \\ &= \max_a Q^*(s,a). \end{aligned}$$

This concludes the proof.

2. We will follow the same proof idea used in class. Let Q satisfy the given equality for all s and a, and let $\|Q - Q^*\|_\infty = \max_{s \in S, a \in A} |Q(s,a) - Q^*(s,a)|$. We will show that $\|Q - Q^*\|_\infty \leq \gamma \|Q - Q^*\|_\infty$, which implies that $\|Q - Q^*\|_\infty = 0$ and therefore $Q = Q^*$. We have, for any s and a,

$$\begin{aligned} |Q(s,a) - Q^*(s,a)| &= \left| \left(r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} \max_{a'} Q(s',a')\right) - \left(r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} \max_{a'} Q^*(s',a')\right) \right| \\ &= \gamma \left| \mathbb{E}_{s' \sim P(s,a)} \max_{a'} Q(s',a') - \mathbb{E}_{s' \sim P(s,a)} \max_{a'} Q^*(s',a')) \right| \\ &= \gamma \left| \mathbb{E}_{s' \sim P(s,a)} \left(\max_{a'} Q(s',a') - \max_{a'} Q^*(s',a') \right) \right| \\ &\leq \gamma \mathbb{E}_{s' \sim P(s,a)} \left| \max_{a'} Q(s',a') - \max_{a'} Q^*(s',a') \right|. \end{aligned}$$

Now, we use the fact that $|\max_x f(x) - \max_x g(x)| \leq \max_x |f(x) - g(x)|$ to conclude that

$$\left| \max_{a'} Q(s',a') - \max_{a'} Q^*(s',a') \right| \leq \max_{a'} \left| Q(s',a') - Q^*(s',a') \right|$$

for all $s' \in S$. Therefore, it follows that

$$\begin{aligned} \gamma \mathbb{E}_{s' \sim P(s,a)} \left| \max_{a'} Q(s',a') - \max_{a'} Q^*(s',a') \right| &\leq \gamma \mathbb{E}_{s' \sim P(s,a)} \max_{a'} \left| Q(s',a') - Q^*(s',a') \right| \\ &\leq \gamma \max_{s'} \max_{a'} \left| Q(s',a') - Q^*(s',a') \right| \\ &= \gamma \|Q - Q^*\|_\infty. \end{aligned}$$

Therefore, for any s and a, we have $|Q(s,a) - Q^*(s,a)| \leq \gamma \|Q - Q^*\|_\infty$, which means that

$$\|Q - Q^*\|_\infty \leq \gamma \|Q - Q^*\|_\infty.$$

This implies that $Q = Q^*$, as desired.

□

3. (**Gardening as MDP**)

Proof.

1. We will write the transition function as two matrices, depending on the action taken. If $a = 1$, i.e. we choose to do work in the garden, then we will be in state 0 with probability 1, regardless of the initial state. As a matrix, we can express this as

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ B \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & B \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

where the initial state s are the rows and the new state s' are the columns. When $a = 0$, then the transition matrix is

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ B \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & B \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} & \frac{2}{3} \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}.$$

Likewise, we can express the reward function as an $s \times a$ matrix R, where $r(s, a) = R_{sa}$. We have

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ B \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} 1 & 1-\alpha \\ 1 & 1-\alpha \\ 1 & 1-\alpha \\ 1 & 1-\alpha \\ 0 & -2\alpha \end{pmatrix} \end{array}.$$

2. Given the threshold policy π_C, we know that $\pi_C(B) = \pi_C(s) = 1$ for $s \geq C$. We also know that $P(s' \mid s, 1) = 1$ for $s' = 0$ and all $s \in S$. Therefore, we have

$$\begin{aligned} V^{\pi_C}(B) &= r(B, \pi_C(B)) + \gamma \mathbb{E}_{s' \sim P(B, \pi_C(B))} V^{\pi_C}(s') \\ &= -2\alpha + \gamma V^{\pi_C}(0). \end{aligned}$$

Similarly, we have for $s \geq C$,

$$\begin{aligned} V^{\pi_C}(s) &= r(s, \pi_C(s)) + \gamma \mathbb{E}_{s' \sim P(s, \pi_C(s))} V^{\pi_C}(s') \\ &= 1 - \alpha + \gamma V^{\pi_C}(0). \end{aligned}$$

Now, when $s < C$, we choose not to work in the garden. Then, there is probability $\frac{s}{3}$ that the garden becomes infested, and probability $\frac{3-s}{3}$ that the garden goes to state $s + 1$. Therefore, we have

$$V^{\pi_C}(s) = 1 + \gamma \left(\frac{3-s}{3} V^{\pi_C}(s+1) + \frac{s}{3} V^{\pi_C}(B) \right).$$

Plugging in our expression for $V^{\pi_C}(B)$ found above, we have

$$V^{\pi_C}(s) = 1 + \gamma\left(\frac{3-s}{3}V^{\pi_C}(s+1) + \frac{s}{3}(-2\alpha + \gamma V^{\pi_C}(0))\right).$$

3. We will calculate the values for $V^{\pi_C}(s)$ for all $s \in S$ and $C \in \{0,1,2,3\}$. Note that from part (2), every value of $V^{\pi_C}(s)$ will be the same for $s \geq C$, and $V^{\pi_C}(B) = V^{\pi_C}(s) - 1 - \alpha = V^{\pi_C}(s) - 1.5$ where $s \geq C$. For $C = 0$, we have $V^{\pi_0}(0) = 1 - 0.5 + \frac{2}{3}V^{\pi_0}(0)$, which yields $V^{\pi_0}(0) = 1.5$. Thus, the value function for $C = 0$ is $V_0 = [1.5, 1.5, 1.5, 1.5, 0]$, where the array is indexed by state $\{0,1,2,3,B\}$, respectively. For $C = 1$, we have the system of equations

$$\begin{aligned} V^{\pi_1}(0) &= 1 + \frac{2}{3}V^{\pi_1}(1) \\ V^{\pi_1}(1) &= 0.5 + \frac{2}{3}V^{\pi_1}(0), \end{aligned}$$

which yields $V^{\pi_1}(0) = 2.4$ and $V^{\pi_1}(1) = 2.1$. Thus, we have value function $V_1 = [2.4, 2.1, 2.1, 2.1, 0.6]$. For $C = 2$, we now have the equations

$$\begin{aligned} V^{\pi_2}(0) &= 1 + \frac{2}{3}V^{\pi_2}(1) \\ V^{\pi_2}(1) &= 1 + \frac{2}{3}\left(\frac{2}{3}V^{\pi_2}(2) + \frac{1}{3}\left(-1 + \frac{2}{3}V^{\pi_2}(0)\right)\right) \\ V^{\pi_2}(2) &= 0.5 + \frac{2}{3}V^{\pi_2}(0). \end{aligned}$$

These equations yield the solution $V^{\pi_2}(0) \approx 2.37, V^{\pi_2}(1) \approx 2.05$, and $V^{\pi_2}(2) \approx 2.08$. In particular, these value are lower than that of $C = 1$. Finally, for $C = 3$, we have the equations

$$\begin{aligned} V^{\pi_3}(0) &= 1 + \frac{2}{3}V^{\pi_3}(1) \\ V^{\pi_3}(1) &= 1 + \frac{2}{3}\left(\frac{2}{3}V^{\pi_3}(2) + \frac{1}{3}\left(-1 + \frac{2}{3}V^{\pi_3}(0)\right)\right) \\ V^{\pi_3}(2) &= 1 + \frac{2}{3}\left(\frac{1}{3}V^{\pi_3}(3) + \frac{2}{3}\left(-1 + \frac{2}{3}V^{\pi_3}(0)\right)\right) \\ V^{\pi_3}(3) &= 0.5 + \frac{2}{3}V^{\pi_3}(0), \end{aligned}$$

which yield the solution $V^{\pi_3}(0) \approx 2.23, V^{\pi_3}(1) \approx 1.84, V^{\pi_3}(2) \approx 1.66$, and $V^{\pi_3}(3) \approx 1.99$. Therefore, we find that the optimal value C_* is $C_* = 1$.

4. Given the value function $V_1 = [2.4, 2.1, 2.1, 2.1, 0.6]$ and π_1, we can determine the Q function. Since $V^{\pi_1}(s) = Q(s, \pi_1(s))$, we already know half the values of the Q function. We calculate the

remaining entries:

$$
\begin{aligned}
Q^{\pi_1}(0,1) &= 1 - 0.5 + \frac{2}{3}V^{\pi_1}(0) = 0.5 + 1.6 = 2.1 \\
Q^{\pi_1}(1,0) &= 1 + \frac{2}{3}\left(\frac{2}{3}V^{\pi_1}(2) + \frac{1}{3}V^{\pi_1}(B)\right) = 1 + \frac{2}{3}(1.4 + 0.2) = \frac{62}{30} \\
Q^{\pi_1}(2,0) &= 1 + \frac{2}{3}\left(\frac{1}{3}V^{\pi_1}(3) + \frac{2}{3}V^{\pi_1}(B)\right) = 1 + \frac{2}{3}(0.7 + 0.4) = \frac{52}{30} \\
Q^{\pi_1}(3,0) &= 1 + \frac{2}{3}V^{\pi_1}(B) = 1 + 0.4 = 1.4 \\
Q^{\pi_1}(B,0) &= 0 + \frac{2}{3}V^{\pi_1}(B) = 0.4.
\end{aligned}
$$

We can represent the Q function as a matrix:

$$
\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ B \end{array}
\begin{array}{c}
\begin{array}{cc} 0 & 1 \end{array} \\
\begin{pmatrix} 2.4 & 2.1 \\ 2.07 & 2.1 \\ 1.73 & 2.1 \\ 1.4 & 2.1 \\ 0.4 & 0.6 \end{pmatrix}
\end{array}.
$$

Note that the policy π_1 satisfies $\pi_1(s) = \operatorname{argmax}_a Q^{\pi_1}(s,a)$ for all s. Therefore, $V^{\pi_1}(s) = \max_a Q^{\pi_1}(s,a)$ for all s, which implies Bellman optimality; that is,

$$
Q^{\pi_1}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} \max_{a'} Q^{\pi_1}(s',a')
$$

for all s and a. This is because the Bellman equation, which replaces $\max_{a'} Q^{\pi_1}(s',a')$ with $V^{\pi_1}(s)$ in the equation above is used to calculate each value of $Q(s,a)$. Therefore, we conclude that π_1 is optimal.

□

2. (**Finding Exact Optimal Policy with Policy Iteration**)

Proof.

1. Suppose we have that

$$Q^{\pi^{t+1}}(s,a) = Q^{\pi^t}(s,a)$$

for all s and a. It follows that $\pi^{t+1}(s) = \underset{a}{\operatorname{argmax}}\, Q^{\pi^t}(s,a) = \underset{a}{\operatorname{argmax}}\, Q^{\pi^{t+1}}(s,a)$ for all s per the policy improvement step. Then, in the policy iteration step, we set

$$Q^{\pi^{t+1}}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} V^{\pi^{t+1}}(s').$$

Furthermore, we also note that

$$\begin{aligned} V^{\pi^{t+1}}(s') &= Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) \\ &= Q^{\pi^{t+1}}(s', \underset{a}{\operatorname{argmax}}\, Q^{\pi^{t+1}}(s,a)) \\ &= \max_{a'} Q^{\pi^{t+1}}(s', a'). \end{aligned}$$

Thus, we have that

$$Q^{\pi^{t+1}}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} \max_{a'} Q^{\pi^{t+1}}(s', a')$$

for all s and a. By Bellman optimality in Problem 2, we conclude that π^{t+1} is an optimal policy.

2. We know that there are exactly $|A|^{|S|}$ possible policies, i.e. $|A|$ possible actions for each of the $|S|$ states. By part (1), we know that if policy iteration does not change the policy, then it has found an optimal policy. Furthermore, in the case where policy iteration chooses a policy π, updates to a different policy π', and then at some point chooses policy π again, we must have

$$Q^{\pi}(s,a) \geq Q^{\pi'}(s,a) \geq Q^{\pi}(s,a)$$

for all s and a due to monotonic improvement. This implies that $Q^{\pi} = Q^{\pi'}$, and so by part (1) both policies are optimal. Therefore, if policy iteration choose a policy π twice, it must be optimal. Thus, in the worst-case scenario, policy iteration chooses a different policy at each iteration until all possible policies are exhausted—then, the next policy must be optimal. If we start with a random policy at the 0^{th} iteration, then by the pigeonhole principle, policy iteration must find an optimal policy within $|A|^{|S|}$ iterations.

□