EL2805 Reinforcement Learning

Exam – January 2022

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

Exam (tentamen), January 11, 2022, kl 14.00 - 19.00

Aids. Slides of the lectures (**not exercises**), lecture notes (summary.pdf), mathematical tables.

Observe. Do not treat more than one problem on each page. Each step in your solutions must be motivated. Write a clear answer to each question. Write name and personal number on each page. Please only use one side of each sheet. Mark the total number of pages on the cover.

The exam consists in 5 problems. The distribution of points among these problems is indicated below.

Grading.
Grade A: ≥ 43 Grade B: ≥ 38
Grade C: ≥ 33 Grade D: ≥ 28
Grade E: ≥ 23 Grade Fx: ≥ 21

Responsible. Alexandre Proutiere 087906351

Results. Posted no later than January 25, 2022

Good luck!

Problem 1 - Quiz

(a) Consider a stationary discounted MDP with random rewards. Assume that you know the probability distributions of the rewards when action a is chosen in state s for all (state, action) pairs (s, a). Hence you know the average reward $\mu(s, a)$ when action a is chosen in state s. Consider now the Q-learning algorithm with the following target value in round t: $y_t = r_t + \lambda \max_{a'} Q_t(s_{t+1}, a')$. Would the algorithm work if we changed r_t (the reward you observe) by $\mu(s_t, a_t)$? [1 pt]

(b) Consider a stationary discounted MDP. Why is it important to use a discount factor in $(0, 1)$ in Q-learning? [1 pt]

(c) Is the policy gradient algorithm based on the stochastic approximation or the stochastic gradient method? [1 pt]

(d) Consider a stationary discounted MDP. Explain the connection between the value function and the Q-function. [1 pt]

(e) What is the use of the 'critic' in actor-critic algorithms? [1 pt]

(f) Consider an irreducible aperiodic markov chain with time-homogenous dynamics. Describe the connection between its stationary distribution π and its transition probability matrix P. [1 pt]

(g) A professor owns n unique books, $\{b_1, \ldots, b_n\}$ which are placed in a single bookshelf. Each morning the professor selects a book from the shelf uniformly at random, reads it for 30 minutes and moves it to the left-most position on the shelf. This describes a markov chain in which the state space, S, consists of book-orderings of the form $s = b_{i_1} \ldots b_{i_n}$, $|S| = n!$, and in which the dynamics are described by the professor's "randomly-select-and-move-to-front" behavior. Does every pair of states in this chain communicate? [1 pt]

(h) Consider a finite time horizon MDP $(\mathcal{S}, \mathcal{A}, T, (p_t)_{t\in[T]}, (r_t)_{t\in[T]})$ with non-stationary transitions and rewards. Does such a problem in general admit a stationary policy as an optimal policy? [1 pt]

(i) Consider a finite time horizon MDP $(\mathcal{S}, \mathcal{A}, T, (p_t)_{t\in[T]}, (r_t)_{t\in[T]})$ with non-stationary transitions and rewards. Propose a modification of the state space in this MDP so that the MDP enjoys stationary transitions and rewards. [2 pts]

Problem 1 – Solution

(a) No it wouldn't make a difference, since in expectation $\mathbb{E}[r_t] = \mu_t$.

(b) Without a discount factor we are not guaranteed to have the contraction property. Without the contraction property, we are not guaranteed to converge to the fixed point of the Bellman's equation.

(c) Stochastic gradient.

(d) $V^\star(s) = \max_{a \in A_s} Q(s, a)$.

(e) To evaluate the policy corresponding to the current parameter.

(f) π is a left eigenvector of P with eigenvalue 1; $\pi P = \pi$.

(g) Yes. To reach any state from any other state $s = b_{i_1} \dots b_{i_n}$, a valid trajectory of length $n-1$ is $b_{i_{n-1}} \dots b_{i_1}$ which occurs with probability $\left(\frac{1}{n}\right)^{n-1}$.

(i) In general the optimal policy for finite horizon MDP problems is not stationary. Yes, such problem can admit a stationary policy as an optimal one. Example: if the rewards are constant across actions and states for every step t, $r_t(\cdot, \cdot) = \text{cst}$.

(j) The solution is $(\mathcal{S}', \mathcal{A}, T, p', r')$ where $\mathcal{S}' = \mathcal{S} \times [T]$, $p'((\cdot, t+1)|(s,t), a) = p_t(\cdot|s, a)$, and $r'((s,t), a) = r_t(s, a)$

Problem 2. The Robot Vacuum Cleaner

You work for a firm designing automated vacuum cleaners and have been tasked to beta-test your newest cleaner over the period of 90 days at your home. You live in a *square* flat that is 36 square meters[1] and will be starting the cleaner each day. The flat is divided into 36 square cells, each of surface 1 m^2. Your flat is characterized by the fact that, provided it was cleaned the previous day, each morning there is new dust distributed over your entire apartment, drawn independently (over days) according to some distribution $q(\cdot)$. Each square cell of your flat is either a clean, dusty, or very dusty each morning. Hence, the distribution $q(\cdot)$ describes the joint probability that each cell of your flat is clean, dusty, or very dusty each morning.

Each morning, before the robot starts moving, it observes the state of each cell (clean, dusty, or very dusty). Then, the robot decides on a path through your apartment. The choice of the path is constrained by the facts that: (i) the robot always starts in the same position, the bottom left cell of your flat, (ii) it visits every unclean cell and (iii) it returns to the bottom left at the end of the path. The robot can move vertically and horizontally but cannot move diagonally. The robot spends 1 minute on an already clean cell of your flat, 2 minutes on a dusty cell and 3 minutes a very dusty cell. We assume that the time it takes for the robot to move from one cell to neighbouring cells in negligible. If a cell has already been visited, assume that the robot still spends one minute there. Note that the path taken by the robot is decided up front each morning and cannot be revised while the robot is cleaning.

The objective is to design a robot minimizing the expected time spent to clean the flat over the 90 days.

(a) Model the problem as an MDP. Do not solve the MDP. (Remember that the robot has access to the amount and location of the dust in the flat each morning). [6 pts]

Your employer does not like the like the assumption that the robot knows the amount and location of the dust. To keep your job as the firm's leading robot vacuum designer, you make another attempt.

(b) Model the problem as an MDP. Do not solve the MDP. This time, assume the robot does not have access to the amount and location of the dust. Here again, the path has to be decided up front before the robot starts moving. [Hint: use random rewards] [4 pts]

[1] Assume the room is 6-by-6 and does not possess any walls except the exterior walls.

Solution

a) The state space may be defined as follows. Let $V = \{s = (s_1, s_2) \in \mathbb{Z}^2 | 0 \leq s_i \leq 5\}$ and $N = \{1, 2, 3\}$ defines how dusty the part of V in the morning. Recall that the notation N^V denotes functions from V to N. Then $S = N^V$ and each element of $s \in S$ is a function $s = n(\cdot)$ taking vertices as inputs and dustiness $\in \{1, 2, 3\}$ as ouput. We define the graph G by letting V be its vertices and drawing an edge between every adjacent vertex (we define adjacency by the coordinates of S differing by at most one entry by magnitude exactly 1). The actions can now be written as

$$A_s = \{\text{Paths over the graph } G \text{ that begin at } (0,0) \\ \text{and terminate at } (0,0) \text{ and visit every unclean point of } G\}$$

where a point v is unclean if $n(v) \neq 1$.

If the action a is a path, the rewards are

$$r(s, a) = -\left(\sum_{v \in a} n(v) + \max\{|\text{occurences of } v \in a| - 1, 0\}\right)$$

which is the negative total time spent cleaning the apartment. To be precise summation above treats a as a set and counts multiple occurences of the same vertex once. Repeated visits are accounted for by the term containing the max.

The goal is to maximize

$$\mathbb{E}^{\pi} \sum_{t=0}^{89} r(s_t, a_t).$$

Finally, the transitions are given by $p(s'|s, a) = q(s')$.

b) There no longer is a state space/singleton state – the problem is a bandit. Otherwise, the problem can be modelled exactly as above with the key distinction that S no longer holds significance as a state space. Instead, the dustiness level of your flat s becomes an auxilliary random variable only used to compute the rewards (which are now random functions of your actions).

Problem 3. How many doses?

In pandemic times, you may decide to update your vaccin at the beginning of every semester. More precisely, when the semester starts, you observe a variable α describing how contagious the virus will be in the coming semester. We assume that $\alpha \in \{0, 1/N, 2/N, \dots, 1\}$ for some integer $N > 1$. The variables α are i.i.d. across semesters and have distribution g (i.e., $g(u)$ is the probability that $\alpha = u$ for any $u \in \{0, 1/N, 2/N, \dots, 1\}$).

You adhere to a vaccin program with a maximum of 3 doses. After observing α, you may decide to get a new dose (you can only take one dose per semester). The cost of a dose is $C > 0$. If you have $i \in \{0, 1, 2, 3\}$ doses, the probability that you get the virus in the coming semester is αp_i, in which case you die. Death is modelled as a cost of $D > 0$. Costs are discounted by a factor $\lambda \in (0, 1)$ every semester. At the beginning, you haven't received any vaccin dose.

(a) Model the problem of minimzing the expected discounted cost as an MDP. [4 pts]

(b) If at the beginning of a semester, you observe that $\alpha = u$ and you have received 3 doses already, what is the expected minimal cost starting at this state? [Hint: first compute the corresponding average (over α)] [3 pts]

(c) If at the beginning of a semester, you observe that $\alpha = u$ and you have received 2 doses already, write the equation satisfied by the value function at this state. Deduce that the optimal strategy is threshold based (i.e., the optimal decision to take a new dose solely depends on whether u is above or below a given threshold). [2 pts]

(d) Are optimal decisions for the first and the second doses also threshold based? [1 pt]

Problem 3. Solution.

(a) We consider an infinite time horizon discounted MDP with discount factor λ.
States. The state should include the information available to the decision maker, and it should be defined so as to get Markovian dynamics. The state is hence recording the number of doses received so far, and the factor α observed at the beginning of the semester.

$$s = (\alpha, i),$$

where $i \in \{0, 1, 2, 3\}$ and $\alpha \in \{0, 1/N, 2/N, \ldots, 1\}$. We also need a state representing death, say d, and a terminal state $\emptyset$.

Actions. In states (i, α) for $i < 3$, the actions available are ND (new dose) or S (skip).

Transition probabilities.

$$\begin{gathered}
p((\alpha', i)|(\alpha, i), S) = g(\alpha')(1 - \alpha p_i), \\
p((\alpha', i+1)|(\alpha, i), ND) = g(\alpha')(1 - \alpha p_{i+1}), \quad i < 3, \\
p(d|(\alpha, i), S) = \alpha p_i, \\
p(d|(\alpha, i), ND) = \alpha p_{i+1}, \quad i < 3, \\
p(\emptyset|d, \cdot) = 1.
\end{gathered}$$

Rewards. $r((\alpha, i), S) = 0$, $r((\alpha, i), ND) = -C$, $r(d, \cdot) = -D$.

(b) Having received 3 doses already, there is nothing to optimize. The expected minimal cost from state $(u, 3)$ is the value function in that state:

$$V^\star(u, 3) = \lambda(1 - up_3) \sum_\alpha g(\alpha) V(\alpha, 3) - \lambda u p_3 D.$$

Define $V_3 := \sum_\alpha g(\alpha) V(\alpha, 3)$. Let us compute V_3 from the above equation. Multiply it by $g(u)$ and sum over u, you get:

$$V_3 = \lambda \sum_u g(u)(1 - up_3) V_3 - \lambda D p_3 \sum_u g(u) u,$$

and hence:

$$V_3 = -\frac{\lambda D p_3 \sum_\alpha g(\alpha)\alpha}{1 - \lambda \sum_\alpha g(\alpha)(1 - \alpha p_3)}.$$

Finally:

$$V^\star(u, 3) = -\lambda(1 - up_3) \frac{\lambda D p_3 \sum_\alpha g(\alpha)\alpha}{1 - \lambda \sum_\alpha g(\alpha)(1 - \alpha p_3)} - \lambda u p_3 D.$$

(c) We are in state $(u, 2)$. Bellman's equation yields:

$$V^\star(u, 2) = \max \left\{ \lambda(1 - up_2) V_2 - \lambda u p_2 D, -C + \lambda(1 - up_3) V_3 - \lambda u p_3 D \right\}.$$

where $V_2 = \sum_\alpha g(\alpha) V(\alpha, 2)$. Hence it is optimal to get a new dose in this state if and only if:

$$\lambda(1 - up_2) V_2 - \lambda u p_2 D \leq -C + \lambda(1 - up_3) V_3 - \lambda u p_3 D.$$

Equivalently if and only if:

$$u(p_3(V_3 + D) - p_2(V_2 + D)) \geq V_2 - V_3 + C/\lambda.$$

We have a threshold based optimal decision.

(d) We have the same results for previous decisions. For example, to decide to get the second dose, we write:

$$V^\star(u, 1) = \max \left\{ \lambda(1 - up_1) V_1 - \lambda u p_1 D, -C + \lambda(1 - up_2) V_2 - \lambda u p_2 D \right\}.$$

where $V_1 = \sum_\alpha g(\alpha) V(\alpha, 1)$.

Problem 4

Exercise 1 Consider a discounted MDP with $\mathcal{S} = \{A, B, C\}$ and $\mathcal{A} = \{u_1, u_2, u_3\}$. We plan to use the Q-learning algorithm to learn to control the system. We initialize the following Q-function

$$Q^{(0)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} .$$

The discount factor is λ and the fixed learning rate is α. The observed trajectory is as follows (for these transitions, we are imposed a policy):

$$(B, u_2, 1); (A, u_2, 2); (C, u_1, 55); (A, u_3, 17); (B, u_3, 32); (B, u_2, 42); (A, u_1, \dots)$$

where each triplet represents the state, the selected action, and the corresponding reward.

(a) Provide the updated Q-values as a function of λ and α, using the Q-learning algorithm, at the 6th iteration. [3 pts]

(b) What is the greedy policy at the 6th iteration? [0.5 pt]

(c) Are the rewards deterministic? Justify your answer. [0.5 pt]

Exercise 2 Consider an MDP similar to the previous one. The reward function is

$$r(s, a) = \mathbb{1}\{s = A \text{ and } a = u_1\} + \mathbb{1}\{s = B \text{ and } a = u_2\} + \mathbb{1}\{s = C \text{ and } a = u_3\}, \quad s \in \{A, B, C\}.$$

Consider a randomized policy π_c, selecting actions uniformly at random in any state.

(a) Write equations satisfied by the state value function of this policy. [0.5 pt]

(b) Write equations satisfied by the (state, action) value function of this policy. [0.5 pt]

(c) To estimate the (state, action) value function of this policy, you run TD learning algorithm: when in state s_t at time t, select an action $a_t \sim \pi_c$, and update

$$Q^{(t+1)}(s_t, a_t) = Q^{(t)}(s_t, a_t) + \frac{1}{t}\left(y_t - Q^{(t)}(s_t, a_t)\right),$$

where the target is

$$y_t = r_t + \lambda \mathbb{E}_{a \sim \pi_c}[Q^{(t)}(s_{t+1}, a)].$$

Assume the Robbins-Monroe conditions are satisfied. To which function does $\lim_{t\to\infty} Q^{(t)}$ converge to? Can you guess the limit (state, action) values in (A, u_1), (A, u_2), (A, u_3)? [2 pts]

Exercise 3
(a) Write the pseudo-code of an algorithm that uses SARSA with a function approximator and ε-greedy policy. Assume to have a discount factor λ and infinite time-horizon. [1.5 pts].
(b) Propose a modification of the above SARSA algorithm with clearly separated policy evaluation and policy improvement steps. [1.5 pts]

Solutions - Problem 4 - exercise 1

(A) The trajectory is

$$(B, u_2, 1); (A, u_2, 2); (C, u_1, 55); (A, u_3, 17); (B, u_3, 32); (B, u_2, 42); (A, u_1, \dots)$$

Therefore the Q-values in the different rounds are:

1. Step 1: the update is
$$Q^{(1)}(B, u_2) = \alpha,$$
thus
$$Q^{(1)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}.$$

2. Step 2: the update is
$$Q^{(2)}(A, u_2) = \alpha(2 + \lambda),$$
thus
$$Q^{(2)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & \alpha(2+\lambda) & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}.$$

3. Step 3: the update is
$$Q^{(3)}(C, u_1) = \alpha(55 + \lambda\alpha(2 + \lambda)),$$
thus
$$Q^{(3)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & \alpha(2+\lambda) & 0 \\ 0 & \alpha & 0 \\ \alpha(55+\lambda\alpha(2+\lambda)) & 0 & 1 \end{bmatrix} \end{array}.$$

4. Step 4: the update is
$$Q^{(4)}(A, u_3) = \alpha(17 + \lambda\alpha),$$
thus
$$Q^{(4)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & \alpha(2+\lambda) & \alpha(17+\lambda\alpha) \\ 0 & \alpha & 0 \\ \alpha(55+\lambda\alpha(2+\lambda)) & 0 & 1 \end{bmatrix} \end{array}.$$

5. Step 5: the update is
$$Q^{(5)}(B, u_3) = \alpha(32 + \lambda\alpha),$$
thus
$$Q^{(5)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & \alpha(2+\lambda) & \alpha(17+\lambda\alpha) \\ 0 & \alpha & \alpha(32+\lambda\alpha) \\ \alpha(55+\lambda\alpha(2+\lambda)) & 0 & 1 \end{bmatrix} \end{array}.$$

6. Step 6: the update is
$$Q^{(6)}(B, u_2) = \alpha(43 + \lambda\alpha(17 + \lambda\alpha) - \alpha),$$
thus
$$Q^{(6)} = \begin{matrix} \\ A \\ B \\ C \end{matrix} \begin{matrix} \begin{matrix} u_1 \end{matrix} \\ \begin{bmatrix} 0 \\ 0 \\ \alpha(55 + \lambda\alpha(2 + \lambda)) \end{bmatrix} \end{matrix} \begin{matrix} \begin{matrix} u_2 \end{matrix} \\ \begin{bmatrix} \alpha(2 + \lambda) \\ \alpha(43 + \lambda\alpha(17 + \lambda\alpha) - \alpha) \\ 0 \end{bmatrix} \end{matrix} \begin{matrix} \begin{matrix} u_3 \end{matrix} \\ \begin{bmatrix} \alpha(17 + \lambda\alpha) \\ \alpha(32 + \lambda\alpha) \\ 1 \end{bmatrix} \end{matrix}.$$

(B) In A the optimal action is clearly u_3. In B the optimal action is u_2 as long as
$$43 + \lambda\alpha(17 + \lambda\alpha) - \alpha > 32 + \lambda\alpha,$$
and u_3 otherwise. In C the optimal action is u_1 as long as
$$\alpha(55 + \lambda\alpha(2 + \lambda)) > 1$$
and u_3 otherwise.
(C) The rewards are not deterministic since we observed different rewards for the tuple (B, u_2).

Solutions - Problem 4 - Exercise 2

(A) We simply have
$$V^{\pi_c}(s) = \sum_a \pi_c(a|s) \left[r(s, a) + \lambda \sum_{s'} P(s'|s, a) V^{\pi_c}(s') \right]$$

(B)
$$Q^{\pi_c}(s, a) = r(s, a) + \lambda \sum_{s'} P(s'|s, a) V^{\pi_c}(s')$$

(C) Since the policy is sufficiently exploratory, and the R.M. conditions are satisfied, the Q function converges to the Q-values of the uniform policy π_c. To see this, note that
$$\mathbb{E}[y_t|(s_t, a_t)] = r(s_t, a_t) + \lambda \sum_{s'} P(s'|s_t, a_t) \sum_{a'} \pi_c(a'|s') Q^{(t)}(s', a')$$

To compute the Q-values, note that
$$\begin{aligned}
Q^{\pi_c}(s, a) &= r(s, a) + \lambda \sum_{s'} P(s'|s, a) V^{\pi_c}(s'), \\
&= r(s, a) + \lambda \sum_{s'} P(s'|s, a) \sum_{a'} \pi_c(a'|s') \left[r(s', a') + \lambda \sum_{s''} P(s''|s', a') V^{\pi_c}(s''), \right], \\
&= r(s, a) + \frac{\lambda}{3} \sum_{s'} P(s'|s, a) \sum_{a'} r(s', a') + \frac{\lambda}{3} \sum_{s'} P(s'|s, a) \sum_{a'} \lambda \sum_{s''} P(s''|s', a') V^{\pi_c}(s''), \\
&= r(s, a) + \frac{\lambda}{3} + \frac{\lambda^2}{3} \sum_{s'} P(s'|s, a) \sum_{a'} \sum_{s''} P(s''|s', a') V^{\pi_c}(s''), \\
&= r(s, a) + \frac{\lambda}{3} + \frac{\lambda^2}{3} + \frac{\lambda^3}{3} + \dots, \\
&= r(s, a) + \frac{\lambda}{3} \sum_{n=0}^{\infty} \lambda^n = r(s, a) + \frac{\lambda}{3(1 - \lambda)}
\end{aligned}$$

Then $Q(A, u_2) = Q(A, u_3) = \frac{1}{3} \frac{\lambda}{1-\lambda}$, $Q(A, u_1) = 1 + \frac{1}{3} \frac{\lambda}{1-\lambda}$.

Solutions - Problem 4 - Exercise 3

(A) We start by parametrizing the Q-values using a parameter θ and we write Q_θ. Note that we can't use an experience replay (otherwise it's off-policy). We perform the update of θ using a semi-gradient approach.

Algorithm 1: Question 1: SARSA with function approximation

Input Discount factor γ

Procedure

Initialize Q_θ
$t \leftarrow 0$
Initialize environment and read initial state s_0; Sample a_0 using a ε-greedy policy w.r.t. Q_θ
while Q_θ has not converged **do**
    Take ε-greedy action a_t (with respect to $Q_\theta(s_t, \cdot)$)
    Observe s_{t+1}, r_t and sample a_{t+1} w.r.t. $Q_\theta(s_{t+1}, \cdot)$.
    Compute the target values $y_t = r_t + \lambda Q_\theta(s_{t+1}, a_{t+1})$ and let $\delta_t = y_t - Q_\theta(s_t, a_t)$ be the TD-error.
    Update θ by performing a gradient descent step

$$\theta \leftarrow \theta + \alpha_t \delta_t \nabla_\theta Q_\theta(s_t, a_t)$$

    $t \leftarrow t + 1$
end while

(B) For this question we use two networks, one parametrized by θ and another one parametrized by θ'. θ' will be used to evaluate the policy θ. The policy θ is then improved by setting $\theta \leftarrow \theta'$.

Algorithm 2: Question 2: SARSA with function approximation

Input Discount factor γ; Policy improvement period T

Procedure

Initialize $Q_\theta, Q_{\theta'}$
$t \leftarrow 0$
Initialize environment and read initial state s_0; Sample a_0 using a ε-greedy policy w.r.t. Q_θ
while Q_θ has not converged **do**
    Take ε-greedy action a_t (with respect to $Q_\theta(s_t, \cdot)$)
    Observe s_{t+1}, r_t and sample a_{t+1} w.r.t. $Q_\theta(s_{t+1}, \cdot)$.
    Compute the target values $y_t = r_t + \lambda Q_{\theta'}(s_{t+1}, a_{t+1})$ and let $\delta_t = y_t - Q_{\theta'}(s_t, a_t)$ be the TD-error.
    Update θ' by performing a gradient descent step

$$\theta' \leftarrow \theta' + \alpha_t \delta_t \nabla_{\theta'} Q_{\theta'}(s_t, a_t)$$

    Policy improvement: every T steps do $\theta \leftarrow \theta'$
    $t \leftarrow t + 1$
end while

Problem 5

Control problems. Consider a control problem where the state and action spaces are continuous. More precisely, $\mathcal{S} \subseteq \mathbb{R}^d$ and $\mathcal{A} \subseteq \mathbb{R}$. The dynamics of the system are evolving according to $s_{t+1} = f(s_t, a_t) + \varepsilon_t$ for all $t \geq 1$, and $s_1 = s_{init}$. The sequence of disturbances $(\varepsilon_t)_{t\geq 1}$ are drawn independently from each other according to a standard gaussian distribution[2] $\mathcal{N}(0, 1)$. You do not know the function f that describes the dynamics of the system, yet you wish to minimize the cumulative cost $\sum_{t=1}^{T} c_t(s_t, a_t)$. To that end, you decide to use a policy gradient method. You decide to parameterize your policy by $\theta \in \mathbb{R}^d$, where at time t, you first sample η_t from the gaussian distribution $\mathcal{N}(0, \sigma^2)$ then select an action as follows:

$$a_t = \theta^\top \phi(s_t) + \eta_t. \tag{1}$$

You may assume that the feature map ϕ is known to you.

(a) Verify that $\pi_\theta(a|s)$ is the density of the gaussian distribution $\mathcal{N}(\theta^\top \phi(s), \sigma^2)$. *(Hint: a random variable X is uniquely determined by its cumulative distribution function $F_X(x) = \mathbb{P}(X \leq x)$)* [1 pt]

(b) Compute the eligibility vector $\nabla_\theta \log \pi_\theta(a|s)$. [1 pt]

(c) What should Y_k be in Algorithm 1 below? Express the update step of the parameter θ at episode k (line 4 in Algorithm 1) using only the feature map ϕ, and the observed trajectory at that episode. *Note: you may assume that the policy gradient theorem also holds for continuous state and action spaces in our case.* [2 pts]

Algorithm 3: REINFORCE algorithm

1 $\theta_1 \leftarrow$ initial value **for** $k \geq 1$ **do**
2 | use the policy π_{θ_k} to generate the trajectory $(s_{k,1}, a_{k,1}, c_{k,1}, \ldots, s_{k,T}, a_{k,T}, c_{k,T})$;
3 | $\theta_{k+1} \leftarrow \theta_k + \alpha_k\, Y_k$;
4 **end**

(d) Explain what may cause your algorithm to have a high variance and propose an idea that solves this issue. Rewrite the update step of Algorithm 1 if it needs to be changed. [2 pts]

Vector-valued control actions. You deal now with a control problem similar to the one described earlier with the only exception that now the action space $\mathcal{A} \subseteq \mathbb{R}^d$. You still wish to use a policy gradient method. After consulting with some experts, you decided to parameterize your policy by a parameter $\theta \in \mathbb{R}^d$ as follows: At each step t, first you sample η_t from a multivariate Gaussian[3] distribution $\mathcal{N}(0, \Sigma(\theta, s))$, where the covariance matrix is defined as follows:

$$\Sigma(\theta, s) = \sum_{i=1}^{n} \omega_i(\theta)\Sigma_i(s) \quad \text{with} \quad \omega_i(\theta) = \frac{e^{\theta_i}}{\sum_{i=1}^{d} e^{\theta_i}}.$$

For each $i \in \{1, \ldots, d\}$, we assume that the mapping $s \mapsto \Sigma_i(s)$ is known to you, and that $\Sigma_i(s)$ is positive definite[4]. Then, you choose action a_t according to

$$a_t = M\phi(s_t) + \eta_t$$

[2] **Gaussian distribution.** A random variable is said to be distributed according to the gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ if its density function is $g : t \mapsto \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{(t-\mu)^2}{2\sigma^2}\right)$. Furthermore, we have $\mathbb{P}(X \leq x) = \int_{t\leq x} g(t)dt$.

[3] **Multivariate Gaussian Distribution.** A multivariate gaussian distribution, denoted by $\mathcal{N}(\mu, \Sigma)$, is described by its mean μ and covariance matrix Σ which must be symmetric positive definite. Its density function is given by $g : v \mapsto \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mu - v)^\top \Sigma^{-1}(\mu - v)\right)$.

[4] **Positive definiteness.** We say that a symmetric matrix Σ is positive definite if and only if for all $x \in \mathbb{R}^d$, $x^\top \Sigma x > 0$.

where M is a matrix in $\mathbb{R}^{d\times d}$ that we assume to be known for simplicity. This action selection process ensures that $\pi_\theta(s|a)$ corresponds to the density function of the multivariate gaussian distribution $\mathcal{N}(M\phi(s), \Sigma(\theta, s))$.

(e) Verify that indeed $\Sigma(s, \theta)$ is positive definite. [1 pt]

(f) Compute the eligibility vector $\nabla_\theta \log(\pi_\theta(a|s))$. Then write the update step of the REINFORCE algorithm *Hint: to help you with your derivations you may use Jacobi's formula*

$$\frac{\partial}{\partial\theta_i}\det(G(\theta)) = \det(G(\theta))\ \mathrm{tr}\left(G(\theta)^{-1}\frac{\partial}{\partial\theta_i}G(\theta)\right),$$

where $\mathrm{tr}(\cdot)$ *denotes the trace and*

$$\frac{\partial}{\partial\theta_i}(G(\theta))^{-1} = G(\theta)^{-1}\left(\frac{\partial}{\partial\theta_i}G(\theta)\right)G(\theta)^{-1}.$$

where $G : \mathbb{R}^n \to \mathbb{R}^{d\times d}$. [3 pts]

Solution

1. Let us verify that $\pi_\theta(a|s)$ corresponds to the density of the gaussian distribution $\mathcal{N}(\theta^\top\phi(s), \sigma^2)$.

$$\begin{aligned}
F_{a_t|s_t=s}(x) &= \mathbb{P}\left(a_t \leq x | s_t = s\right) \\
&= \mathbb{P}\left(\theta^\top\phi(s) + \varepsilon_t \leq x | s_t = s\right) \\
&\overset{(a)}{=} \mathbb{P}\left(\varepsilon_t \leq x - \theta^\top\phi(s)\right) \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x-\theta^\top\phi(s)} \exp\left(-\frac{t^2}{2\sigma^2}\right) dt \\
&\overset{(b)}{=} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{(u-\theta^\top\phi(s))^2}{2\sigma^2}\right) dt
\end{aligned}$$

where in (a), we used the fact that ε_t is independent of s_t, and in (b) we used the change of variable $u = t + \theta^\top\phi(s)$. Thus,

$$F_{a_t|s_t=s}(x) = F_Z(x)$$

where Z is distributed according to the Gaussian distribution $\mathcal{N}(\theta^\top\phi(s), \sigma^2)$. Finally, from the hint, since the cumulative distribution uniquely determines the probability distribution of a given ranodm variable, we conclude $a_t|s_t = s$ and Z have the same probability distribution. Therefore, $\pi(a|s)$ is indeed the density of $\mathcal{N}(\theta^\top\phi(s), \sigma^2)$.

2. Straithforward computations should give

$$\nabla_\theta \log(\pi_\theta(a|s)) = \nabla_\theta\left(-\frac{(\theta^\top\phi(s)-a)^2}{2\sigma^2}\right) = \frac{1}{\sigma^2}(\theta^\top\phi(s)-a)\phi(s) \in \mathbb{R}^d$$

3. We wish to maximize

$$J(\theta) = \mathbb{E}^{\pi_\theta}\left[-\sum_{t=1}^{T} c_t(s_t, a_t)\middle| s_1 = s_{init}\right]$$

Note that we have introduced a negative sign. The policy gradient theorem gives us

$$\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}\left[-\frac{1}{\sigma^2}\left(\sum_{t=1}^{T}(\theta_k^\top\phi(s_{k,t}) - a_{k,t})\phi(s_{k,t})\right)\left(\sum_{t=1}^{T} c_t(s_{k,t}, a_{k,t})\right)\right] \\
&= \mathbb{E}\left[-\frac{1}{\sigma^2}\left(\sum_{t=1}^{T}\eta_{k,t}\phi(s_{k,t})\right)\left(\sum_{t=1}^{T} c_t(s_{k,t}, a_{k,t})\right)\right]
\end{aligned}$$

Therefore, we choose

$$\begin{aligned}
Y_k &= -\frac{1}{\sigma^2}\left(\sum_{t=1}^{T}(\theta_k^\top\phi(s_{k,t}) - a_{k,t})\phi(s_{k,t})\right)\left(\sum_{t=1}^{T} c_t(s_{k,t}, a_{k,t})\right) \\
&= -\frac{1}{\sigma^2}\left(\sum_{t=1}^{T}\eta_{k,t}\phi(s_{k,t})\right)\left(\sum_{t=1}^{T} c_t(s_{k,t}, a_{k,t})\right)
\end{aligned}$$

In fact, as you may observe, Y_k is not even dependent on θ_k. Both equalities in the above are acceptable solutions.

4. One idea to reduce the variance is to use the following result (See exercise session 5 sheet)

$$\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}\left[-\sum_{t=1}^{T}\left((\theta_k^\top\phi(s_{k,t}) - a_{k,t})\phi(s_t)\sum_{u=t+1}^{T} c_k(s_{k,t}, a_{k,t})\right)\right] \\
&= \mathbb{E}\left[-\sum_{t=1}^{T}\left(\eta_{k,t}\phi(s_t)\sum_{u=t+1}^{T} c_k(s_{k,t}, a_{k,t})\right)\right]
\end{aligned}$$

We then use

$$\begin{aligned} Y_k &= -\sum_{t=1}^{T}\left((\theta_k^\top\phi(s_{k,t}) - a_{k,t})\phi(s_t)\sum_{u=t+1}^{T} c_k(s_{k,t}, a_{k,t})\right) \\ &= -\sum_{t=1}^{T}\left(\eta_{k,t}\phi(s_t)\sum_{u=t+1}^{T} c_k(s_{k,t}, a_{k,t})\right) \end{aligned}$$

5. Clearly $\Sigma(\theta, s) \succ 0$ because of the choice of ω. We conclude that the parameterization of $\pi_\theta(a|s)$ is a valid one since the density is well defined.

6. Let us compute $\nabla_\theta \log(\pi_\theta(a|s))$

$$\begin{aligned} \nabla_\theta \log(\pi_\theta(a|s)) &= \nabla_\theta\left(-\frac{1}{2}\log\det(\Sigma(\theta, s))\right) - \nabla_\theta\left(\frac{1}{2}(a - M\phi(s))^\top\Sigma(\theta, s)^{-1}(a - M\phi(s))\right) \\ &= -\frac{1}{2}\nabla_\theta \log\det(\Sigma(\theta, s)) - \frac{1}{2}(a - M\phi(s))^\top\nabla_\theta\left(\Sigma(\theta, s)^{-1}\right)(a - M\phi(s)) \end{aligned}$$

Now we compute $\nabla_\theta \log\det(\Sigma(\theta, s))$. We have

$$\begin{aligned} \frac{\partial}{\partial\theta_i}\log\det(\Sigma(\theta, s)) &= \frac{1}{\det(\Sigma(\theta, s))}\frac{\partial}{\partial\theta_i}\det(\Sigma(\theta, s)) \\ &= \operatorname{tr}\left(\Sigma(\theta, s)^{-1}\frac{\partial}{\partial\theta_i}\Sigma(\theta, s)\right) \\ &= \operatorname{tr}\left(\Sigma(\theta, s)\left(\sum_{j=1}^{d}\frac{\partial}{\partial\theta_i}\omega_j(\theta)\Sigma_j(s)\right)\right) \\ &= \omega_i(\theta)\operatorname{tr}\left(\Sigma(\theta, s)\left(\Sigma_i(s) - \Sigma(\theta, s)\right)\right) \end{aligned}$$

where we used

$$\frac{\partial}{\partial\theta_i}\omega_j(\theta) = \begin{cases} \omega_i(\theta)(1 - \omega_i(\theta)) & \text{if } i = j \\ -\omega_i(\theta)\omega_j(\theta) & \text{otherwise} \end{cases}$$

We then compute $\nabla_\theta\left(\Sigma(\theta, s)^{-1}\right)$

$$\begin{aligned} \frac{\partial}{\partial\theta_i}\Sigma(\theta, s)^{-1} &= \Sigma(\theta, s)^{-1}\frac{\partial}{\partial\theta_i}\Sigma(\theta, s)\Sigma(\theta, s)^{-1} \\ &= \omega_i(\theta)\Sigma(\theta, s)^{-1}\left(\Sigma_i(s) - \Sigma(\theta, s)\right)\Sigma(\theta, s)^{-1} \end{aligned}$$