Reinforcement learning group assignment 1

Enzo Keuning (12878502), Hsi Yun Chien (12534919), Jamie Mo (12475440)

March 2, 2022

NOTE: For the actions that cannot be chosen or do not make sense to be chosen we have assumed a reward of $-\infty$. By doing this they will not influence the state values in following iterations.

1 Question a

In this question, we have two actions to take for each day, the boat is either rented out (action = 0) or under maintenance (action = 1). There are four states (1: Excellent, 2: Good, 3: Average, 4: Bad) which represents the condition of the boat.

The cost of maintenance is \$40 for the boat which rates 2 or 3. \$100 for the boat in the bad (4) condition, including the reimbursement cost and the cost of maintenance. The rental income is \$60 per day. If the boat is in excellent state, it is logical to take the action 'rent out', which leaves 'under maintenance' a redundant. Same logic applies to the bad condition, only the option 'under maintenance' is needed, since it is impossible to rent out a broken boat. For the above reasons, the reward vector is set to [[60, $-\infty$], [60, -40], [60,-40], [$-\infty$, -100]]].

The initial state is set to day 9 with the initial value function [0, 0, 0, -100], this is due to the fact that the boat must be rented again before the next holiday period, and the boat is assumed to be repaired on or after day 9 if necessary. In order to find the optimal policy for the spring break, we take the greedy policy for each time step. This is an episodic task, the total iteration is set to 8 with gamma = 1 by default. It shows that the future rewards are equally important as the current one.

As aforementioned, the algorithm adapted in the code is based on Bellman optimality equation. We create a nested list r to store all the maximal value for state 1, 2, 3, 4 for each day. A value in vector k is updated after each iteration. The action-value function $q_\pi(s, a)$ is maximised, and the corresponding policy is printed out using np.argmax(q).

The optimal policy for day 8 is [0 0 0 1] with the output value [60, 50, 10, -100]. For the rest of the spring break, the optimal policy is [0 0 1 1], which implies that the optimal policy suggests to rent out a boat of excellent or good quality and to perform maintenance on boats of average and bad quality.

This result seems fairly reasonable. As the transition probabilities show, for boats in state 1 and 2, if action 'rent out' is chosen, it is less likely to drop to state 4, so it is more profitable to 'rent out'. Whereas, for the boat in state 3, the 'rent out' option leads to state 4 with the probability of 0.5 (staying at state 3 with the probability of 0.5), this results in performing maintenance in state 3 as the optimal choice. Note that state 4 can only take action 'under maintenance'.

If we round up the value vector, the state values for day 1 are: [389, 327, 304, 244]

2 Question b

In this question, the boat could have two actions, renting out and maintenance, and there are four states about the condition of the boat.(1: Excellent, 2: Good, 3: Average, 4: Bad).

$$\text{States} = 0, \cdots, 4 \quad \text{(Condition of a machine)}$$
$$\text{Actions} = 0, 1$$

The actions are specified in the following way:

$$0 = \text{rent out}$$
$$1 = \text{maintenance}$$

$$\begin{aligned} p = &[[0.8, 0.1, 0.1, 0], [0, 0, 0, 0]], \\ &[[0, 0.6, 0.3, 0.1], [1, 0, 0, 0]], \\ &[[0, 0, 0.5, 0.5], [1, 0, 0, 0]], \\ &[[0, 0, 0, 0], [1, 0, 0, 0]]] \end{aligned}$$

The transition probability and the rewards vector are the same as in part (a).

$$r = [[60, -\infty], [60, -40], [60, -40], [-\infty, -100]]].$$

The reason we put the value $-\infty$ in state 1 is because it is not beneficial or doesn't make sense to do maintenance for state 1, so we connected this action to an extremely unattractive reward. When the boat is in excellent condition, there will be a reward of 60 for renting out. Therefore in state 1 the reward for renting out is \$60, while we connected maintenance to a reward of $-\infty$
Since the rental income is \$60 for per day, the rewards for rent out (action = 0), is \$60 for state 1 till state 3 . Maintenance for state 2 and 3 infers a cost of \$40. Therefore the rewards vector of state 2 and 3 are [60, -40]. Renting out of a boat in bad condition is not possible (state = 4), hence we connected a reward of $-\infty$ to this option as it is redundant.
Since the boat is broken in state 4, by summing up the reimbursement cost and cost of maintenance, rewards for maintenance should be -100.
The main difference was that in this part (b), we assume a continuing problem without terminal states, while part (a) is an episodic problem with terminal state. Therefore in part (b) we use gamma = 0.9, while in part (a) gamma was 1. This is done, because in a continuing task we need to discount the rewards, as otherwise the return would go to ∞. This does not happen in an episodic task, so there is no need for discounting in part (a).
To evaluate the optimal policy for each states, we calculated the state values and chose the one which had the highest total discounted reward(profit). This is consistent with the method in question (a).
The resulting optimal policy is consistent with part (a), while the state-values have a similar structure, but they are moderately higher:

Optimal policy : [0, 0, 1, 1]
Estimated v-values : [473.90182244, 421.1830066, 386.5116402, 326.5116402]

From the result of our coding, we can see that the optimal policy in state 1 and 2 is action = 0, which implies that for a boat in an excellent or good condition it is optimal to rent it out. In state 3 and 4, so for boats in an average or good condition, the optimal policy shows that it is most favorable to do maintenance.

Like in part (a) this result seems sensible because when the boat is in an excellent or good state, it is better to rent out, since their rewards are high and there is only a low probability to drop drop to an average or bad condition during the day. However, the probability of a boat of average condition dropping to a bad condition is 0.5, hence it seems to be more profitable to already do maintenance rather than renting out a boat in an average condition. Furthermore a boat in a bad condition has no choice but to be repaired.

3 Question c

For this questions we specify the following:

$$\begin{aligned}&\text{States} = 0, \cdots, 10 \quad \text{(Condition of a machine)}\\&\text{Actions} = 0, \cdots, 5\end{aligned}$$

The actions are specified in the following way:

$$\begin{aligned}&0 = \text{buy a new one}\\&1 = \text{buy an almost new one}\\&2 = \text{buy a second-hand one}\\&3 = \text{minor repairs}\\&4 = \text{major repairs}\\&5 = \text{use the machine}\end{aligned}$$

For the actions that cannot be chosen or do not make sense to be chosen we have assumed a reward of $-\infty$. By doing this these actions will never be selected and hence will not influence the state values in following iterations. Also, all transition probabilities for these actions are set equal to zero.
Initially, we have defined the following values:

$$\begin{aligned}&c_{new} = 80,\ c_{an} = 70,\ c_{sh} = 60,\ c_{minor} = 30,\ c_{major} = 50\\&r = (100, 90, 80, 70, 60, 50, 40, 30, 20, 10)\end{aligned}$$

The r-vector is the reward scheme for 'using' a machine in condition= $0, \cdots, 9$. Where the first value, 100, corresponds to the reward for using a machine in condition= 0, up to the last value, 10, which corresponds to the reward for using a machine in condition= 9.
In Table 1 and Table 2 the transition probabilities and the reward scheme is given. The code for the calculations is analogous to the code in part (b), here we also use gamma = 0.9.
With this reward scheme we find the optimal policy:

$$\pi = [5,\ 5,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]$$

Which implies that according to the optimal policy we need to buy a new machine when the rating of the current machine is 2 or higher; otherwise we do nothing and keep using it.

4 Question d

Here, we focus on two different cases:
i) Explore the influence of different reward schemes on the optimal policy.
ii) Explore the influence of different replacement cost schemes on the optimal policy.

Referring to the python code for question (d) we can sum up a few findings:
i)

Table 1: Table with transition probabilties, $P(s'|s,a)$, and rewards, $r(s,a)$

states	actions	0	1	2	3	4	5	6	7	8	9	10	r(s,a)	q(s,a)
0	0	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	1	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	2	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	3	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	4	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	5	0.8	0.1	0.05	0.05	0	0	0	0	0	0	0	100	
1	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	0	$100 - c_{new}$	
	1	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	2	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	3	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	4	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	5	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	90	
2	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	0	$100 - c_{new}$	
	1	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	$90 - c_{an}$	
	2	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	3	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	4	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	5	0	0	0.8	0.1	0.05	0.05	0	0	0	0	0	80	
3	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	0	$100 - c_{new}$	
	1	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	$90 - c_{an}$	
	2	0	0	0.8	0.1	0.05	0.05	0	0	0	0	0	$80 - c_{sh}$	
	3	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	4	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	5	0	0	0	0.8	0.1	0.05	0.05	0	0	0	0	70	
4	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	0	$100 - c_{new}$	
	1	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	$90 - c_{an}$	
	2	0	0	0.8	0.1	0.05	0.05	0	0	0	0	0	$80 - c_{sh}$	
	3	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	4	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	5	0	0	0	0	0.8	0.1	0.05	0.05	0	0	0	60	
5	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	0	$100 - c_{new}$	
	1	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	$90 - c_{an}$	
	2	0	0	0.8	0.1	0.05	0.05	0	0	0	0	0	$80 - c_{sh}$	
	3	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	4	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	5	0	0	0	0	0	0.8	0.1	0.05	0.05	0	0	50	
6	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	0	$100 - c_{new}$	
	1	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	$90 - c_{an}$	
	2	0	0	0.8	0.1	0.05	0.05	0	0	0	0	0	$80 - c_{sh}$	
	3	0	0	0	1	0	0	0	0	0	0	0	$-c_{minor}$	
	4	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	5	0	0	0	0	0	0	0.8	0.1	0.05	0.05	0	40	
7	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	0	$100 - c_{new}$	
	1	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	$90 - c_{an}$	
	2	0	0	0.8	0.1	0.05	0.05	0	0	0	0	0	$80 - c_{sh}$	
	3	0	0	0	0	1	0	0	0	0	0	0	$-c_{minor}$	
	4	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	5	0	0	0	0	0	0	0	0.8	0.1	0.05	0.05	30	

Table 2: Table with transition probabilties, $P(s'|s,a)$, and rewards, $r(s,a)$ (states 8, 9, and 10)

states	actions	0	1	2	3	4	5	6	7	8	9	10	r(s,a)	q(s,a)
8	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	0	$100 - c_{new}$	
	1	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	$90 - c_{an}$	
	2	0	0	0.8	0.1	0.05	0.05	0	0	0	0	0	$80 - c_{sh}$	
	3	0	0	0	0	0	1	0	0	0	0	0	$-c_{minor}$	
	4	0	0	0	1	0	0	0	0	0	0	0	$-c_{major}$	
	5	0	0	0	0	0	0	0	0	0.8	0.15	0.05	20	
9	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	0	$100 - c_{new}$	
	1	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	$90 - c_{an}$	
	2	0	0	0.8	0.1	0.05	0.05	0	0	0	0	0	$80 - c_{sh}$	
	3	0	0	0	0	0	0	1	0	0	0	0	$-c_{minor}$	
	4	0	0	0	0	1	0	0	0	0	0	0	$-c_{major}$	
	5	0	0	0	0	0	0	0	0	0	0.8	0.2	10	
10	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	0	$100 - c_{new}$	
	1	0	0.8	0.1	0.05	0.05	0	0	0	0	0	0	$90 - c_{an}$	
	2	0	0	0.8	0.1	0.05	0.05	0	0	0	0	0	$80 - c_{sh}$	
	3	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	4	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	
	5	0	0	0	0	0	0	0	0	0	0	0	$-\infty$	

- When dividing all rewards in every condition(=state) by 2, we see that the optimal policy says that when a machine is in condition 0 up until 3 we should keep using it, and otherwise replace it with a new one. This is different compared to the initial rewards, where the optimal policy showed that we should only keep using a machine in condition 0 and 1.
- When multiplying all rewards by 4, we see that the optimal policy suggests to only keep using a machine of condition 0 and replace machines with another condition. Hence, when multiplying all rewards by a significantly high factor we see that we should start replacing the machine in a better condition (lower state), whereas when dividing all rewards by a significantly high factor we see that we should still keep using a machine in a worse condition according, to the optimal policy.
- When we have a reward scheme with (nearly) constant rewards, in this case slightly decreasing from 100 in state 0 to 90 in state 9. We see that the optimal policy suggests to keep using a machine in all conditions, except in condition 10 in which it should be replaced by a second-hand one.
- When we have the first $i = 10, .., 0$ rewards in the r-vector equal to 100, and the last $j = 10-i$ rewards in the r-vector equal to 0, we see that in general (most of the times) according to the optimal policy we want to keep using a machine in condition$= 0, \cdots, i$, and replace a machine in condition$= 9, \cdots, 9-i$. The replacement when $i = 0, \cdots, 3$ happens with a new one, when $i = 4$ with an almost new one, and when $i \geq 5$ with a second-hand one.

ii)

- When doubling all replacement costs, the optimal policy shows that we should keep using the machine in condition 0 until condition 3; otherwise buy a new machine. Hence, when comparing this to the initial optimal policy in part (c) with lower replacement costs, we see that we should keep using the machine in worse conditions when the replacement costs increase.
- When multiplying all replacement costs by 5 the optimal policy shows that we should keep using the machine in even worse conditions, from condition 0 to condition 5. However, a

machine in condition 6 to 9 should only be undergoing repairs in this situation. A machine in condition 10, unrepairable, should still be replaced with a new one.

- When decreasing the replacement costs by a substantially high number (in the code we used division by 3), we see that we should replace the machine with a new one in all conditions except in condition 0, as this already is a new one. Hence, when decreasing the replacement costs we see that, in general, replacement of a machine will be done earlier (in better conditions).