Artificial Intelligence
Exercises week 4 - RL - **Solutions**

COMP3411/9814

Question 1: Value functions

Consider a world with two states $S = \{S_1, S_2\}$ and two actions $A = \{a_1, a_2\}$, where the transitions δ and reward r for each state and action are as follows:

$$\begin{array}{ll} \delta(S_1, a_1) = S_1 & r(S_1, a_1) = 0 \\ \delta(S_1, a_2) = S_2 & r(S_1, a_2) = -1 \\ \delta(S_2, a_1) = S_2 & r(S_2, a_1) = +1 \\ \delta(S_2, a_2) = S_1 & r(S_2, a_2) = +5 \end{array}$$

i. Draw a picture of this world, using circles for the states and arrows forthe transitions.

Answer: See Figure 1 below.

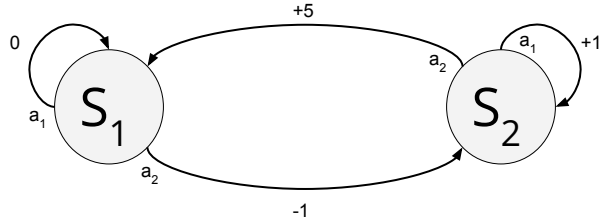

Figure 1: Draw of the world transitions.

ii. Assuming a discount factor of $\gamma = 0.9$, determine:

(a) the optimal policy $\pi^* : S \to A$

Answer: The optimal policy is:

$\pi^*(S_1) = a_2$
$\pi^*(S_2) = a_2$

(b) the state-value function $V^* : S \to R$

Remember $Q(s, a) = r(s, a) + \gamma V^*(s')$, if we follow the optimal policy then previous equation is also equal to the optimal state-value V^*

Answer: The optimal state-value function V^* is calculated as follows:

$V^*(S_1) = -1 + \gamma V^*(S_2)$
$V^*(S_2) = 5 + \gamma V^*(S_1)$
So $V^*(S_1) = -1 + 5\gamma + \gamma^2 V^*(S_1)$
$V^*(S_1) - \gamma^2 V^*(S_1) = -1 + 5\gamma$
$(1 - \gamma^2)V^*(S_1) = -1 + 5\gamma$
i.e. $V^*(S_1) = (-1 + 5\gamma)/(1 - \gamma^2) = 3.5/0.19 = 18.42$
And $V^*(S_2) = 5 + \gamma V^*(S_1)$
i.e. $V^*(S_2) = 5 + 0.9 * 3.5/0.19 = 21.58$

(c) the action-value function $Q^* : S \times A \to R$

Answer: As in the previous question $Q(s, a) = r(s, a) + \gamma V^*(s')$. So we only need to complete it for the other state-action pairs. The action-value function for the optimal policy is calculated as follows:

$Q(S_1, a_1) = 0 + \gamma V^*(S_1) = 16.58$
$Q(S_1, a_2) = V^*(S_1) = 18.42$
$Q(S_2, a_1) = 1 + \gamma V^*(S_2) = 20.42$
$Q(S_2, a_2) = V^*(S_2) = 21.58$

iii. Write the Q-values in a table (a.k.a. Q-table) as follows:

Q	a_1	a_2
S_1	16.58	18.42
S_2	20.42	21.58

iv. Trace through the first few steps of the action-value function learning algorithm, with all Q-values initially set to zero. Explain why it is necessary to force exploration through probabilistic choice of actions in order to ensure convergence to the true Q-values.

Answer: For the first three action selections

current state	chosen action	new Q-value
S_1	a_1	$0 + \gamma * 0 = 0$
S_1	a_2	$-1 + \gamma * 0 = -1$
S_2	a_1	$1 + \gamma * 0 = 1$

At this point, the Q-table looks like this:

Q	a_1	a_2
S_1	0	-1
S_2	1	0

If the agent always chooses the current best action, it can have a policy where it always prefers a suboptimal action, e.g., a_1 in state S_2, so will never sufficiently explore action a_2. This means that $Q(S_2, a_2)$ will remain zero forever, instead of converging to the true value of 21.58. With exploration, the next few steps might look like this:

current state	chosen action	new Q-value
S_2	a_2	$5 + \gamma * 0 = 5$
S_1	a_1	$0 + \gamma * 0 = 0$
S_1	a_2	$-1 + \gamma * 5 = 3.5$
S_2	a_1	$1 + \gamma * 5 = 5.5$
S_2	a_2	$5 + \gamma * 3.5 = 8.15$

Now the Q-table looks like this:

Q	a_1	a_2
S_1	0	3.5
S_2	5.5	8.15

From this point on, the agent will prefer action a_2 both in state S_1 and in state S_2. Further steps refine the Q-value estimates, and, in the limit, they will converge to their true values.

current state	chosen action	new Q-value
S_1	a_1	$0 + \gamma * 3.5 = 3.15$
S_1	a_2	$-1 + \gamma * 8.15 = 6.335$
S_2	a_1	$1 + \gamma * 8.15 = 8.335$
S_2	a_2	$5 + \gamma * 6.34 = 10.70$
...	...	...

Question 2: Temporal-difference learning

Consider the same world as the previous question. Assume the use of temporal-difference learning with the following parameters: learning rate $\alpha = 0.3$, discount factor $\gamma = 0.9$, and ϵ-greedy action selection method with $\epsilon = 0.1$. After a few steps of iterating, the learning agent performs action a_1 from state S_1 with the Q-table containing the following values:

Q	a_1	a_2
S_1	0.15	3.55
S_2	5.72	9.18

i. How would look the Q-table after one iteration of the off-policy method Q-learning?

Answer: We need to update $Q(S_1, a_1)$. As Q-learning update is:
$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a \in A(s_{t+1})} Q(s_{t+1}, a) - Q(s_t, a_t)]$
Then $Q(S_1, a_1) = 0.15 + \alpha(0 + \gamma Q(S_1, a_2) - 0.15$
$Q(S_1, a_1) = 0.15 + 0.3 * (0.9 * 3.55 - 0.15) = 1.0635$

The updated table looks like this:

Q	a_1	a_2
S_1	1.0635	3.55
S_2	5.72	9.18

ii. How would look the Q-table after one iteration of the on-policy method Sarsa? Assume a random value $rnd = 0.01$.

Answer: We also need to update $Q(S_1, a_1)$, however, as Sarsa update is as $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$, the update will depend on the random selected action by ϵ-greedy. Therefore, as the random number $rnd < \epsilon$, there are two possibilities.

First option, action a_1 is randomly selected:
$Q(S_1, a_1) = 0.15 + \alpha(0 + \gamma Q(S_1, a_1) - 0.15$
$Q(S_1, a_1) = 0.15 + 0.3 * (0.9 * 0.15 - 0.15) = 0.1455$

The updated table looks like this:

Q	a_1	a_2
S_1	0.1455	3.55
S_2	5.72	9.18

Second option, if action a_2 is randomly selected, the update is just the same as Q-learning update.
$Q(S_1, a_1) = 0.15 + \alpha(0 + \gamma Q(S_1, a_2) - 0.15$
$Q(S_1, a_1) = 0.15 + 0.3 * (0.9 * 3.55 - 0.15) = 1.0635$

The updated table looks like this:

Q	a_1	a_2
S_1	1.0635	3.55
S_2	5.72	9.18

iii. Explain how differently would the on-policy Sarsa method converge to the optimal value function in comparison to off-policy Q-learning.

Answer: This highly depends on the scenario the learning agent is interacting with. For instance, in the cliff walking scenario shown in Figure 2, the Q-learning agent learns the optimal path towards the goal while the Sarsa agent learns the safest, longest path. As the Sarsa agent avoids getting into the cliff the average collected reward over episodes is higher that the one collected by the Q-learning agent.

Particularly in the world shown in this tutorial, although both temporal-difference methods will converge to the optimal policy, Sarsa will need more episodes for that with probability $(1 - \epsilon)$, given that the action-selection method used in the next state s_{t+1} is ϵ-greedy with $\epsilon = 0.1$.

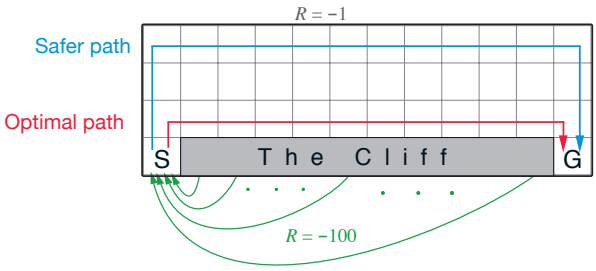

Figure 2: The cliff walking scenario.

Question 3: Returns

Consider a robot learning a task with a discount factor $\gamma = 0.5$ and receiving the following reward sequence: $R_1 = -1$, $R_2 = 2$, $R_3 = 6$, $R_4 = 3$, and $R_5 = 2$, and then 0 all the time. What are $G_0, G1, ..., G_5$? Hint: Work backwards.

Answer: We need to consider the discount return equation:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{1}$$

Then,
$G_5 = 0$ (and all the following ones)
$G_4 = 2 + 0 = 2$
$G_3 = 3 + 2 \times 0.5 = 4$
$G_2 = 6 + 3 \times 0.5 + 2 \times 0.5^2 = 6 + 1.5 + 0.5 = 8$
$G_1 = 2 + 6 \times 0.5 + 3 \times 0.5^2 + 2 \times 0.5^3 = 2 + 3 + 0.75 + 0.25 = 6$
$G_0 = -1+2\times0.5+6\times0.5^2+3\times0.5^3+2\times0.5^4 = -1+1+1.5+0.375+0.125 = 2$

Alternatively, as we worked backwards, we can also compute each return as $G_t = R_{t+1} + \gamma G_{t+1}$:
$G_5 = 0$
$G_4 = 2$
$G_3 = 3 + 0.5 \times G_4 = 3 + 1 = 4$
$G_2 = 6 + 0.5 \times G_3 = 6 + 2 = 8$

$G_1 = 2 + 0.5 \times G_2 = 2 + 4 = 6$
$G_0 = -1 + 0.5 \times G_1 = -1 + 3 = 2$