EL2805 Reinforcement Learning

Re-exam – April 17, 2020

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

Aids. Slides of the lectures (**not exercises**), blackboard notes, 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 April 30, 2020

Good luck!

Problem 1

Provide short answers to the following questions **and** a short motivation (not more than ca 5 sentences per question).

a) Name two algorithms to solve Bellman's equations in infinite-horizon discounted MDPs. [1 pt]

b) Write Bellman's equation for finite-time horizon MDPs. Explain why solving this equation requires $\Theta(S^2AT)$ floating operations. [2 pts]

c) Suppose that you are using the ϵ-greedy policy in the Q-learning and SARSA algorithms. Are the two algorithms converging to the same Q-table? [1 pt]

d) Suppose you are trying to minimize a convex function f. To this aim, we have access to noisy function evaluations (if we pick x, we observe a noisy version of the gradient of the function at x, namely $\nabla f(x) + \eta$ where η is a zero-mean random variable). Does the stochastic gradient descent algorithm always decrease the function value in each iterate? [1 pt]

e) What does "TD" learning mean? [1 pt]

f) Is the basic Q-learning algorithm based on Robbins-Monroe algorithm or the stochastic gradient algorithm? [1 pt]

g) Why do we use a target network in deep Q-learning algorithms? [1 pt]

h) Is Q-learning a model-free or a model-based algorithm? [1 pt]

i) Provide a motivation for RL algorithms with function approximation. [1 pt]

Solutions to Problem 1

a) Policy and value iteration algorithms.

b) See the slides of part 2 of the course.

c) SARSA is an on-policy algorithm, and $\lim_{t\to\infty} Q_{SARSA}(t)$ will converge to the Q function of π_ϵ, the optimal policy among the policies exploring randomly with probability ϵ. On the other hand, Q-learning is off-policy and only uses ϵ-greedy to gather samples and will converge to the optimal Q-function.

d) No, the stochastic gradient is only a descent direction on average.

e) Time Difference.

f) Robbins Monroe.

g) The target is fixed so as to let the parameter update converge before changing the target.

h) Model-free.

i) Problems with large state-space, or continuous state and action spaces.

Problem 2

Let $\{X_k\}_{k\geq 1}$, $\{Y_k\}_{k\geq 1}$, and $\{Z_k\}_{k\geq 1}$ be sequences of i.i.d. random variables taking values, respectively, in $\{0,1\}$, $\{0,1,2\}$ and $\{0,1\}$.

$$X_k = \begin{cases} 0 & \text{w.p. } p_1, \\ 1 & \text{w.p. } p_2, \end{cases} \qquad Y_k = \begin{cases} 0 & \text{w.p. } q_1, \\ 1 & \text{w.p. } q_2, \\ 2 & \text{w.p. } q_3, \end{cases} \qquad Z_k = \begin{cases} 0 & \text{w.p. } r_1, \\ 1 & \text{w.p. } r_2, \end{cases}$$

with $p_1, p_2, q_1, q_2, q_3, r_1, r_2 > 0$ and $p_1 + p_2 = q_1 + q_2 + q_3 = r_1 + r_2 = 1$. Consider the sequence $\{S_k\}_{k\geq 0}$ defined for all $k \geq 0$ as[1]

$$S_{k+1} = \begin{cases} S_k + X_k & \text{if } S_k = 0, \\ S_k + Y_k & \text{if } S_k = 1, \\ S_k + Z_k \pmod 3 & \text{if } S_k = 2, \\ S_k & \text{otherwise,} \end{cases}$$

and $S_0 = 0$.

a) Show that $\{S_k\}_{k\geq 1}$ is a Markov chain. Precise its state space. [2 pts]

b) What is the transition matrix? [1 pt]

c) Draw the transition graph. Specify the communication classes and precise which ones are recurrent and which ones are transient. *Motivate your answer.* [2 pts]

Model, if it is at all possible, the following problem using a Markov Decision Process . Precise the time horizon, state and action spaces, the transition probabilities, the rewards and objective. *Do not try to solve the MDP.*

d) A processing unit is designed to execute a set of n tasks $\{t_1, \ldots, t_n\}$ and can function in k different modes $\{m_1, \ldots, m_k\}$. The processing unit is going to be assigned T tasks independently and in a sequential manner. Tasks are assigned according to the probability distribution $(p_1, \ldots, p_n)$. Executing a task t_i while functioning under mode m_j costs $w_{ij} > 0$. At the end of the execution, before the next task is revealed, the unit may switch to another mode m_l and incur a cost $d_{jl} > 0$, or remain in the same mode and incur cost $d_{jj} \geq 0$. The problem is to preset a sequence of modes under which the processing unit should operate so that the expected total cost is minimized. [5 pts]

[1] The modulo operation finds the remainder of the euclidean division of a positive number by another. For instance $3 \pmod 3 = 0$ and $2 \pmod 3 = 2$.

Solution

a) The state space is $\{0, 1, 2, 3\}$. Note that for all $k \geq 0$, S_{k+1} only depends on S_k and one of the random variables X_k, Y_k, Z_k which are all independent of $S_0, S_1, \ldots, S_{k-1}$, so the markov property holds.

b) The transition matrix is

$$P = \begin{bmatrix} p_1 & p_2 & 0 & 0 \\ 0 & q_1 & q_2 & q_3 \\ r_2 & 0 & r_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

c) Figure 1 illustrates the transition graph of $\{S_k\}_{k\geq 0}$.

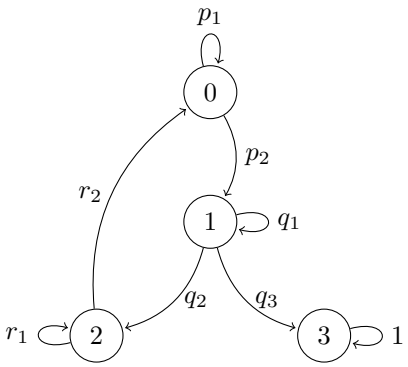

Figure 1: Transition graph of $\{S_k\}_{k\geq 0}$

There are two communication classes: $\{0, 1, 2\}$ and $\{3\}$. To see that, recall that $p_2, q_2, r_2 > 0$, so the states $0, 1$ and 2 communicate. Additionally, no state is accessible from 3.

The classe $\{0, 1, 2\}$ is transient (because $q_3 > 0$ and 3 is an absorbing state), and $\{3\}$ is recurrent (because 3 is an absorbing state).

d) (Modelling problem)

- (Time horizon) T.
- (State space) $\mathcal{S} = \{m_1, \ldots, m_k\} \times \{t_1, \ldots, t_n\}$. A state $s \in \mathcal{S}$ is a pair (m, t) indicating the the mode m under which the unit is functioning, and the task t to be executed.
- (Action space) $\mathcal{A} = \{m_1, \ldots, m_k\}$. At the end of a task execution, the unit selects the mode under which to function for the next assignment.
- (Transition probabilities) $\mathbb{P}((a, t_j)|(m, t_i), a) = p_j$.
- (Rewards) The non terminal reward is $r_s((m_j, t_i), m_l) = -(w_{ij} + d_{jl})$, and the terminal reward is $r_T((m_j, t_j)) = -w_{ij}$.
- (Objective) $\max_{a_1,\ldots,a_{t-1}} \mathbb{E}\left[\sum_{t=1}^{T-1} r_t(s_t, a_t) + r_T(s_T)\right]$

Motivation

You may think that the processing unit has an internal memory with rapid access and an external memory with a slow access. Functioning under a certain mode, corresponds to only using the currently loaded set of instructions in the internal memory. Switching modes then, corresponds to loading a set of instructions from the external memory to the internal memory, hence the switching costs (d_{ij}). The processing unit may execute the same task with different set of instructions, but some are more optimized for the execution of a certain task than others, hence the execution costs (w_{ij}).

Problem 3

Your tomatoes are attacked by a parasite. At the beginning of day 1, you have $N \geq T(T+1)/2$ healthy tomatoes (and 0 infected tomato). Your tomatoes need to grow T days before you can sell them at the market. At the beginning of a day $t \geq 1$, if there were n newly infected tomatoes during day $t-1$, then $n + Z_t$ *new* tomatoes are infected at the end of day t, where Z_t is a Bernoulli random variable with mean μ ($\mathbb{P}[Z = 0] = 1 - \mu$ and $\mathbb{P}[Z = 1] = \mu$). $(Z_t)_{t \geq 1}$ are i.i.d.. This infection process corresponds to a classical model where each newly infected tomato infects exactly one more tomato, and where there is an additional tomato infected with probability μ every day.

At the beginning of a day, you may decide to remove infected tomatoes so that they do not contaminate the others, but this work has a cost $c > 0$. At the end of day T, you pick healthy tomatoes and sell them at a unit price p. Your objective is to maximize your expected revenue at the end of the T days.

(a) Model this problem as a Markov Decision Process (describe this MDP in detail). [3pts]

(b) Denote by $P(t)$, the day when you removed infected tomatoes the last time before day t and let $w_t = \sum_{j=P(t)}^{t-1} Z_j$. Establish that the optimal policy is threshold-based, i.e., at the beginning of day t, the optimal action is to remove infected tomatoes if and only if $w_t \geq \alpha_t$ for some threshold α_t. [3pts]

(c) Assume from now on that $c = 25$, $p = 1$, and $\mu = 0.5$. Compute the optimal policy for the two last days. [2pts]

(d) With the same numerical example as that considered in the previous question, establish that under the optimal policy, you never remove the infected tomatoes at the beginning of the last day. [2pts]

Solution:

(a) There are T days, we decide an action at the beginning of each day, and we let a_t be the action taken at the beginning of day t. At the end of day t, we observe the realization of Z_t. An important quantity is w_t, the amount of newly infected tomatoes during day $t-1$. We have:

$$w_t = \sum_{j=P(t)}^{t-1} Z_j,$$

where $P(t)$ is the day before day t where you removed the infected tomatoes last. w_t is important because the number of newly infected tomatoes in day t will be either w_t or $w_t + 1$ if we do not remove infected tomatoes at the beginning of day t. And it will be 0 or 1 if we remove infected tomatoes at the beginning of day t. w_t can hence serve as the state.

Time horizon. T days.

States. The state at the beginning of day t can be described by t and the number w_t of newly infected tomatoes in day $t-1$. Hence $S = (\{1, ..., T\} \cup \{0, 1, ..., T\})$ (the first coordinate represents t, the second represents w_t).

Actions. There are two actions continue (C) or remove infected tomatoes (R). One of these actions is taken at the beginning of each day.

Rewards. The reward function is defined as follows. At the beginning of day t, we count the number of tomatoes that have been infected in day $t-1$. And at the end of the last day T, we count the number of infected tomatoes in day T. This yields:

$$\forall t = 1, \ldots, T-1, \quad r((t, w), X) = -w \times p - c \times 1_{\{a_t = R\}},$$

$$r(T, w) = -w \times p \quad \text{(terminal cost)}.$$

Transition probabilities. We have:

$$p(w|w,C) = 1-\mu, \quad p(w+1|w,C) = \mu,$$
$$p(0|w,R) = 1-\mu, \quad p(1|w,R) = \mu.$$

The objective is to maxmize the cumulative expected reward:

$$\mathbb{E}[\sum_{t=1}^{T-1} r(w_t, a_t) + r(w_T)].$$

(b) To simplify the notations, we let $V_t(w)$ denote the value function at the beginning of day t when the state is w. We first investigate the case where there is a single decision remaining, i.e., at the beginning of day T.

$$V_T(w) = \max\{-c - wp - \mu p, -2wp - \mu p\}.$$

Hence at the beginning of day T, it is optimal to remove the infected tomatoes if and only if $pw_T \geq c$. Now consider $t < T$. Bellman's equation gives:

$$V_t(w) = \max\{-c - wp + \mu V_{t+1}(1) + (1-\mu)V_{t+1}(0), -wp + \mu V_{t+1}(w+1) + (1-\mu)V_{t+1}(w)\}.$$

Hence, it is optimal to remove infected tomatoes at the beginning of day t if and only if:

$$-c + \mu V_{t+1}(1) + (1-\mu)V_{t+1}(0) > \mu V_{t+1}(w+1) + (1-\mu)V_{t+1}(w).$$

Now observe that the function $(1-\mu)V_{t+1}(w) + \mu V_{t+1}(w+1)$ is decreasing in w. We deduce that it is optimal to remove tomatoes iff $w \geq \alpha_t$. The optimal strategy is hence threshold based.

(c) From the previous question, the optimal policy is to remove tomatoes at the beginning of the day T if and only if $w_T \geq 25$. We also have: $V_T(w) = -0.5 - w - \min(25, w)$. Now we have:

$$V_{T-1}(w) = \max\{-25 - w + 0.5(V_T(0) + V_T(1)), -w + 0.5(V_T(w) + V_T(w+1))\}.$$

We have $V_T(0) = -0.5$ and $V_T(1) = -1.5$. Hence:

$$V_{T-1}(w) = \max\{-26 - w, -2w - 0.5 - 0.5(\min(25, w) + \min(25, w+1))\}.$$

On can easily check that if $w > 23$, then it is optimal to remove infected tomatoes. Consider the case $w \leq 23$. Then:

$$V_{T-1}(w) = \max\{-26 - w, -3w - 1\}.$$

It is optimal to remove infected tomatoes iff $w \geq 12.5$.

(d) At the beginning of day $T-1$, infected tomatoes are removed if $w \geq 13$. In particular, we have $w_T \leq 14$, and hence from the optimal policy, we do not remove infected tomatoes at the beginning of the last day.

Problem 4

Consider a discounted MDP with $\mathcal{S} = \{\text{A, B}\}$ and $\mathcal{A} = \{a, b\}$. We will use a Q-learning algorithm with learning rate $\alpha = 0.1$ and discount factor $\lambda = 0.5$. With initial Q values $Q^{(0)} = 0$, we get the following updates (the algorithm terminates after the last update):

$$Q^{(1)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} a & b \end{matrix} \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}, \quad Q^{(2)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} a & b \end{matrix} \begin{bmatrix} 10 & 20 \\ 0 & 0 \end{bmatrix}, \quad Q^{(3)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} a & b \end{matrix} \begin{bmatrix} 10 & 20 \\ 30 & 0 \end{bmatrix}$$

$$Q^{(4)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} a & b \end{matrix} \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}, \quad Q^{(5)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} a & b \end{matrix} \begin{bmatrix} 20 & 20 \\ 30 & 40 \end{bmatrix}, \quad Q^{(6)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} a & b \end{matrix} \begin{bmatrix} 20 & 10 \\ 30 & 40 \end{bmatrix}$$

a) Infer and state the trajectory observed by the algorithm (as a sequence of triplets of state, action, and reward.) [3 pts]

b) Assume that the inferred trajectory is:

$$(A, a, 100); (A, b, 200); (B, a, 300); (B, b, 390); (A, a, 100); (A, b, 200).$$

Using this trajectory, apply the SARSA algorithm with learning rate $\alpha = 0.5$ and discount factor $\lambda = 0.9$. State the updated Q-tables. [3 pts]

c) Assume that we are going to use an e-greedy policy with $\epsilon = 0.2$. What is the probability of taking the greedy action at a given state? [2 pts]

d) What are the greedy policies of Q-learning and SARSA with respect to their last updated Q-tables? Are any of them optimal? [2 pts]

Solution:

a) At each time step, only one element of the Q-table is updated at a time, which corresponds to the state-action pair observed by the algorithm, thus the trajectory is:

$$(A, a, ?); (A, b, ?); (B, a, ?); (B, b, ?); (A, a, ?); (A, b, ?).$$

It remains to compute the rewards, which can be done using the Q update:

$$Q^{(t+1)}(s_t, a_t) = (1 - \alpha)Q^{(t)}(s_t, a_t) + \alpha(r_t + \lambda \max_a Q^{(t)}(s_{t+1}, a))$$

Applying this update yields $r_1 = 100$, $r_2 = 200$, $r_3 = 390$, $r_4 = 100$. Since the algorithm terminates after six time steps, we also have $Q^{(6)}(A, b) = (1 - \alpha)Q^{(5)}(A, b) + \alpha r_5$, so that $r_5 = -80$. The resulting trajectory is:

$$(A, a, 100); (A, b, 200); (B, a, 300); (B, b, 390); (A, a, 100); (A, b, -80).$$

b) The SARSA update is:

$$Q^{(t+1)}(s_t, a_t) = (1 - \alpha)Q^{(t)}(s_t, a_t) + \alpha(r_t + \lambda Q^{(t)}(s_{t+1}, a_{t+1}))$$

Applying it to our trajectory yields:

$$Q^{(1)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} \begin{matrix} a & b \end{matrix} \\ \begin{bmatrix} 50 & 0 \\ 0 & 0 \end{bmatrix} \end{matrix}, \quad Q^{(2)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} \begin{matrix} a & b \end{matrix} \\ \begin{bmatrix} 50 & 100 \\ 0 & 0 \end{bmatrix} \end{matrix}, \quad Q^{(3)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} \begin{matrix} a & b \end{matrix} \\ \begin{bmatrix} 50 & 100 \\ 150 & 0 \end{bmatrix} \end{matrix}$$

$$Q^{(4)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} \begin{matrix} a & b \end{matrix} \\ \begin{bmatrix} 50 & 100 \\ 150 & 217.5 \end{bmatrix} \end{matrix}, \quad Q^{(5)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} \begin{matrix} a & b \end{matrix} \\ \begin{bmatrix} 120 & 100 \\ 150 & 217.5 \end{bmatrix} \end{matrix}, \quad Q^{(6)} = \begin{matrix} \\ A \\ B \end{matrix} \begin{matrix} \begin{matrix} a & b \end{matrix} \\ \begin{bmatrix} 120 & 10 \\ 150 & 217.5 \end{bmatrix} \end{matrix}$$

Where for the last update, we used $Q^{(6)}(A, b) = (1 - \alpha)Q^{(5)}(A, b) + \alpha r_5$, since we know that the algorithm terminates.

c) Denote by $a^\star$ the greedy action at a given state. Then we have $\mathbb{P}[A_t = a^\star] = (1-\epsilon)+\frac{\epsilon}{2} = 0.9$.

d) The greedy policy of both Q-Learning is $\pi(A) = a$, $\pi(B) = b$, while the greedy policy of SARSA is $\pi(A) = b$, $\pi(B) = b$. Only the greedy policy of Q-Learning can converge to the optimal policy, but as we have terminated the algorithm in finite time without checking for convergence, we cannot verify its optimality.

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. Recall also that $|\Sigma|$ is the determinant. One way to parametrize $\pi(s,a)$ is to "tilt" the density by a parameter $\theta \in \mathbb{R}^d$. We define the so-called exponentially-tilted density by:

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

The following three exercises familiarize you with the tilted density.

a) Show that $\pi_\theta(s,a)$ is a well-defined probability density over actions (that is, prove that it is non-negative and integrates to 1). *Hint: It may be useful to observe that the normalizing constant in the denominator is a moment generating function and satisfies*

$$e^{\theta^\top \mu + \frac{1}{2}\theta^\top \Sigma \theta} = \int_{\mathbb{R}^d} e^{\theta^\top a}\pi(s,a)da = \mathbb{E}_{A\sim\pi(s,\cdot)} e^{\theta^\top A}.$$

You do not have to prove this identity. [2 pts]

b) Show that $\pi_\theta(s,a)$ above is also Gaussian with covariance Σ but that the new mean of $a \sim \pi_\theta(s,a)$ above is $\mu + \Sigma\theta$ instead of μ. [3 pts]

We now use our tilted density $\pi_\theta(s,a)$ in the context of policy gradient and REINFORCE.

c) Compute the score function $\nabla_\theta \log \pi_\theta(s,a)$.
Hint: For $x \in \mathbb{R}^d$ and $Q \in \mathbb{R}^{d\times d}$ a symmetric matrix, the gradient of a symmetric quadratic form $x^\top Qx$ is $\nabla_x x^\top Qx = 2Qx$. [2 pts]

d) What is the REINFORCE update $\theta_t \mapsto \theta_{t+1}$ upon observing the episode (trajectory) $\tau = (s_1, a_1, r_1, \ldots, s_T, a_T, r_T)$? [1 pt]

e) What is a reasonable choice for the learning rate α_k in REINFORCE? Motivate. [1 pt]

f) Can you think of any reason why tilting is a bad or good idea for policy gradient? Give one advantage and one disadvantage. [1 pt]

Problem 5 - Solution

a) Using the hint, this is more or less immediate by construction of the tilting:

$$\begin{aligned}\int_{\mathbb{R}^d} \pi_\theta(s,a)da &= \int_{\mathbb{R}^d} \left(\frac{e^{\theta^\top a}}{e^{\theta^\top \mu + \frac{1}{2}\theta^\top \Sigma \theta}}\right) \frac{\exp\left(-\frac{1}{2}(a-\mu)^\top \Sigma^{-1}(a-\mu)\right)}{\sqrt{(2\pi)^d|\Sigma|}} da \\ &= \frac{1}{e^{\theta^\top \mu + \frac{1}{2}\theta^\top \Sigma \theta}} \int_{\mathbb{R}^d} e^{\theta^\top a} \frac{\exp\left(-\frac{1}{2}(a-\mu)^\top \Sigma^{-1}(a-\mu)\right)}{\sqrt{(2\pi)^d|\Sigma|}} da \\ &= \frac{1}{\mathbb{E}_{A\sim\pi(s,\cdot)} e^{\theta^\top A}} \mathbb{E}_{A\sim\pi(s,\cdot)} e^{\theta^\top A} \\ &= 1.\end{aligned}$$

Moreover, exponentials are strictly positive, and since the original density is non-negative, the product is non-negative as required.

b) The easiest way to answer all parts of this question is to just show that we can write the new density as a Gaussian:

$$\begin{aligned}\pi_\theta(s,a) &= \left(\frac{e^{\theta^\top a}}{e^{\theta^\top \mu + \frac{1}{2}\theta^\top \Sigma \theta}}\right) \frac{\exp\left(-\frac{1}{2}(a-\mu)^\top \Sigma^{-1}(a-\mu)\right)}{\sqrt{(2\pi)^d|\Sigma|}} \\ &= \frac{\exp\left(\theta^\top a - \theta^\top \mu - \frac{1}{2}\theta^\top \Sigma \theta - \frac{1}{2}(a-\mu)^\top \Sigma^{-1}(a-\mu)\right)}{\sqrt{(2\pi)^d|\Sigma|}} \\ &= \frac{\exp\left(\theta^\top a - \theta^\top \mu - \frac{1}{2}\theta^\top \Sigma\Sigma^{-1}\Sigma \theta - \frac{1}{2}(a-\mu)^\top \Sigma^{-1}(a-\mu)\right)}{\sqrt{(2\pi)^d|\Sigma|}} \\ &= \frac{\exp\left(-\frac{1}{2}(a-\mu-\Sigma\theta)^\top \Sigma^{-1}(a-\mu-\Sigma\theta)\right)}{\sqrt{(2\pi)^d|\Sigma|}}.\end{aligned}$$

To see the that the last two lines are equal, expand

$$(a-\mu-\Sigma\theta)^\top \Sigma^{-1}(a-\mu-\Sigma\theta)$$

using that $(\Sigma\theta)^\top \Sigma^{-1} a = a^\top \Sigma^{-1}\Sigma\theta = a^\top\theta = \theta^\top a$ by symmetry of the dot product and similarly for μ. We thus recognize the above as a Gaussian density with mean $\mu + \Sigma\theta$ and covariance Σ, as desired.

c) Since $\pi_\theta(s,a)$ only depends on the θ through the tilting factor, we observe that

$$\begin{aligned}\nabla_\theta \log \pi_\theta(s,a) &= \nabla_\theta \left[\log \frac{e^{\theta^\top a}}{e^{\theta^\top \mu + \frac{1}{2}\theta^\top \Sigma \theta}} \times \pi(s,a)\right] \\ &= \nabla_\theta \left[\log \frac{e^{\theta^\top a}}{e^{\theta^\top \mu + \frac{1}{2}\theta^\top \Sigma \theta}} + \log \pi(s,a)\right].\end{aligned}$$

and the gradient of the second part is clearly zero. Using that

$$\begin{aligned}\nabla_\theta \log\left(\frac{e^{\theta^\top a}}{e^{\theta^\top \mu + \frac{1}{2}\theta^\top \Sigma \theta}}\right) &= \nabla_\theta \left(\theta^\top a - \theta^\top \mu - \frac{1}{2}\theta^\top \Sigma \theta\right) \\ &= a - \mu - \Sigma\theta.\end{aligned}$$

we therefore have

$$\nabla_\theta \log \pi_\theta(s,a) = a - \mu - \Sigma\theta.$$

d) This is found on slide 27, part 6.

e) For instance $\alpha_t \asymp 1/t$ is reasonable in light of the convergence constraints in SA.

f) Advantage: Simple to implement. Disadvantage: The covariance of the policy does not change over time (given the state), which may prove to be a serious limitation.