Reinforcement Learning (EL2805) Homework 2

EL2320, Autumn 2023

Oscar Eriksson	Philip Ahrendt
oscer@kth.se	pcah@kth.se
001130-1991	960605-R119

December 2023

Part 1: Q-learning and SARSA

Consider a discounted MDP with $S = \{A, B, C\}$ and $A = \{a, b, c\}$. We plan to use either the Q-learning or the SARSA algorithm in order to learn to control the system. We initialize the estimated Q-function as all zeros – that is:

$$Q^{(0)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The observed trajectory is as follows (for these transitions, we are imposed a policy):

$$(?, ?, ?); (A, ?, ?, ?); (B, a, 100); (A, b, 60); (B, c, 70); (C, b, 40); (A, a, 20); (C, c, ...)$$

a) Before the information became corrupt, we ran the Q-learning algorithm and obtained that

$$Q^{(2)} = \begin{bmatrix} 11 & 0 & 0 \\ 0 & 0 & 60 \\ 0 & 0 & 0 \end{bmatrix}$$

The discount factor was $\lambda = 0.5$ and the learning rate was fixed to $\alpha = 0.1$. Can you infer what the corrupt information was (i.e., the first state, the first and second selected actions, and the first and second observed rewards)? **Answer**: Given that the Q-learning update is written

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha\Big(R(s, a) + \lambda \max_{a' \in \mathcal{A}} Q_t(s', a') - Q_t(s, a)\Big)$$

we note that for $Q_t(s, a)$, $t > 0$ to be populated with anything other than a zero, we must have taken the action a at state s at some point.
At time step $t = 1$ we are in state A. We may then state that $[Q_2(A, a) \neq 0, s_1 = A] \iff [s_1 = A, a_1 = a]$.
Furthermore $[Q_2(A, a) \neq 0, Q_1(B, c) \neq 0, Q_0(\cdot, \cdot) = 0, s_1 = A] \iff [s_0 = B, a_0 = c]$.
Since (A, a) was only ever taken at $t = 1$ by $t = 2$, we may state that $Q_2(A, a) = 11$ while $Q_1(A, a) = 0$.
As such

$$\begin{aligned} & Q_2(A, a) = Q_1(A, a) + \alpha\Big(R(A, a) + \lambda \max_{a' \in \mathcal{A}} Q_1(B, a') - Q_1(A, a)\Big) \\ \iff & Q_2(A, a) = Q_1(A, a) + \alpha\Big(R(A, a) + \lambda Q_1(B, c) - Q_1(A, a)\Big) \\ \iff & R(A, a) = \alpha^{-1}(Q_2(A, a) - Q_1(A, a)) - \lambda Q_1(B, c) \end{aligned}$$

Given that $Q_2(A,a) = 11$, $Q_1(B,c) = 60$, $Q_1(A,a) = 0$ and $\alpha = 0.1$, $\lambda = 0.5$ we find that for $s_1 = A, a_1 = a$ that $R(A,a) = R_1 = 80$.
Again we state that $Q_1(B,c) = 60$ while $Q_0(B,c) = 0$. This implies that

$$\begin{aligned} & Q_1(B,c) = Q_0(B,c) + \alpha\Big(R(B,c) + \lambda \max_{a' \in \mathcal{A}} Q_0(A,a') - Q_0(B,c)\Big) \\ \iff & Q_1(B,c) = Q_0(B,c) + \alpha\Big(R(B,c) + 0 - Q_0(B,c)\Big) \\ \iff & R(A,a) = \alpha^{-1}(Q_1(B,c) - Q_0(B,c)) \end{aligned}$$

Given that $Q_1(B,c) = 60$, $Q_0(B,c) = 0$, $\lambda \max_{a' \in \mathcal{A}} Q_0(A,a') = 0$ and $\alpha = 0.1$, $\lambda = 0.5$ we find that for $s_0 = B, a_0 = c$ that $R(B,c) = R_0 = 600$.
We may now state that the full trajectory may be recovered, and it is

$$(B,c,600); (A,a,110); (B,a,100); (A,b,60); (B,c,70); (C,b,40); (A,a,20); (C,c,...) \quad \square$$

b) Provide the updated Q-values, using the Q-learning algorithm, at the 7th iteration. Use the same values for λ and α as in a). **Answer**: We list every update below

$$Q_1(s_0,a_0) = Q_0(s_0,a_0) + \alpha(R_0 + \lambda \max Q_0(s_1,a') - Q_0(s_0,a_0)) = 0.0 + 0.1(600.0 + 0.5\max(0.0,0.0,0.0) - 0.0) = 60.00$$
$$Q_2(s_1,a_1) = Q_1(s_1,a_1) + \alpha(R_1 + \lambda \max Q_1(s_2,a') - Q_1(s_1,a_1)) = 0.0 + 0.1(80.0 + 0.5\max(0.0,0.0,60.0) - 0.0) = 11.00$$
$$Q_3(s_2,a_2) = Q_2(s_2,a_2) + \alpha(R_2 + \lambda \max Q_2(s_3,a') - Q_2(s_2,a_2)) = 0.0 + 0.1(100.0 + 0.5\max(11.0,0.0,0.0) - 0.0) = 10.55$$
$$Q_4(s_3,a_3) = Q_3(s_3,a_3) + \alpha(R_3 + \lambda \max Q_3(s_4,a') - Q_3(s_3,a_3)) = 0.0 + 0.1(60.0 + 0.5\max(10.55,0.0,60.0) - 0.0) = 9.00$$
$$Q_5(s_4,a_4) = Q_4(s_4,a_4) + \alpha(R_4 + \lambda \max Q_4(s_5,a') - Q_4(s_4,a_4)) = 60.0 + 0.1(70.0 + 0.5\max(0.0,0.0,0.0) - 60.0) = 61.00$$
$$Q_6(s_5,a_5) = Q_5(s_5,a_5) + \alpha(R_5 + \lambda \max Q_5(s_6,a') - Q_5(s_5,a_5)) = 0.0 + 0.1(40.0 + 0.5\max(11.0,9.0,0.0) - 0.0) = 4.55$$
$$Q_7(s_6,a_6) = Q_6(s_6,a_6) + \alpha(R_6 + \lambda \max Q_6(s_7,a') - Q_6(s_6,a_6)) = 11.0 + 0.1(20.0 + 0.5\max(0.0,4.55,0.0) - 11.0) = 12.13$$

at that point we find that

$$Q_7 = \begin{bmatrix} 12.127 & 9 & 0 \\ 10.550 & 0 & 61 \\ 0 & 4.550 & 0 \end{bmatrix}$$

c) What is the greedy policy w.r.t. the estimated Q function at the 7th iteration? $\pi(A) = ?, \pi(B) = ?, \pi(C) = ?$. We simply calculate

$$\pi(s) = \arg\max_{a' \in \mathcal{A}} Q_7(s,a')$$

and find that

$$\pi(A) = \arg\max_{a' \in \{a,b,c\}} ({}^{a)}12.127, {}^{b)}\,9, {}^{c)}\,0) = a$$
$$\pi(B) = \arg\max_{a' \in \{a,b,c\}} ({}^{a)}10.550, {}^{b)}\,0, {}^{c)}\,61) = c$$
$$\pi(C) = \arg\max_{a' \in \{a,b,c\}} ({}^{a)}0, {}^{b)}\,4.550, {}^{c)}\,0) = b$$

d) Provide the updated Q-values at the 7th iteration using the SARSA algorithm (initialized with $Q(0)$ as all zeros). Take the first two (state, action, reward)-triplets as those given in your **Answer** to a). Let the discount factor be $\lambda = 0.5$ and the learning rate fixed to $\alpha = 0.1$. **Answer**:
The SARSA algorithm is written

$$Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha\Big(R(s_t,a_t) + \lambda Q_t(s_{t+1},a_{t+1}) - Q_t(s_t,a_t)\Big)$$

If we run these steps using our trajectory of state, action and reward we get the following updates

$$Q_1(s_0,a_0) = Q_0(s_0,a_0) + \alpha(R_0 + \lambda Q_0(s_1,a_1) - Q_0(s_0,a_0)) = 0.0 + 0.1(600.0 + 0.5 \cdot 0.00 - 0.00) = 60.00$$
$$Q_2(s_1,a_1) = Q_1(s_1,a_1) + \alpha(R_1 + \lambda Q_1(s_2,a_2) - Q_1(s_1,a_1)) = 0.0 + 0.1(80.0 + 0.5 \cdot 0.00 - 0.00) = 8.00$$
$$Q_3(s_2,a_2) = Q_2(s_2,a_2) + \alpha(R_2 + \lambda Q_2(s_3,a_3) - Q_2(s_2,a_2)) = 0.0 + 0.1(100.0 + 0.5 \cdot 0.00 - 0.00) = 10.00$$
$$Q_4(s_3,a_3) = Q_3(s_3,a_3) + \alpha(R_3 + \lambda Q_3(s_4,a_4) - Q_3(s_3,a_3)) = 0.0 + 0.1(60.0 + 0.5 \cdot 60.00 - 0.00) = 9.00$$
$$Q_5(s_4,a_4) = Q_4(s_4,a_4) + \alpha(R_4 + \lambda Q_4(s_5,a_5) - Q_4(s_4,a_4)) = 60.0 + 0.1(70.0 + 0.5 \cdot 0.00 - 60.00) = 61.00$$
$$Q_6(s_5,a_5) = Q_5(s_5,a_5) + \alpha(R_5 + \lambda Q_5(s_6,a_6) - Q_5(s_5,a_5)) = 0.0 + 0.1(40.0 + 0.5 \cdot 8.00 - 0.00) = 4.40$$
$$Q_7(s_6,a_6) = Q_6(s_6,a_6) + \alpha(R_6 + \lambda Q_6(s_7,a_7) - Q_6(s_6,a_6)) = 8.0 + 0.1(20.0 + 0.5 \cdot 0.00 - 8.00) = 9.20$$

The resulting Q matrix is

$$Q_7 = \begin{bmatrix} 9.2 & 9 & 0 \\ 10 & 0 & 61 \\ 0 & 4.4 & 0 \end{bmatrix}$$

e) What is the greedy policy at the 7th iteration? $\pi(A) =?, \pi(B) =?, \pi(C) =?$.
We simply calculate

$$\pi(s) = \arg\max_{a' \in \mathcal{A}} Q_7(s, a')$$

and find that

$$\pi(A) = \arg\max_{a' \in \{a,b,c\}} ({}^{a)}9.2, {}^{b)}\, 9, {}^{c)}\, 0) = a$$
$$\pi(B) = \arg\max_{a' \in \{a,b,c\}} ({}^{a)}10, {}^{b)}\, 0, {}^{c)}\, 61) = c$$
$$\pi(C) = \arg\max_{a' \in \{a,b,c\}} ({}^{a)}0, {}^{b)}\, 4.4, {}^{c)}\, 0) = b$$

f) (Tick the correct circle) Are the rewards deterministic?
No, they are not. Time steps $t = 0$ and $t = 4$ have the same state and action, but yield different rewards (600 and 70) respectively.

Part 2: Policy gradient and function approximation

Policy gradients. We consider an episodic RL problem with finite state-space S and action space $A = \{1, \ldots, n+1\}$. For all states s, let $f(s)$ be a real valued function in $[1, 2]$. We parameterize the policy using parameter vector $\theta = (\theta_1, \ldots, \theta_n) \in [0, 1]^n$ according to the following recursion: For $i \in \{1, \ldots, n\}$, initialize $i = 1$ and draw independent random variable Z_i uniformly from $[0, f(s)]$. If $Z_i \leq \theta_i$, choose action $a = i$, otherwise, set $i \leftarrow i + 1$ and repeat. At the last step of the recursion, if $Z_n > \theta_n$, choose $a = n + 1$.
a) Compute in state s, the probability $\pi_\theta(s, i)$ of choosing action i. **Answer**:

$$\pi_\theta(s, 1) =?, \quad \pi_\theta(s, i) =? \text{ for } i \in \{2, \ldots, n\}, \quad \pi_\theta(s, n + 1) =?$$

We let

$$\mathbf{1}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

be the indicator function. We also note that $f_Z(z) = 1/f(s)$, that is we have a uniform probability density function for each Z. $\pi_\theta(s, 1)$ is the probability that we choose the first action, which also happens to be the probability of $p(z \leq \theta_1) = \mathbb{E}[\mathbf{1}_{\{x|x\leq\theta_1\}}(z)]$, which may be calculated below

$$\pi_\theta(s, 1) = \int_0^{f(s)} \mathbf{1}_{\{x|x\leq\theta_1\}}(z) f_Z(z) dz = f(s)^{-1} \int_0^{\min(f(s),p_1)} dz = \frac{\theta_1}{f(s)}$$

We then state that $\pi_\theta(s, 2) = p(z_1 > \theta_1, z_2 \leq \theta_2) = p(z_1 > \theta_1)p(z_2 \leq \theta_2)$ which may also be written $\pi_\theta(s, 2) = (1 - \mathbb{E}[\mathbf{1}_{\{x|x\leq\theta_1\}}(z)])\mathbb{E}[\mathbf{1}_{\{x|x\leq\theta_2\}}(z)]$. We then recognize that

$$\pi_\theta(s, 2) = \left(1 - \frac{\theta_1}{f(s)}\right)\frac{\theta_2}{f(s)}$$

and we infer that

$$\pi_\theta(s, i) = \left[\prod_{k=1}^{i-1}\left(1 - \frac{\theta_k}{f(s)}\right)\right]\frac{\theta_i}{f(s)}, \; i = 1, ..., n$$

(the likelihood of action k is the likelihood of the first $k - 1$ first random variables being greater than their respective parameters, but the k:th random variable being less than or equal to its' corresponding parameter) and that

$$\pi_\theta(s, n + 1) = \prod_{k=1}^{n}\left(1 - \frac{\theta_k}{f(s)}\right)$$

(the likelihood of no random variable being less than or equal to its' corresponding parameter). □

b) What is the Monte-Carlo REINFORCE update of θ upon observing an episode $\tau = (s_1, a_1, r_1, \ldots, s_T, a_T, r_T)$? Provide explicit formulas using the function f, θ, and τ only.

$$\frac{\partial \ln \pi_\theta(s,i)}{\partial \theta_i} =?, \quad \frac{\partial \ln \pi_\theta(s,i)}{\partial \theta_k} =? \text{ for } k < i, \quad \frac{\partial \ln \pi_\theta(s,i)}{\partial \theta_k} =? \text{ for } k > i$$

We state that for $i \leq n$

$$\ln(\pi_\theta(s,i)) = \ln\left(\pi_\theta(s,i) = \left[\prod_{k=1}^{i-1}\left(1 - \frac{\theta_k}{f(s)}\right)\right]\frac{\theta_i}{f(s)}\right) = \ln(\theta_i) - \ln(f(s)) + \sum_{k=1}^{i-1}\ln\left(1 - \frac{\theta_k}{f(s)}\right)$$
$$= \ln(\theta_i) + \sum_{k=1}^{i-1}\ln(f(s) - \theta_k) - i\ln(f(s))$$

and for action $n+1$

$$\ln(\pi_\theta(s,n+1)) = \sum_{k=1}^{n}\ln(f(s) - \theta_k) - n\ln(f(s))$$

We hence state that

$$\frac{\partial \pi_\theta(s,i)}{\partial \theta_i} = \begin{cases} \theta_i^{-1} & \text{if } i \leq n \\ (\theta_i - f(s))^{-1} & \text{otherwise} \end{cases}$$

as is trivial to see from the relation $\frac{\partial \ln(f(x))}{\partial x} = \frac{\partial f(x)}{\partial x}\frac{1}{f(x)}$. We state that for the case where $k < i \leq n$ we write

$$\frac{\partial \pi_\theta(s,i)}{\partial \theta_k} = (\theta_k - f(s))^{-1}$$

and for the case where $k > i$ we may simply write

$$\frac{\partial \pi_\theta(s,i)}{\partial \theta_k} = 0$$

since the k:th parameter is not present in the corresponding expression. □

c) We observe the transition (s_t, a_t, r_t, s_{t+1}). State the Q update in the Q-learning algorithm with function approximation. Why is it a semi-gradient algorithm? **Answer**:
The Q update is as follows

$$\theta \leftarrow \theta + \alpha(r + \gamma \max_{a' \in \mathcal{A}} Q_\theta(s', a') - Q_\theta(s,a))\nabla_\theta Q_\theta(s,a;\theta)$$

where θ are the parameters, γ is the discount factor and α is the step size. We call it a 'semi'-gradient algorithm since the update in the parameters ignores the effect on the target $r + \gamma \max_{a' \in \mathcal{A}} Q_\theta(s', a')$, and hence the gradient is not whole.

d) In the previous updates, the "target" evolves in every step, which could affect the algorithm convergence. What do we mean by target? Can you propose a modification that addresses this problem? **Answer**:
The target is $r + \gamma \max_{a' \in \mathcal{A}} Q_\theta(s', a')$ and by updating θ we are also changing $Q_\theta(s', a')$. One way to address this problem is to use two sets of parameters, the parameters θ and then the independent target parameters ϕ. This way we can use a fixed target for many iterations at a time, and then occasionally let the target parameters catch up to the network parameters. In other words:
Do

$$\theta \leftarrow \theta + \alpha(r + \gamma \max_{a' \in \mathcal{A}} Q_\phi(s', a') - Q_\theta(s,a))\nabla Q_\theta(s,a)$$

C number times, and then do

$$\phi \leftarrow \theta$$

one time. Then repeat.

References