Q1: A Variant of Double Q-Learning

The psuedocode of modified SARSA is given below. A few things to note:

1. In line 8 of the psuedocode,
$$a_{\star}^{'} = \underset{a^{'} \in \mathcal{A}(x^{'})}{\arg\max}\, q_i\left(x^{'}, a^{'}\right)$$
is the **selection step**.

2. In line 8 of the psuedocode,
$$q_{next} = \frac{1}{2}\sum_{j \in U_i} q_j\left(x^{'}, a_{\star}^{'}\right)$$
is the **evaluation step**. In this step, we are finding the average of the Q-functions that we are not updating (hence divide by 2). An alternate approach to do this step is,
$$q_{next} = q_j\left(x^{'}, a_{\star}^{'}\right)$$
where j is choosen uniformly at random from the set U_i.

Algorithm 1: Psuedocode of Triple Q-Learning

1 For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize $q_1(x,a)$, $q_2(x,a)$, and $q_3(x,a)$ to any real value.
2 **for** *every episode* **do**
3 Reset the environment to get the current state x.
4 **while** *episode did not end* **do**
5 Choose action, a, for current state, x, **using any policy** (a common choice is an ε-greedy policy derived using the sum of current Q-functions, $q_1(x,\cdot)$, $q_2(x,\cdot)$, and $q_3(x,\cdot)$).
6 Take action, a. Environment will return reward, r, and go to next state, $x^{'}$.
7 Choose and index i uniformly at random from the set $\{1,2,3\}$. i is the index of the Q-function which we are going to update. Let, $U_i = \{j \in \{1,2,3\} : j \neq i\}$.
8 Update Q-function $q_i(\cdot,\cdot)$ only for the current state-action pair (x,a) using the following equations,
$$a_{\star}^{'} = \underset{a^{'} \in \mathcal{A}(x^{'})}{\arg\max}\, q_i\left(x^{'}, a^{'}\right)$$
$$q_{next} = \frac{1}{2}\sum_{j \in U_i} q_j\left(x^{'}, a_{\star}^{'}\right)$$
$$q_i(x,a) = q_i(x,a) + \alpha\left(r + \beta q_{next} - q_i(x,a)\right)$$
9 Set $x = x^{'}$ for the next time slots.

Q2: Bandit Setup

(a) In policy gradient, the last layer of the neural network is a softmax function. The output of the neural network is basically the output of the softmax function which are "probability-like" value. These probability like value are used to select the actions. More specifically, let p_i be the i^{th} output of the softmax function. Then action a_i is selected with probability p_i. If p_i is strictly greater than zero then action a_i will have a finite probability of getting chosen.

For p_i to be zero, the weights and biases of the softmax function has to be $\pm\infty$ which is not possible if the initial weights of the entire neural network is finite and the gradients during gradient-ascent are also finite. Now obviously, initial weights of the entire neural network are finite. Also, gradient of the commonly used loss functions and activation functions are finite. To conclude $p_i = 0$ is not possible and hence $p_i > 0$.

(b) The key observation is that since the average reward does not depend on the action, we don't need have to consider the action for time slot $t = 1$ because it will not effect the return and hence the value function in any way. However, we

still have to consider the action at time slot $t = 0$ because this action decides the state in time slot $t = 1$ which in turn decides the reward. So again, only action a_0 (action at time $t = 0$) matters. Hence, the following answers.

The context is x_0 (the state at time $t = 0$) and correspondingly the context space is S (the state space itself).

The action is a_0 and correspondingly the action space is A (same as action space of the RL setup).

Let r_0 and r_1 be the reward at time slot 0 and 1. So the average reward for context $x_0 = x$ and action $a_0 = a$ is

$$\begin{aligned} r(x,a) &= E[r_0 + \beta r_1 | x_0 = x, a_0 = a] \\ &= E[r_0 | x_0 = x, a_0 = a] + \beta E[r_1 | x_0 = x, a_0 = a] \\ &= r(x) + \beta \sum_{x' \in S} E\left[r_1 | x_1 = x', x_0 = x, a_0 = a\right] P\left[x_1 = x' | x_0 = x, a_0 = a\right] \quad (1) \\ &= r(x) + \beta \sum_{x' \in S} E\left[r_1 | x_1 = x'\right] P\left[x' | x, a\right] \quad (2) \\ &= r(x) + \beta \sum_{x' \in S} r\left(x'\right) P\left[x' | x, a\right] \end{aligned}$$

Equations (1) and (2) are obtained using law of total conditional expectation and Markovian property respectively.

Q3: MDP to Speed-Up Airport Immigration Process

(a) The states are the queue lengths of the three queues and two counters, one for each of the two immigration officers. So the state is $(l_0, l_1, l_2, c_1, c_2)$ where l_0, l_1, and l_2 are the queue lengths of the main queue, the queue of officer 1, and the queue of officer 2 respectively. And, c_1 and c_2 are the counter states corresponding to officers 1 and 2 respectively.

We have $l_i \in \{0, 1, 2, \ldots\}$, $i = 1, 2, 3$ and $c_i \in \{0, 1, \ldots, \tau_i - 1\}$, $i = 1, 2$. Hence, the state space is

$$S_l \times S_l \times S_l \times S_{c_1} \times S_{c_2}$$

where $S_l = \{0, 1, 2, \ldots\}$ and $S_{c_i} = \{0, 1, \ldots, \tau_i - 1\}$. Since, S_l is an infinite set, the state space is infinite. But still, we can write Bellman optimality equation for it (just that we can't solve it using value/policy iteration).

(b) The action (a_1, a_2) where $a_i = 1$ ($a_i = 0$) if a passenger is sent (not sent) from the main queue to queue of immigration office i. The action space is,

$$\mathcal{A}(l_0, l_1, l_2, c_1, c_2) = \begin{cases} \{(0,0)\} & , l_0 = 0 \\ \{(0,0), (1,0), (0,1)\} & , l_0 > 0 \end{cases}$$

(c) The average reward corresponding to a state-action (x, a) pair is

$$r\left(\underbrace{l_0, l_1, l_2, c_1, c_2}_{x}, a\right) = -\sum_{i=0}^{2} l_i$$

(d) The Bellman optimality equation is,

$$V^\star(l_0, l_1, l_2, c_1, c_2) = \max_{(a_1, a_2) \in \mathcal{A}(l_0, l_1, l_2, c_1, c_2)} q^\star(l_0, l_1, l_2, c_1, c_2, a_1, a_2)$$

where,

$$q^\star(l_0, l_1, l_2, c_1, c_2, a_1, a_2) = -\sum_{i=0}^{2} l_i + \beta \sum_{m=0}^{M} p_m \widetilde{V}(l_0 - a_1 - a_2 + m, l_1 + a_1, l_2 + a_2, c_1, c_2)$$

Qualitatively, $\widetilde{V}$ is a function that captures the expected return starting from the state that follows immediately after main queue's decision and passenger arrival in the main queues. And $\widetilde{V}$ if given by the following equation,

$$\widetilde{V}\left(\widetilde{l}_0,\widetilde{l}_1,\widetilde{l}_2,c_1,c_2\right) = \begin{cases} V^\star\left(\widetilde{l}_0,\widetilde{l}_1,\widetilde{l}_2,c_1-1,c_2-1\right) & ,c_1>0,c_2>0 \\ \theta_1 V^\star\left(\widetilde{l}_0,\left(\widetilde{l}_1-1\right)^+,\widetilde{l}_2,\tau_1-1,c_2-1\right)+\left(1-\theta_1\right)V^\star\left(\widetilde{l}_0,\widetilde{l}_1,\widetilde{l}_2,0,c_2-1\right) & ,c_1=0,c_2>0 \\ \theta_2 V^\star\left(\widetilde{l}_0,\widetilde{l}_1,\left(\widetilde{l}_2-1\right)^+,c_1-1,\tau_2-1\right)+\left(1-\theta_2\right)V^\star\left(\widetilde{l}_0,\widetilde{l}_1,\widetilde{l}_2,c_1-1,0\right) & ,c_1>0,c_2=0 \\ \phi_1 V^\star\left(\widetilde{l}_0,\left(\widetilde{l}_1-1\right)^+,\left(\widetilde{l}_2-1\right)^+,\tau_1-1,\tau_2-1\right)+\phi_2 V^\star\left(\widetilde{l}_0,\left(\widetilde{l}_1-1\right)^+,\widetilde{l}_2,\tau_1-1,0\right) & ,c_1=0,c_2=0 \\ +\phi_3 V^\star\left(\widetilde{l}_0,\widetilde{l}_1,\left(\widetilde{l}_2-1\right)^+,0,\tau_2-1\right)+\phi_4 V^\star\left(\widetilde{l}_0,\widetilde{l}_1,\widetilde{l}_2,0,0\right) & \end{cases}$$

where $x^+=\max\left(0,x\right)$ and $\phi_1=\theta_1\theta_2$, $\phi_2=\theta_1\left(1-\theta_2\right)$, $\phi_3=\left(1-\theta_1\right)\theta_2$, $\phi_4=\left(1-\theta_1\right)\left(1-\theta_2\right)$.