EL2805 Reinforcement Learning

Exam – January 2020

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

Exam (tentamen), January 11, 2019, kl 9.00 - 14.00

Aids. (possibly annotated) slides of the lectures (**nothing else**).

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, each worth 10 pts.

Grading.

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

Responsible.
Yassir Jedra 0738865235
Alexandre Proutiere 087906351
Damianos Tranos 0732797826
Ingvar Max Ziemann 0737309355

Results. Posted no later than January 25, 2019

Good luck!

Problem 1

(a) A random Fibonacci sequence $\{X_n, n \geq 1\}$ is defined as follows

$$\forall n \geq 2, \qquad X_{n+1} = \begin{cases} X_n + X_{n-1} & \text{w.p.} \quad 1/2, \\ X_n - X_{n-1} & \text{w.p.} \quad 1/2, \end{cases}$$

where $X_1 = X_2 = 1$. Is $\{X_n, n \geq 1\}$ a Markov chain? If not, construct one from it. [2 pts]

(b) Describe two variance reduction ideas for policy gradient methods. [1 pt]

(c) Assume you are using SARSA with an ε-greedy policy where ε is fixed. Does the algorithm converge to the true Q-function? Is there a way to ensure convergence to the true Q-function? [1 pt]

(d) Why do we need to update the target network in DQN rarely? [1 pt]

(e) Provide two RL problems where you would need to use function approximation. Motivate your choice. [2 pts]

(f) Provide the upper bound of the minimum expected squared error up to iteration k when using Robbins-Monro's stochastic approximation algorithm. Use this bound to motivate classical choices of step-sizes. [1 pt]

(g) Given an MDP with geometrically distributed time-horizon and such that $\mathbb{E}(T) = 20$. Which type (finite horizon, discounted infinite horizon) of MDP does it correspond to? Motivate your answer, and compute the discount factor if any. [1 pt]

(h) Is Q-learning a synchronous or asynchronous stochastic approximation algorithm? Motivate your answer.

[1 pt]

Solution

(a) No, $\{X_n, n \geq 1\}$ is not a Markov chain because for all $n \geq 3$, X_n depends on X_{n-1} and X_{n-2}. Define for all $n \geq 2$ the vector $Z_n = \begin{bmatrix} X_n & X_{n-1} \end{bmatrix}^\top$. Note that

$$Z_n = \begin{cases} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} Z_{n-1} & \text{w.p.} \quad 1/2, \\ \\ \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} Z_{n-1} & \text{w.p.} \quad 1/2. \end{cases}$$

$\{Z_n, n \geq 2\}$ is a Markov chain.

(b) Here are three ideas: (i) the baseline trick, (ii) generating i.i.d trajectories, or (iii) using the fact that $\mathbb{E}_{\pi_\theta}[\nabla \log \pi_\theta(s_t, a_t) r_u] = 0$ when $u < t$ (See slides of lecture 6)

(c) No, the algorithm will not converge to the true Q-function. One can ensure convergence by letting ε tend to 0.

(d) In the update of the Q-network, the target needs to be fixed for the algorithm to be able to converge, otherwise the target will be non-stationary making it hard to track.

(e) Here are few examples: (i) a board game such as GO, and the reason for choosing function approximation is because the state and action spaces are too large. (ii) a control problem such us the cartpole problem in lab 2, and the reason for choosing function approximation here is because you have a continuous state space.

(f) The upper bound is:

$$e^k_{min} = \frac{\|x^{(0)} - x^\star\|_2^2 + G^2 \sum_{i=0}^k \alpha_i^2}{\beta \sum_{i=0}^k \alpha_i}$$

We can see that for this error to converge to 0 as $k \to \infty$, we must have $\sum_{i=0}^\infty \alpha_i = \infty$ and $\sum_{i=0}^\infty \alpha_i^2 < \infty$. This is satisfied by the choice $\alpha_i = \frac{1}{i+1}$.

(g) An MDP with a geometrically distributed random time horizon can be interpreted as having an infinite horizon with discount factor $\lambda = \frac{\mathbb{E}(T)-1}{\mathbb{E}(T)} = 0.95$. The correct objective is to find a policy maximizing:

$$\mathbb{E}\left[\sum_{t=1}^\infty \lambda^{t-1} r_t(s_t^\pi, a_t^\pi)\right].$$

(h) Q-Learning is asynchronous, because at each iteration you are only updating the value of the Q function at a single state action pair (s, a).

Problem 2

The empirical mean. Let $X_1, X_2, \ldots$ be a sequence of i.i.d. binary random variables, with mean $1/\sqrt{2}$. Consider a sequence of random variables $Z_1, Z_2, \ldots$ such that for all $t \geq 1$

$$Z_t = \frac{X_1 + \cdots + X_t}{t}$$

(a) Show that $\{Z_t, t \geq 1\}$ is a Markov chain. Is it time homogeneous? What is the state space? [2 pts]

(b) Provide a communication class whose states are transient. [1 pt]

(c) You can sequentially observe $Z_1, \ldots, Z_T$ for some fixed horizon T. More precisely, in step t, you observe Z_t. After observing Z_t, you may decide to 'stop' or to 'continue' towards the next step. Your objective is to find a stopping rule minimizing the average distance between the value of Z_t when you stop and $p = 1/\sqrt{2}$. Model this as an MDP (do not try to solve the MDP). [2 pts]

A motorcycle problem. You are a motorcycle racer. Your objective is to complete T laps as fast as possible. At the beginning of each lap, you decide to take risks and race fast, or to be careful and race slowly. You further decide to change your tyres or continue with the same tyres. Completing a lap when racing fast (resp. slow) takes $C + N_f$ (resp. $C + N_s$) seconds, where $C \geq 20$ is a constant and N_f (resp. N_s) is a random variable uniformly distributed over the set $\{-1, \ldots, -9\}$ (resp. $\{-1, -2, -3\}$). The time required to change tyres is random and uniformly distributed over $\{5, \ldots, 10\}$.

The state of tyres at a beginning of a lap is described by a number X. $X = L$ when the tyres are newly changed where L is a positive integer. By the end of each lap, the tyres have deteriorated depending on how fast you race. When driving fast one lap, the state of the tyres decreases by an amount of 2, and when driving slow, it decreases by an amount of 1. When the state reaches 0, you must change the tyres. But you may change tyres at the beginning of each lap.

(d) Model the motorcycle problem as an MDP if that is at all possible. Specify the horizon, the state space, action space, transition probabilities and rewards. *Do not solve the MDP.* [5 pts]

Solution

Empirical mean.

(a)

- The sequence $\{Z_t, t \geq 1\}$ is a markov chain because
$$Z_{t+1} = \frac{t}{t+1} Z_t + \frac{1}{t+1} X_{t+1},$$
and X_{t+1} is independent of $Z_1, \dots, Z_t$.
- The markov chain is not time-homogeneous because
$$\mathbb{P}\left(Z_2 = \frac{1}{2} | Z_1 = 1\right) = 1 - \frac{1}{\sqrt{2}}, \qquad \text{and} \qquad \mathbb{P}\left(Z_3 = \frac{1}{2} | Z_2 = 1\right) = 0.$$
- Note that for all $t \geq 1$, Z_t is a rational number, since we have $X_1 + \dots + X_t \in \mathbb{N}$. Hence, the state space is $\mathcal{S} = [0,1] \cap \mathbb{Q}$.

(b) An exemple of communication classes that are transient are $\{0\}$ and $\{1\}$. Note that $Z_t = 1$ (resp. $Z_t = 0$) if and only if $X_1 = \dots = X_t = 1$ (resp. $X_1 = \dots = X_t = 0$). Furthermore, $\mathbb{E}[X_t] = 1/\sqrt{2}$ thus there must exist a time τ at which $X_\tau \neq X_{\tau+1}$, once this happens, the communication classes $\{0\}$ and $\{1\}$ will never be visited.

(c) Let us model the problem as an MDP.

- From the problem statement it is clear that we have a finite horizon T. Decisions are taken at steps $\{1, \dots, T-1\}$.
- At each step t, the state in which we are in, is either described by the pair (Z_t, t) or the value ∞. The latter correponds to the case where we have stopped observing the realizations. The state space is $\mathcal{S} = [0,1] \cap \mathbb{Q} \times \{1, \dots, T\} \cup \{\infty\}$.
- We can either continue (C) or stop (S). Thus, the action space is $\mathcal{A} = \{\mathrm{S}, \mathrm{C}\}$.
- First, let us describe the transitions where end up in the state ∞.
$$\begin{aligned} \mathbb{P}(s' = \infty | s = (c/d, t), a = \mathrm{S}) &= 1, \\ \mathbb{P}(s' = \infty | s = \infty, a = \mathrm{S}) &= 1, \\ \mathbb{P}(s' = \infty | s = \infty, a = \mathrm{C}) &= 1. \end{aligned}$$

  To describe the remaining transitions recall from (a) the recursive expression $Z_{t+1} = (t/t+1)Z_t + (1/t+1)X_t$.
$$\begin{aligned} \mathbb{P}\left(s' = \left(\frac{ta+b}{(t+1)b}, t+1\right) \middle| s = \left(\frac{a}{b}, t\right), a = C\right) &= p, \\ \mathbb{P}\left(s' = \left(\frac{ta}{(t+1)b}, t+1\right) \middle| s = \left(\frac{a}{b}, t\right), a = C\right) &= 1-p, \\ \mathbb{P}\left(s' = \infty \middle| s = \left(\frac{a}{b}, t\right), a = S\right) &= 1. \end{aligned}$$

- We seek to minimize the average distance between the value of Z_t when we stop and $p = 1/\sqrt{2}$. We may model our rewards as follows:

$$\begin{aligned}
r_t(\infty, a) &= 0, \quad \forall a \in \mathcal{A} && \text{(non terminal rewards)}\\
r_t((a/b, t), \mathrm{S}) &= -|(a/b) - p|, && \text{(non terminal rewards)}\\
r_t((a/b, t)), \mathrm{C}) &= 0, && \text{(non terminal rewards)}\\
r_T(\infty) &= 0, && \text{(terminal rewards)}\\
r_T((a/b, T))) &= -|(a/b) - p|. && \text{(terminal rewards)}
\end{aligned}$$

(d) Let us model the motorcycle problem as an MDP

- The problem is a finite horizon problem since we race for T laps. Note that decisions are made at the begining of each lap, thus at steps $\{0, \dots, T-1\}$.
- We define the state space as follows

$$\mathcal{S} = \{-1, \dots, L\} \times \{T(C-9), \dots, T(C+9)\}.$$

  - The state at stage t can be defined as a pair $s_t = (l_t, g_t) \in \mathcal{S}$, where l_t and g_t correspond respectively to the state of the tyres and your time, by the end of lap t. Initially, you may assume the timer to be set at 0, which means $g_0 = 0$.
  - The maximum (resp. minimum) number of seconds it takes to finish a lap is $C+9$ (resp. $C-9$).

- We define the action space as follows

$$\mathcal{A} = \{\mathrm{F}, \mathrm{S}\} \times \{\mathrm{C}, \mathrm{N}\}.$$

  An action taken at the start of a lap is a pair $a = (a_1, a_2) \in \mathcal{A}$, where a_1 correponds to whether you decide to race fast (F) or slow (S), and a_2 correponds to whether you further decide to change your tyres (C) or not (N).

- First, we define the transition probabilities where you are forced to change your tyres, that is when $s = (l, g)$ and $l \leq 0$. In this case, we have

$$\begin{aligned}
\mathbb{P}(s' = (L-1, g + C + \tau_s + \tau_c) | s = (l, g), a = (S, a_2)) &= \frac{1}{18},\\
\mathbb{P}(s' = (L-2, g + C + \tau_f + \tau_c) | s = (l, g), a = (F, a_2)) &= \frac{1}{54},
\end{aligned}$$

  for $a_2 \in \{\mathrm{C}, \mathrm{N}\}$, $\tau_f \in \{-1, \dots, -9\}$, $\tau_s \in \{-1, \dots, -3\}$, and $\tau_c \in \{5, \dots, 10\}$. Next, we define the transitions when you are not forced to change your tyres but you may decide to do so, that is when $s = (g, l)$ and $l > 0$. We have

$$\begin{aligned}
\mathbb{P}(s' = (L-1, g + C + \tau_s + \tau_c) | s = (l, g), a = (S, C)) &= \frac{1}{18},\\
\mathbb{P}(s' = (L-2, g + C + \tau_f + \tau_c) | s = (l, g), a = (F, C)) &= \frac{1}{54},\\
\mathbb{P}(s' = (l-1, g + C + \tau_s) | s = (l, g), a = (S, N)) &= \frac{1}{3},\\
\mathbb{P}(s' = (l-2, g + C + \tau_f) | s = (l, g), a = (F, N)) &= \frac{1}{9},
\end{aligned}$$

  for $\tau_f \in \{-1, \dots, -9\}$, $\tau_s \in \{-1, \dots, -3\}$, and $\tau_c \in \{5, \dots, 10\}$.

- The objective is to complete T laps as fast as possible, which corresponds to minimizing g_T. Thus, we model our rewards as follows

$$\begin{aligned}
r_t(s, a) &= 0, && \text{(non terminal rewards)}\\
r_T(s) &= -g. && \text{(terminal rewards)}
\end{aligned}$$

Problem 3

You are given a research paper to proofread. The paper contains M typos. M is a random variable with known distribution F over the set $\{0, 1, \dots, Z\}$ for some integer Z. Each time you read the paper, and each time you read over a word with a typo, you have a probability p to detect the typo. The cost of proofreading the paper is $c > 0$. For each detected typo, you receive a reward $r > 0$. Your objective is to sequentially decide to proofread the paper so as to maximize your expected reward (e.g. after proofreading once, you may decide to proofread again or to stop depending on the number of detected typos).

Known number of typos. Assume that M is known to you when you get the paper.

(a) Model the problem as an MDP. [2 pts]

(b) Write Bellman's equation. Identify the best 'one-step look-ahead' strategy. By definition, a one-step look-ahead strategy is as follows: in a given state, it decides to either stop or to proofread exactly one more time. [3 pts]

Unknown number of typos. Assume that M is unknown, but that its distribution F is known. Naturally, the number of detected typos each time you proofread the paper provides some information about M.

(c) Let N_1 denote the number of typos you detect during the first time you proofread the paper. Provide the expression of the posterior distribution of M given that $N_1 = n_1$. This distribution is denoted by $P^{(1)}(n_1, F)$ and is defined as:

$$\forall z \in \{0, 1, \dots, Z\}, \quad P^{(1)}(n_1, F)(z) := \mathbb{P}\left[M = z | N_1 = n_1\right].$$

Hint: If a urn is filled sequentially with N balls, and each ball is red with probability p and blue with probability $1 - p$, the probability that the urn contains exactly z red balls is given by the binomial distribution $\binom{N}{z} p^z (1 - p)^{N-z}$. [2 pts]

(d) In the remaining of the problem, we assume that you were able to recursively compute the following posterior distributions of M. If you have proofread the paper k times, and you have detected $N_1 = n_1, \dots, N_k = n_k$ typos in each round (e.g. n_k is the number of typos you have detected the k-th time you proofread), the posterior distribution of M is denoted by $P^{(k)}(n_1, \dots, n_k, F)$. By definition:

$$\forall z \in \{0, 1, \dots, Z\}, \quad P^{(k)}(n_1, \dots, n_k, F)(z) := \mathbb{P}\left[M = z | N_1 = n_1, \dots, N_k = n_k\right].$$

Model the problem as an MDP. [2 pts]

(e) Provide an optimal one-step look-ahead stopping rule. [1 pt]

Solution

a) We propose the following MDP:

- Time horizon: T very large.
- Action space: (C) proofread one more time, (S) stop.
- State: the state captures the information available to the decision maker when deciding to stop or to continue. The reward that we can collect in the future only depends on the number of remaining typos. Hence, since we know the number of typos M initially present in the paper, we can use the state (k,n) where k is the number of times you have proofread already, and n is the number of detected typos so far. The state space is $\{(k,n), k \in \{0,1,\dots,T\}, n \in \{0,1,\dots,M\}\}$. As usual we add to the state space, a state $'0'$ indicating that we stopped.
- Rewards: they depend on the state only and are given by $r(k,n) = -kc + nr$.
- Transition probabilities: $p('0'|(k,n),S) = 1$, and for all $n' \in \{n,\dots,M\}$,

$$p((k+1,n')|(k,n),C) = \binom{M-n}{n'-n} p^{n'-n}(1-p)^{M-n'}.$$

b) Bellman's equation is given by: for any (k,n),

$$V(k,n) = \max\{\underbrace{-kc+nr}_{Stop}, \underbrace{-(k+1)c + \sum_{n'=n}^{M} \binom{M-n}{n'-n} p^{n'-n}(1-p)^{M-n'} \left(n'r + V(k+1,n')\right)}_{Continue}\},$$

Now, for one-step look-ahead strategies, assume that we are in state (k,n). If we stop, we collect a reward $-kc+nr$, and we proofread one more time (and then stop) we collect a reward equal to:

$$-(k+1)c + nr + \mathbb{E}[X|k,n],$$

where $\mathbb{E}[X|k,n]$ denotes the average number of newly detected typos, given that you already proofread k times and detected n typos. Each remaining typo is detected with probability p, and hence:

$$\mathbb{E}[X|k,n] = p \times (M-n).$$

Finally, if we proofread one more time, we collect a reward equal to $-(k+1)c + (n + p(M-n))r$. We deduce that the optimal one-step look-ahead strategy consists in stopping if and only if:

$$p(M-n)r \leq c.$$

Note that this optimal strategy does not depend on k.

c) We use the definition:

$$\mathbb{P}\left[M = z | N_1 = n_1\right] = \frac{\mathbb{P}\left[(N_1 = n_1) \cap (M = z)\right]}{\mathbb{P}[N_1 = n_1]}.$$

We have:

$$\mathbb{P}[(N_1 = n_1) \cap (M = z)] = 1_{\{n_1 \leq z\}} F(z) \binom{z}{n_1} p^{n_1}(1-p)^{z-n_1},$$

$$\mathbb{P}[N_1 = n_1] = \sum_{m=n_1}^{Z} F(m) \binom{m}{n_1} p^{n_1}(1-p)^{m-n_1}.$$

Finally, we have:

$$P^{(1)}(n_1,F)(z) = \frac{1_{\{n_1 \leq z\}} F(z) \binom{z}{n_1} p^{n_1}(1-p)^{z-n_1}}{\sum_{m=n_1}^{Z} F(m) \binom{m}{n_1} p^{n_1}(1-p)^{m-n_1}}.$$

d) We propose the following MDP:

- Time horizon: T very large.
- Action space: (C) proofread one more time, (S) stop.
- State: the state captures the information available to the decision maker when deciding to stop or to continue. Here, if we have proofread k times, the numbers of detected typos in each round $n_1, \dots, n_k$ provide information about the total number of typos M. Observe that the posterior distribution $P^{(k)}(n_1, \dots, n_k, F)$ depends on the individual values of the n_k's (not only on their sum). This posterior also determines the number of remaining typos in the paper, and hence future rewards. As a consequence, we need to have the values $(n_1, \dots, n_k)$ in the state. The state space is:

$$\mathcal{S} = \{'0'\} \cup \{(0,0)\} \cup \left(\cup_{k=1}^{T} \{(k, n_1, \dots, n_k) : \sum_i n_i \leq Z\} \right).$$

  Note that $(0,0)$ is the initial state.
- Rewards: they depend on the state only and are given by $r(k, n_1, \dots, n_k) = -kc + \sum_{i=1}^{k} n_i r$.
- Transition probabilities: $p('0'|s, S) = 1$ for all $s \in \mathcal{S}$, and for all $s' = (k+1, n_1, \dots, n_k, n_{k+1})$,

$$p(s'|(k, q_1, \dots, q_k), C) = \left(\prod_{i=1}^{k} 1_{\{n_i = q_i\}} \right) \times \binom{M - \sum_{i=1}^{k} n_i}{n_{k+1}} p^{n_{k+1}} (1-p)^{M - \sum_{i=1}^{k+1} n_i}.$$

  All other transitions have probability zero to occur.

e) Consider a one-step look-ahead policy. You are in state $(k, n_1, \dots, n_k)$. From this, we know that the posterior distribution of the initial number M of typos is $P^{(k)}(n_1, \dots, n_k, F)$. We compute the average reward when we stop and when we proofread just one more time:

- If we stop, the collected reward is: $-kc + r \sum_{i=1}^{k} n_i$.
- If we proofread one more time and then stop, the collected reward becomes:

$$-(k+1)c + r \left(\sum_{i=1}^{k} n_i + \mathbb{E}[X|(k, n_1, \dots, n_k)] \right),$$

  where $\mathbb{E}[X|(k, n_1, \dots, n_k)]$ is the expected number of detected typos in the $(k+1)$-th round, given that in the k first rounds, $n_1, \dots, n_k$ typos had been detected. We have:

$$\begin{aligned}
\mathbb{E}[X|(k, n_1, \dots, n_k)] &= \sum_{z=0}^{Z} \sum_{x} x \mathbb{P}[X = x, M = z | (k, N_1 = n_1, \dots, N_k = n_k] \\
&= \sum_{z=\sum_{i=1}^{k} n_i}^{Z} P^{(k)}(n_1, \dots, n_k, F)(z) \sum_{x=0}^{z - \sum_{i=1}^{k} n_i} x \times \binom{z - \sum_{i=1}^{k} n_i}{x} p^x (1-p)^{z - \sum_{i=1}^{k} n_i - x} \\
&= \sum_{z=\sum_{i=1}^{k} n_i}^{Z} P^{(k)}(n_1, \dots, n_k, F)(z) p(z - \sum_{i=1}^{k} n_i).
\end{aligned}$$

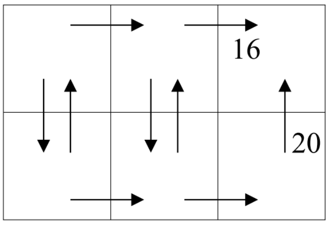
As a consequence, in state $(k, n_1, \dots, n_k)$, it is optimal to stop if and only if:

$$\sum_{z=\sum_{i=1}^{k} n_i}^{Z} P^{(k)}(n_1, \dots, n_k, F)(z) p(z - \sum_{i=1}^{k} n_i) r \leq c.$$

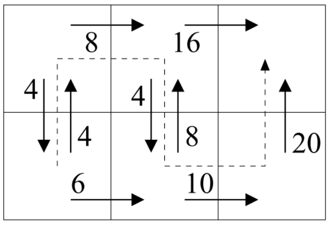
Problem 4

Consider the deterministic world shown in Figure (a). At every time step k, the agent is at a cell or state s_k and takes action a_k, observing reward r_k before transitioning to the next state s_{k+1}. Admissible moves are shown by the arrows, the numbers next to each arrow indicate the reward for performing the action. If there is no number, the reward is zero. The state shown in the cell in the upper right corner is terminal (no moves are done and no reward is collected after reaching this state).

An agent follows the path shown in Figure (b) by the dotted line (starting at the lower left cell, with the upper right cell as the goal). The agent does not know the environment, and seeks to learn the optimal policy using Q-learning or SARSA, with discount factor $\lambda = 0.9$ and learning rate $\alpha = 1$. The initial estimates of the Q values are shown in Figure (b).



(a)



(b)

a) Choose a suitable labeling for every state and action (e.g. $A, B, \dots F$ and a, b). Provide the observations of the agent along its path (e.g. $(s_k, a_k, r_k, s_{k+1}, \dots)$) and provide the initial Q-values in tabular form. [2 pts]

b) Provide the updated Q-values when the agent has reached the goal state, if the agent uses Q-learning. [2 pts]

c) What is the greedy policy with respect to these updated Q-values? Comment on whether it is optimal. [2 pts]

d) Provide the updated Q-values when the agent has reached the goal state, if the agent uses SARSA. [2 pts]

e) What is the greedy policy with respect to these updated Q-values? Comment on whether it is optimal and compare it to the greedy policy you found in c). [2 pts]

Solution

a) We choose to label states from left to right, i.e. $\begin{bmatrix} A & B & C \\ D & E & F \end{bmatrix}$

With the exception of the states C and F, there are two actions per state. The actions are either horizontal (left, right) or vertical (up, down) and there is at most one horizontal and one vertical action at each state. We thus choose to label every horizontal action as a and every vertical action as b.

The trajectory and observed rewards of the agent can then be written as the sequence:

$$(D, b, 0, A, a, 0, B, b, 0, E, a, 0, F, b, 20, C)$$

.

The initial Q values can be written in table form as:

$$Q^{(0)} = \begin{array}{c|cc} & a & b \\ \hline A & 8 & 4 \\ B & 16 & 4 \\ C & & \\ D & 6 & 4 \\ E & 10 & 8 \\ F & & 20 \end{array}.$$

Note that the table has been left empty at each state where the action is not admissible.

b) Recall that the Q-Learning update at every iteration k can be written as:

$$Q^{(k+1)}(s_k, a_k) = (1 - \alpha)Q^{(k)}(s_k, a_k) + \alpha \left(r_k + \lambda \max_x Q^{(k)}(s_{k+1}, x) \right)$$

Because the of the learning rate $\alpha = 1$ and the fact that for all but the last iteration, the obtained reward is 0, the computations simplify to:

$$\begin{aligned} Q^{(1)}(D, b) &= \lambda \max_x Q^{(0)}(A, x) = \lambda \cdot Q^{(0)}(A, a) = 0.9 \cdot 8 = 7.2 \\ Q^{(2)}(A, a) &= \lambda \max_x Q^{(1)}(B, x) = \lambda \cdot Q^{(1)}(B, a) = 0.9 \cdot 16 = 14.4 \\ Q^{(3)}(B, b) &= \lambda \max_x Q^{(2)}(E, x) = \lambda \cdot Q^{(2)}(E, a) = 0.9 \cdot 10 = 9 \\ Q^{(4)}(E, a) &= \lambda \max_x Q^{(3)}(F, x) = \lambda \cdot Q^{(3)}(F, b) = 0.9 \cdot 20 = 18 \end{aligned}$$

For the last iteration, since $s_5 = C$ (the goal state), the update will be just the observed reward:

$$Q^{(5)}(F, b) = r_4 = 20$$

Thus, the updated Q-values when the agent has reached the goal are:

$$Q^{(5)} = \begin{array}{c|cc} & a & b \\ \hline A & 14.4 & 4 \\ B & 16 & 9 \\ C & & \\ D & 6 & 7.2 \\ E & 18 & 8 \\ F & & 20 \end{array}.$$

c) The greedy policy is $\pi(A) = a$, $\pi(B) = a$, $\pi(D) = b$, $\pi(E) = a$, $\pi(F) = b$. The policy is not optimal because starting at state D, the agent obtains a cumulative reward of 16 in three steps, whereas it could have obtained a reward of 20 in the same number of steps, if $\pi(D) = a$.

d) Recall that the SARSA update at every iteration k can be written as:

$$Q^{(k+1)}(s_k, a_k) = (1 - \alpha)Q^{(k)}(s_k, a_k) + \alpha \left(r_k + \lambda Q^{(k)}(s_{k+1}, a_{k+1})\right)$$

The updates follow similarly to those in question b), leading to the updated Q-values:

$$Q^{(5)} = \begin{array}{cc} & \begin{array}{cc} a & b \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array} & \begin{bmatrix} 3.6 & 4 \\ 16 & 9 \\ & \\ 6 & 7.2 \\ 18 & 8 \\ & 20 \end{bmatrix} \end{array}.$$

e) The greedy policy is $\pi(A) = b$, $\pi(B) = a$, $\pi(D) = b$, $\pi(E) = a$, $\pi(F) = b$. The policy is not optimal and it is worse than the policy of c) as it leads to zero reward; Starting at state D, the agent repeatedly transitions between states D and A.

Problem 5

Consider a reinforcement learning problem where our action space is continuous, $\mathcal{A} = \mathbb{R}^d, d \in \mathbb{N}$. One way to parametrize the policy is by a Gaussian family:

$$\pi(s,a) = \frac{\exp\left(-\frac{1}{2}(a-\mu)^\top \Sigma^{-1}(a-\mu)\right)}{\sqrt{(2\pi)^d|\Sigma|}}$$

where for instance the mean is $\mu = \mu(s)$, and the (symmetric positive definite) covariance matrix $\Sigma = \Sigma(s)$ could be used to implement dependence of the policy on the state. However, using the mean and convariance, $\mu \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d\times d}$, is not the only possible parametrization of the Gaussian family. We outline another approach below.

a) Show that $\pi(s,a)$ above may be written on the form

$$\pi(s,a) = \exp\Big(\eta \cdot T(a) - Z(\eta)\Big)$$

where $\eta = \eta(\mu, \Sigma) = \begin{bmatrix}\eta_1 \\ \eta_2\end{bmatrix} = \begin{bmatrix}\Sigma^{-1}\mu \\ -\frac{1}{2}\Sigma^{-1}\end{bmatrix}$, $Z(\eta) = -\frac{1}{4}\eta_1^\top \eta_2^{-1}\eta_1 - \frac{1}{2}\log|-2\eta_2| \qquad \underbrace{+\frac{d}{2}\log 2\pi}_{\text{typo corrected here, was } -k/2\log 2\pi}$

and $T(a) = \begin{bmatrix} a \\ aa^\top\end{bmatrix}$.[1]

Hint: For a matrix Q and a vector y one has $Q \cdot yy^\top := \textbf{trace}[Qyy^\top] = y^\top Qy$. [2pts]

b) What are the admissible values of η_2 for $\pi(s,a)$ to be a well-defined probability density? [2 pts]

Suppose now that you have scalar features $g(s)$ and decide to parametrize the first natural parameter η_1 by these. That is, let $\eta_1 = \theta g(s)$ so that you obtain a policy $\pi(s,a) = \pi_\theta(s,a)$.

c) Express $\pi_\theta(s,a)$ in terms of θ and $g(s)$ and compute the score function $\nabla_\theta \log \pi_\theta(s,a)$. [2 pts]

d) Prove that $\mathbb{E}_{\pi_\theta}\nabla_\theta \log \pi_\theta(s_t, a_t) = 0$. *Hint: Recall the proof for the discrete version of this result and note that you may differentiate under the integral sign.* [2 pts]

Suppose now that $d = \dim\theta = 1$, and that you wish to solve a finite time horizon problem. Imagine that you seek to use the parametrization we derived above in a policy gradient algorithm. However, you are rather worried about the variance of your updates. Recall that in terms of a trajectory τ and observed rewards $R(\tau)$, the optimal constant variance reduction term is

$$b = \frac{\mathbb{E}_{\pi_\theta}[(\nabla_\theta \log \pi_\theta(\tau))^2 R(\tau)]}{\mathbb{E}_{\pi_\theta}[(\nabla_\theta \log \pi_\theta(\tau))^2]}.$$

The denominator $\mathbb{E}_{\pi_\theta}[(\nabla_\theta \log \pi_\theta(\tau))^2]$ is referred to as the *Fisher information* of the model π_θ.

e) Compute the Fisher information for the Gaussian $\pi_\theta(s,a)$ above.[2] [2 pts]

[1] This verifies that the multivariate normal family is a so-called *exponential family* which are frequently used in statistics and machine learning. $T(a)$ is known as the sufficient statistic, $Z(\eta)$ the log-partition function and η is referred to as the natural parameter.

[2] For the interested student, the Fisher information measures the sensivity of the model to changes in the parameter θ. In a statistical setting, it also allows one to quantify how much information a random sample contains.

Problem 5 - Solution

a) To find Z, use the identity

$$\sqrt{(2\pi)^d|\Sigma|} = \exp\left(\frac{1}{2}\log\,(2\pi)^d|\Sigma|\right)$$

to bring everything inside the exponential. Then expand the square and identify

$$\begin{aligned}
(\eta_1,\eta_2) &= (\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1})\\
Z(\eta) &= -\frac{1}{4}\eta_1^\top\eta_2^{-1}\eta_1 - \frac{1}{2}\log|-2\eta_2| + \frac{d}{2}\log 2\pi\\
T(a) &= \begin{bmatrix} a \\ aa^\top \end{bmatrix}
\end{aligned}$$

where you need to use the hint to see that

$$a^\top\eta_2 a = \mathbf{trace}(\eta_2 aa^\top) = \eta_2\cdot aa^\top$$

where $aa^\top$ is the second coordinate of the sufficient statistic $T(a)$. In detail:

$$\begin{aligned}
\pi(s,a) &= \frac{\exp\left(-\frac{1}{2}(a-\mu)^{\mathrm{T}}\Sigma^{-1}(a-\mu)\right)}{\sqrt{(2\pi)^d|\Sigma|}}\\
&= \exp\left(-\frac{1}{2}\log\,(2\pi)^d|\Sigma|\right)\exp\left(-\frac{1}{2}(a-\mu)^\top\Sigma^{-1}(a-\mu)\right)\\
&= \exp\left(-\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(a-\mu)^\top\Sigma^{-1}(a-\mu)\right)\\
&= \exp\left(-\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma| - \frac{1}{2}\left[a^\top\underbrace{\Sigma^{-1}}_{=-2\eta_2}a + \underbrace{\mu^\top\Sigma^{-1}\mu}_{=(\Sigma^{-1}\mu)^\top\Sigma\Sigma^{-1}\mu} - 2\underbrace{\mu^\top\Sigma^{-1}}_{=\eta_1^\top}a\right]\right)\\
&= \exp\left(-\frac{d}{2}\log 2\pi - \frac{1}{2}\underbrace{\log|-(2\eta_2)^{-1}|}_{=-\log|-2\eta_2|} + \eta_2\cdot aa^\top + \frac{1}{4}\eta_1^\top\eta_2^{-1}\eta_1 + \eta_1\cdot a\right)\\
&= \exp\left(\underbrace{\eta_1\cdot a + \eta_2\cdot aa^\top}_{=\eta\cdot T(a)}\underbrace{+\frac{1}{4}\eta_1^\top\eta_2^{-1}\eta_1 + \frac{1}{2}\log|-2\eta_2| - \frac{d}{2}\log 2\pi}_{=-Z(\eta)}\right)\\
&= \exp(\eta\cdot T(a) - Z(\eta)).
\end{aligned}$$

b) Note that η_2 is ill-defined unless Σ has full rank. The Gaussian needs to have positive (semi-) definite covariance matrix (since all covariance matrices are PSD). This translates into the condition $-\eta_2 \succ 0$. NB: there are equivalent conditions such as $(-2\eta_2)^{-1} \succ 0$.

c) One writes

$$\begin{aligned}
\pi_\theta(s,a) &= \exp\Big(\eta(\mu,\Sigma)\cdot T(a) - Z(\eta)\Big)\\
&= \exp\Big(\begin{bmatrix}\eta_1\\ \eta_2\end{bmatrix}\cdot\begin{bmatrix} a\\ aa^\top\end{bmatrix} + \frac{1}{4}\eta_1^\top\eta_2^{-1}\eta_1 + \frac{1}{2}\log|-2\eta_2| - \frac{d}{2}\log 2\pi\Big)\\
&= \exp\Big(\begin{bmatrix}\theta g(s)\\ \eta_2\end{bmatrix}\cdot\begin{bmatrix} a\\ aa^\top\end{bmatrix} + \frac{1}{4}[\theta g(s)]^\top\eta_2^{-1}\theta g(s) + \frac{1}{2}\log|-2\eta_2| - \frac{d}{2}\log 2\pi\Big).
\end{aligned}$$

Computing the gradient of the logarithm therefore gives

$$\begin{aligned}\nabla_\theta \log \pi_\theta(s,a) &= g(s)a + \frac{1}{2}g^2(s)\eta_2^{-1}\theta \\ &= g(s)\left(a + \frac{1}{2}\eta_2^{-1}\eta_1\right)\end{aligned}$$

d) Observe that we actually need to know nothing about the Gaussian except for the fact that its density is regular enough for us to differentiate under the integral. It is easiest to use iterated expectation directly to see that

$$\begin{aligned}\mathbb{E}_{\pi_\theta}\nabla_\theta \log \pi_\theta(s_t,a_t) &= \mathbb{E}_{\pi_\theta}\mathbb{E}[\nabla_\theta \log \pi_\theta(s_t,a_t)|s_0,s_1,a_1,\ldots,a_{t-1},s_t] \\ &= \mathbb{E}_{\pi_\theta}\int_{\mathcal{A}} \nabla_\theta \log \pi_\theta(s_t,a)\pi_\theta(s_t,a)da \\ &= \mathbb{E}_{\pi_\theta}\int_{\mathcal{A}} \frac{1}{\pi_\theta(s_t,a)}\nabla_\theta \pi_\theta(s_t,a)\pi_\theta(s_t,a)da \\ &= \mathbb{E}_{\pi_\theta}\int_{\mathcal{A}} \nabla_\theta \pi_\theta(s_t,a)da \\ &= \mathbb{E}_{\pi_\theta}\nabla_\theta\int_{\mathcal{A}} \pi_\theta(s_t,a)da \\ &= \mathbb{E}_{\pi_\theta}\nabla_\theta 1 \\ &= 0.\end{aligned}$$

e) Direct computation is possible, but the slicker way is to first notice that

$$\mathbb{E}_{\pi_\theta}[(\nabla_\theta \log \pi_\theta(\tau))^2] = -\mathbb{E}_{\pi_\theta}[\nabla_\theta^2 \log \pi_\theta(\tau)].$$

To see this, observe the scalar identity[3] ($\nabla_\theta = \frac{\partial}{\partial\theta}$)

$$\begin{aligned}\nabla_\theta^2 \log \pi_\theta(s_t,a_t) &= \frac{\nabla_\theta^2 \pi_\theta(s_t,a_t)}{\pi_\theta(s_t,a_t)} - \left(\frac{\nabla_\theta \pi_\theta(s_t,a_t)}{\pi_\theta(s_t,a_t)}\right)^2 \\ &= \frac{\nabla_\theta^2 \pi_\theta(s_t,a_t)}{\pi_\theta(s_t,a_t)} - (\nabla_\theta \log \pi_\theta(s_t,a_t))^2\end{aligned}$$

and note that the expectation of the first term on the right is zero since by exchanging integral and derivative one sees that it is equal to the Jacobian of the expectation of the score – but the expectation of the score is already zero. Computing this second term by using the result in c) gives

$$-\mathbb{E}_{\pi_\theta}[\nabla_\theta^2 \log \pi_\theta(\tau)] = -\nabla_\theta\left(g(s)a + \frac{1}{2}g^2(s)\eta_2^{-1}\theta\right) = -\frac{1}{2}g^2(s)\eta_2^{-1}.$$

Therefore

$$\mathbb{E}_{\pi_\theta}[(\nabla_\theta \log \pi_\theta(\tau))^2] = -\mathbb{E}_{\pi_\theta}[\nabla_\theta^2 \log \pi_\theta(\tau)] = -\frac{1}{2}\eta_2^{-1}\mathbb{E}_{\pi_\theta}g^2(s)$$

and further progress is impossible without knowledge of g.

[3] although the conclusion remains valid in $\mathbb{R}^d$