EL2805 Reinforcement Learning

Homework 2

December 2, 2023

Division of Decision and Control Systems
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

Instructions (read carefully):

- Answer the questions of Parts 1 and 2.
- Work in groups of 2 persons.
- **Both** students in the group should upload their scanned report as a .pdf-file to Canvas before December 19, 23:59. The deadline is strict. Please mark your answers directly on this document, and **append** hand-written or typed notes justifying your answers. Reports without justification will not be graded.

Good luck!

1 Part 1. Q-learning and SARSA

Consider a discounted MDP with $\mathcal{S} = \{\text{A, B, C}\}$ and $\mathcal{A} = \{a, b, c\}$. 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:

$$Q^{(0)} = \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} .$$

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

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

where each triplet represents the state, the selected action, and the corresponding reward. Some of the information has been corrupted (marked with question marks) in the above sequence.

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

$$Q^{(2)} = \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 11 & 0 & 0 \\ 0 & 0 & 60 \\ 0 & 0 & 0 \end{bmatrix} \end{array} .$$

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**:

$(\underline{B}, \underline{c}, \underline{600}); (A, \underline{a}, \underline{80}); (B, a, 100); (A, b, 60); (B, c, 70); (C, b, 40); (A, a, 20); (C, c, \ldots)$

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**:

$$Q^{(7)} = \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 12{,}1275 & 9 & 0 \\ 10{,}55 & 0 & 61 \\ 0 & 4{,}55 & 0 \end{bmatrix} \end{array} .$$

c) What is the greedy policy w.r.t. the estimated Q function at the 7th iteration? $\pi(A) = \underline{a}$, $\pi(B) = \underline{c}$, $\pi(C) = \underline{b}$.

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**:

$$Q^{(7)} = \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 9{,}2 & 9 & 0 \\ 10 & 0 & 61 \\ 0 & 4{,}4 & 0 \end{bmatrix} \end{array} .$$

e) What is the greedy policy at the 7th iteration? $\pi(A) = \underline{a}$, $\pi(B) = \underline{c}$, $\pi(C) = \underline{b}$.

f) (Tick the correct circle) Are the rewards deterministic? ○ Yes - ⊗ No

a) Since we have only two non-zero entries the possibilities are

$$(A,a) \quad \text{or} \quad (B,c)$$

However, since we know that the second observed state is A then the we know in Step 1: $(B, C, r_1(B,C))$

$$Q^{(1)}(B,C) = 0 + 0{,}1 \cdot [r(B,C) + \lambda \cdot \max_{a'} Q^{(0)}(A,a') - 0]$$

$$60 \stackrel{!}{=} Q^{(1)}(B,C) = 0{,}1 \cdot r(B,C)$$

$$\Rightarrow \boxed{r(B,C) = 600}$$

• Step 2: $(A, a, r_2(A,a))$

$$Q^{(2)}(A,a) = 0 + 0{,}1 \cdot [r(A,a) + \lambda \cdot \max_{a'} Q^{(1)}(B,a') - 0]$$

$$11 \stackrel{!}{=} Q^{(2)}(A,a) = 0{,}1 \cdot r(A,a) + 0{,}05 \cdot 60$$

$$\Rightarrow \boxed{r(A,a) = 80}$$

b)

• Step 3: $Q^{(3)}(B,a) = 0 + 0{,}1 \cdot [100 + 0{,}5 \cdot \max_{a'} Q^{(2)}(A,a') - 0]$
$= 10 + 0{,}05 \cdot 11 = \underline{\underline{10{,}55}}$

• Step 4: $Q^{(4)}(A,b) = 0 + 0{,}1 \cdot [60 + 0{,}5 \max_{a'} Q^{(3)}(B,a') - 0]$
$= 6 + 0{,}05 \cdot \max_{a'} Q^{(3)}(B,a')$
$= 6 + 0{,}05 \cdot 60 = \underline{\underline{9}}$

• Step 5: $Q^{(5)}(B,c) = 60 + 0{,}1 \cdot [70 + 0{,}5 \max_{a'} Q^{(4)}(C,a') - 60]$
$= 60 + 7 + 0{,}05 \max_{a'} Q^{(4)}(C,a') - 6 = \underline{\underline{61}}$

• Step 6: $Q^{(6)}(C,b) = 0 + 0{,}1 \cdot [40 + 0{,}5 \max_{a'} Q^{(5)}(A,a') - 0]$
$= 4 + 0{,}05 \cdot 11 = \underline{\underline{4{,}55}}$

• Step 7: $Q^{(7)}(A,a) = 11 + 0{,}1 \cdot [20 + 0{,}5 \max_{a'} Q^{(6)}(C,a') - 11] = 11{,}0 + 2 + 0{,}05 \cdot 4{,}55 = \underline{\underline{12{,}1275}}$

c) Using $Q^{(7)}(A,a)$ we can extract the greedy policy:

$$\pi^{(7)}(s) = \underset{a'}{\operatorname{argmax}}\ Q^{(7)}(s,a')$$

$$\Rightarrow \quad \pi^{(7)}(A) = a$$
$$\pi^{(7)}(B) = c$$
$$\pi^{(7)}(C) = b$$

d) SARSA-Updates: $Q^{(k+1)}(S,A) = Q^{(k)}(S,A) + \alpha \cdot [r(S,A) + \lambda \cdot Q^{(k)}(S',A') - Q^{(k)}(S,A)$

Step 1: $Q^{(1)}(B,C) = 0 + 0{,}1 \cdot [600 + 0{,}5 \cdot Q^{(0)}(A,a) - 0]$
$= \underline{\underline{60}}$

Step 2: $Q^{(2)}(A,a) = 0 + 0{,}1 \cdot [80 + 0{,}5\, Q^{(1)}(B,a) - 0]$
$= \underline{8}$

Step 3: $Q^{(3)}(B,a) = 0 + 0{,}1 \cdot [100 + 0{,}5 \cdot Q^{(2)}(A,b) - 0]$
$= \underline{10}$

Step 4: $Q^{(4)}(A,b) = 0 + 0{,}1 \cdot [60 + 0{,}5 \cdot Q^{(3)}(B,C) - 0]$
$= \underline{\underline{9}}$

Step 5: $Q^{(5)}(B,C) = 60 + 0{,}1 \cdot [70 + 0{,}5 \cdot Q^{(4)}(C,b) - 60]$
$= 60 + 7 - 6 = \underline{\underline{61}}$

Step 6: $Q^{(6)}(C,b) = 0 + 0{,}1 \cdot [40 + 0{,}5 \cdot Q^{(5)}(A,a) - 0]$
$= 4 + 0{,}05 \cdot 8 = \underline{\underline{4{,}4}}$

Step 7: $Q^{(7)}(A,a) = 8 + 0{,}1 \cdot [20 + 0{,}5 \cdot Q^{(6)}(C,c) - 8]$
$= 8 + 2 - 0{,}8 = \underline{\underline{9{,}2}}$

<u>Note</u>: Here we use the next state, next (s_{t+1}, a_{t+1}) action pair that was observed to update $\underline{\underline{Q^{(t)}(s_t, a_t)}}$!

e) Again the greedy-policy w.r.t $Q^{(7)}$ is $\pi^{(7)}(s) = \underset{a'}{\operatorname{argmax}}\ Q^{(7)}(s,a')$

$$\pi^{(7)}(A) = a$$
$$\pi^{(7)}(B) = C$$
$$\pi^{(7)}(C) = b$$

f) Rewards is clearly not deterministic, since for the state-action pair (B,C) we observe the reward 600 in the first step and 70 in the fifth step.
The same observation can be made for $(s_t, a_t) = (A, a)$, where $r(A,a) = 140$ in the 2^{nd}-step.
$r(A,a) = 20$ in the 7^{th}-step.

2 Part 2: policy gradient and function approximation

Policy gradients. We consider an episodic RL problem with finite state-space $\mathcal{S}$ and action space $\mathcal{A} = \{1, \ldots, n+1\}$. 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 $Z_i \le \theta_i$, 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**:

$$\pi_\theta(s, 1) = \theta_1 / f(s)$$

$$\pi_\theta(s, i) = \left(\prod_{j=1}^{i-1} \left(1 - \frac{\theta_j}{f(s)}\right) \right) \cdot \frac{\theta_i}{f(s)} \qquad \text{for } i \in \{2, \ldots, n\}$$

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

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.

$$\frac{\partial \ln \pi_\theta(s, i)}{\partial \theta_i} = \frac{1}{\theta_i}$$

$$\frac{\partial \ln \pi_\theta(s, i)}{\partial \theta_k} = \frac{1}{\theta_k - f(s)} \qquad \text{for } k < i$$

$$\frac{\partial \ln \pi_\theta(s, i)}{\partial \theta_k} = 0 \qquad \text{for } k > i$$

Off-policy control with function approximation. Consider a discounted RL problem, that we wish to solve using approximations of the (state, action) value function (i.e., parametrized by vector θ).

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:**

Update: $\theta^{(k+1)} := \theta^{(k)} + \alpha \cdot \left(r_t + \lambda \cdot \max_b Q_\theta(s_{t+1}, b) - Q_\theta(s_t, a_t)\right) \cdot \nabla_\theta Q_\theta(s_t, a_t)$

It is a semi-gradient method since we only consider the part of the gradient w.r.t the estimated Q_θ-f^{cn} but ignore the effect of the target which also depends on θ!

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 the term: $r_t + \lambda \max_b Q_\theta(s_{t+1}, b)$, which we back our estimate in the direction of. We fix the weight vectors of the target function for a couple of steps k and after that we update the weight vectors of the target function by setting them equal to the weight vectors of the estimated Q-function after those k update steps! See Pseudocode below!

a) Setting: Finite state space $\mathcal{S}$

Action Space $\mathcal{A} := \{1, \dots, n+1\}$

Furthermore, we have $f: \mathcal{S} \to [1,2]$ and the policy is parametrized using $\theta = [\theta_1, \dots, \theta_n]^T \in [0,1]^n$ according to the following recursion: ① Initialize $i=1$

for $i = 1, \dots, n$

② Draw independen random variable Z_i uniformly from $[0, f(s)]$

③ If $z_i \leq \theta_i$

choose action $a = i$

$i \neq n$ else

$i \leftarrow i+1$ and jump to ②

$i=n$: Only at the last step:

else: If $z_n > \theta_n$, choose action $a = n+1$

- For choosing action $a=1$, the random variable Z_1 (which is uniformly distributed with support $[0, f(s)]$) must be less or equal to θ_1:

$$P(z_1 \leq \theta_1) = \int_0^{\theta_1} \frac{1}{f(s)} dz_1 = \frac{\theta_1}{f(s)}$$

$\Rightarrow$ Hence, the probability of selecting action a_1: $P[a = a_1] = \frac{\theta_1}{f(s)}$

- For choosing action $a=2$, the random variable Z_1 must be greater than θ_1 and Z_2 (which is again uniformly distributed with support $[0, f(s)]$ must be less or equal than θ_2

Hence $P[a = a_2] = \left(1 - \frac{\theta_1}{f(s)}\right) \cdot \frac{\theta_2}{f(s)}$

⋮

We can generalize this result for $i \in \{2, \dots, n\}$

$$P[a = a_i] = \left(\prod_{j=1}^{i-1} \left(1 - \frac{\theta_j}{f(s)}\right) \right) \cdot \frac{\theta_i}{f(s)}$$

b) $\frac{\partial \ln \pi_\theta(s,i)}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} \left[\ln \left(\prod_{j=1}^{i-1} \left(1 - \frac{\theta_j}{f(s)}\right) \cdot \frac{\theta_i}{f(s)} \right) \right] = \frac{\partial}{\partial \theta_i} \left[\sum_{j=1}^{i-1} \ln\left(1 - \frac{\theta_j}{f(s)}\right) + \ln(\theta_i) - \ln(f(s)) \right]$

$$= \frac{1}{\theta_i}$$ This also holds for $i=1$

Next, we consider $\frac{\partial \ln \pi_\theta(s,i)}{\partial \theta_k}$, where $k<i$:

Since $k<i$, we only focus on the second case:

$$\frac{\partial \ln \pi_\theta(s,i)}{\partial \theta_k} = \frac{\partial}{\partial \theta_k}\left(\ln\left(\left(\prod_{j=1}^{i-1}\left(1-\frac{\theta_j}{f(s)}\right)\right)\cdot \frac{\theta_i}{f(s)}\right)\right) = \frac{\partial}{\partial \theta_k}\left(\sum_{j=1}^{i-1}\ln\left(1-\frac{\theta_j}{f(s)}\right)+\ln\left(\frac{\theta_i}{f(s)}\right)\right)$$

$$= \frac{\partial}{\partial \theta_k}\left(\ln\left(1-\frac{\theta_k}{f(s)}\right)\right) = \frac{\partial}{\partial \theta_k}\left(\ln\left(f(s)-\theta_k\right)-\ln\left(f(s)\right)\right)$$

$$= \underline{\underline{\frac{1}{\theta_k - f(s)}}}$$

Lastly, we consider $\frac{\partial \ln \pi(s,i)}{\partial \theta_k}$ where $k>i$

$$\frac{\partial \ln \pi(s,i)}{\partial \theta_k} = \frac{\partial}{\partial \theta_k}\left[\ln\left(\left(\prod_{j=1}^{i-1}\left(1-\frac{\theta_j}{f(s)}\right)\right)\cdot\frac{\theta_i}{f(s)}\right)\right] \overset{\text{Since } k>i}{=} \underline{\underline{0}}$$

Hence the <u>REINFORCE update rule</u> is:

$$\theta^{(r+1)} = \theta^{(r)} + \alpha_r \left(\sum_{t=1}^{T} r_t\right)\cdot\left(\sum_{t=1}^{T}\nabla \ln \pi_\theta(s_t, a_t)\right)$$

d) $$\theta^{(k+1)} = \theta^{(k)} + \alpha\left(r_t + \lambda \max_b \underline{Q_\phi}(s_{t+1}, b) - Q_\theta(s_t,a_t)\right)\cdot \nabla_\theta Q_\theta(s_t,a_t)$$

Other weight vectors!

After k update steps: $\phi \leftarrow \theta$