EL2805 Reinforcement Learning

Exam – January 2019

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

Exam (tentamen), January 9, 2019, kl 8.00 - 13.00

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. Robert Matila 0762014056
Damianos Tranos 0732797826
Alexandre Proutiere 087906351

Results. Posted no later than January 21, 2019

Good luck!

Problem 1

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

a) Let $X_1, X_2, \ldots$ be a collection of random binary variables, such that: $X_1 = 0$, $X_2 = 1$ and for all $t \geq 3$

$$X_t = \begin{cases} 1 & \text{with probability } \frac{1}{2} + \frac{X_{t-2}}{4} \\ 0 & \text{with probability } \frac{1}{2} - \frac{X_{t-2}}{4} \end{cases}$$

Is $(X_t, t \geq 0)$ a Markov chain? If not, propose a non-trivial Markov chain constructed based on the random variables $(X_t, t \geq 0)$. [3 pts]

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

c) What is the complexity of evaluating the state value function of a history-dependent policy in finite-horizon MDPs? [1 pt]

d) Consider a discounted RL problem with discount factor λ. Give a minimax lower bound on the sample complexity of a RL algorithm to determine an ϵ-optimal policy with probability $1 - \delta$. Why do we need to resort to function approximation when the state space is large? [2 pts]

e) Assume that we run a ϵ-greedy policy in SARSA algorithm, but ϵ varies over time and is equal to $1/t$ at step t. What is the average number times we explore random actions until time T? [1 pt]

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

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

a) This is not a Markov chain since

$$\mathbb{P}[X_t = 1 | X_{t-1}, X_{t-2}] = \mathbb{P}[X_t = 1 | X_{t-2}] \neq \mathbb{P}[X_t = 1 | X_{t-1}] = \mathbb{P}[X_t = 1]$$

A Markov chain can be $Y_t = (X_{2t}, X_{2t+1})$.

b) The Value Iteration and Policy Iteration algorithms.

c) $\Theta(A^T S^{T+1})$: the number of possible length-T trajectories.

d) The lower bound is $\frac{SA}{\epsilon^2(1-\lambda)^3} \log(\delta^{-1})$. When SA is large, we cannot treat RL problems without approximations. Using function approximation allows to remove the term SA in the sample complexity.

e) We have $\mathbb{P}[E_t] = 1/t$ where E_t is the event that a random action is explored at step t. Hence, the average number of times we explore is:

$$1 + \frac{1}{2} + \ldots + \frac{1}{T} = \ln T + \gamma + O(1/T)$$

where γ is the Euler-Mascheroni constant.

f) Time Difference.

g) It is a stochastic approximation algorithm (RM algorithm).

Problem 2

Model, if this is at all possible, the two following problems using a Markov Decision Process. Precise the time horizon, state and action spaces, the transition probabilities, and the rewards. *Do not try to solve these MDPs.*

a) You get to observe a fair 20-sided die being rolled T times and generating outcomes $X(i)$, $i \in \{1, ..., T\}$. You wish to stop observing the die when the product of its outcomes observed so far is as close as possible to a given number M. [5 pts]

b) You are searching for new wasp species. In each experiment, you get to observe wasps from the same species, drawn in an i.i.d. manner from the distribution $(p_1, \ldots, p_N)$ (N possible species). The cost of each experiment is $c > 0$. After each experiment, you may decide to stop or to run a new experiment. If you stop, your reward is equal to the number of species observed minus the total cost of your experiments. You wish to maximize the average reward. [5 pts]

a) The state is the current product P of the previous outcomes, so the state space can be the set of integers. We add an extra state '0' to record that we stopped. The action is to stop S or continue C. The stationary reward can be for example $r(P, C) = 0$ and $r(P, S) = -|P - M|$. Transition probabilities are: $p(P.k|P, C) = 1/20$ for $k = 1, \ldots, 20$; $p('0'|P, S) = 1 = p('0'|'0', a)$ for $a = S$ or C.

b) The state is the number t of experiments made so far and the set A of observed species. We add the state '0' to record that we stopped. A priori we need to record the number of experiments made so far, for it impacts the optimal stopping policy through the experiment costs. The actions are to stop S or to continue C. The stationary reward is: $r((t, A), S) = |A| - t \times c$ and $r((t, A), C) = 0$. Transition probabilities:

$$p(0|(t, A), S) = 1,$$

$$\forall j \notin A,\ p((t + 1, A \cup \{j\})|(t, A), C) = p_j,$$

$$p((t + 1, A)|(t, A), C) = \sum_{j \in A} p_j.$$

Problem 3

You are driving along a street toward your destination, and you cannot do a u-turn. There are parking places along the street but most of them are taken. More precisely, you start driving from position 0, and your destination is at position T. The probability that there is an empty place at point $j \geq 0$ (j is an integer) is $1-p$, and you only discover whether this is the case when you reach point j. If you park at position j, your reward is $-|T-j|$ (the cost is the distance you have to walk to your destination). If you drive up to position T, and if there is no available place there, the optimal strategy is of course to park as soon there is an empty place, and in this case the average reward is $-1/(1-p)$. Now you have to design an optimal strategy before you reach position T, i.e., a strategy that decides when to park to maximize the average reward.

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

b) Establish that if it is optimal to park in position $j < T$, it is also optimal to park at position $j+1$. Hint: Use Bellman's equation. [2pts]

c) Deduce that the optimal strategy is threshold-based. Let $\mathcal{P}_r$ denote the threshold-based policy that consists in parking at the first available place at a distance less or equal to r from the destination, and denote by P_r its average reward.
Prove that $P_r = -(1-p)r + pP_{r-1}$. [2pts]

d) Show by induction on r that $P_r = -(r+1) - \frac{2p^{r+1}-1}{1-p}$. Identify the optimal policy, i.e., the optimal threshold $r^\star$. Hint: consider $P_{r+1} - P_r$. [3pts]

a) We have a time horizon T, since we know what to do after we passed position T and the associated average reward. The state of the system at time j is just 0 if there is a vacant place at position j and 1 otherwise. In addition we need a state indicating that we already parked, let us call $\emptyset$ this state: $S = \{0, 1\} \cup \emptyset$. The two possible actions are park (P) or continue (C), and the transition probabilities are: for $j < T$, $p_j(\emptyset|0, P) = 1$, $p_j(0|0, C) = p(0|1, C) = 1 - p$, and $p_j(1|0, C) = p_j(1|1, C) = p$. For $j = T$, $p_j(\emptyset|y, x) = 1$ for any $y \in \{0, 1\}$ and $x = P$ or C. The rewards are for $j < T$, $r_j(0, P) = j - T$, $r_j(1, C) = r_j(0, C) = r_j(\emptyset, x) = 0$, and $r_T(0, P) = 0$, $r_T(1, C) = r_T(0, C) = -1/(1 - p)$.

b) Bellman's equation is: $V_T(0) = 0$, $V_T(1) = -1/(1 - p)$, and for $j < T$,

$$\begin{aligned} V_j(0) &= \max(j - T, pV_{j+1}(1) + (1 - p)V_{j+1}(0)) \\ V_j(1) &= pV_{j+1}(1) + (1 - p)V_{j+1}(0). \end{aligned}$$

From the above two equations we deduce that:

$$\begin{aligned} V_j(0) &\geq pV_{j+1}(1) + (1 - p)V_{j+1}(0) \\ V_j(1) &\geq pV_{j+1}(1) + (1 - p)V_{j+1}(0). \end{aligned}$$

and thus

$$pV_j(0) + (1 - p)V_j(1) \geq pV_{j+1}(1) + (1 - p)V_{j+1}(0). \tag{1}$$

Now assume that it is optimal to park in position j. It means from Bellman's equation that:

$$j - T \geq pV_{j+1}(1) + (1 - p)V_{j+1}(0)$$

Applying (1), we get:

$$\begin{aligned} j + 1 - T \geq j - T &\geq pV_{j+1}(1) + (1 - p)V_{j+1}(0) \\ &\geq pV_{j+2}(1) + (1 - p)V_{j+2}(0), \end{aligned}$$

which means that it is optimal to park in position $j + 1$. We have shown that the optimal policy is threshold-based.

c) P_r is the average reward of policy $\mathcal{P}_r$. If in position r from the destination, the place is vacant, then we park and the reward is $-r$, otherwise we move to the next position and get a reward P_{r-1}. Hence $P_r = -r(1 - p) + P_{r-1}p$.

d) Note that the result holds for $r = 0$, because $P_0 = -p/(1 - p)$. Assume that the result holds for $r - 1$. Then:

$$P_r = -r(1 - p) + p(-r - \frac{2p^r - 1}{1 - p}) = -(r + 1) - \frac{2p^{r+1} - 1}{1 - p}.$$

Observe that $P_{r+1} - P_r = 2p^{r+1} - 1$. Hence the optimal threshold is:

$$r^\star = \min\{r \geq 0 : p^{r+1} \leq 1/2\}.$$

Problem 4

Consider a discounted MDP with $\mathcal{S} = \{\text{A, B, C}\}$ and $\mathcal{A} = \{a, b, c\}$. We plan to use Q-learning or SARSA algorithm, and initialize the estimated Q-function as

$$Q^{(0)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 10 & 20 & 20 \\ 0 & 10 & 40 \\ 50 & 10 & 15 \end{bmatrix} \end{array} .$$

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

$$(?, ?, ?); (A, a, 80); (A, b, 60); (B, c, 70); (B, b, 40); (C, a, 20); (C, b, 0); \ldots$$

where each triplet represents the state, the selected action, and the corresponding reward. Note that we do not observe the first state, the first selected action, and the first observed reward.

a) Assume that we run the Q-learning algorithm, and that

$$Q^{(1)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 10 & 20 & 20 \\ 0 & 20 & 40 \\ 50 & 10 & 15 \end{bmatrix} \end{array} .$$

Assume that the discount factor is $\lambda = 0.5$ and the learning rate is fixed to $\alpha = 0.1$. Can you infer the first state, the first selected action, and the first observed reward? [2 pts]

b) Provide the updated Q-values, if we are using the Q-learning algorithm, after these observations, assuming that the discount factor is $\lambda = 0.5$ and the learning rate is fixed to $\alpha = 0.1$. [3 pts]

c) What is the current greedy policy after these updates? [1 pts]

d) Provide the updated Q-values, if we are using SARSA and if the first triplet (state, action, reward) is given by the answer of a), assuming that the discount factor is $\lambda = 0.5$ and the learning rate is fixed to $\alpha = 0.1$. [3 pts]

e) What is the current greedy policy after these updates? [1 pts]

a) The only entry of the matrix that changed is that for (B, b), hence the first state is B and the first action is b. Now let R denote the first observed reward. We have:

$$\begin{aligned} Q^{(2)}(B,b) &= Q^{(1)}(B,b) + 0.1 \times (R + 0.5 \times \max_{\phi} Q^{(1)}(A,\phi) - Q^{(1)}(B,b)) \\ &= 10 + 0.1 \times (R + 10 - 10) \end{aligned}$$

Hence $R = 100$.

b)

$$Q^{(6)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 18 & 26 & 20 \\ 0 & 24.5 & 45 \\ 49.5 & 10 & 15 \end{bmatrix} \end{array} .$$

c) The greedy policy is $\pi(A) = b$, $\pi(B) = c$, $\pi(C) = a$.

d)

$$Q^{(6)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 18 & 26 & 20 \\ 0 & 23.95 & 44 \\ 47.5 & 10 & 15 \end{bmatrix} \end{array} .$$

e) The greedy policy is $\pi(A) = b$, $\pi(B) = c$, $\pi(C) = a$.

Problem 5

Policy gradient. We consider an episodic RL problem where we parametrize the policy using parameter $\theta \in [0, 1]$ as follows. In any state s, we let $f(s) \in [1, 2]$, and draw Z a random variable uniformly at random in $[-f(s), f(s)]$. Three actions are possible, -1, 0, and 1. If $Z \leq -\theta$, then -1 is selected; if $Z \geq \theta$, 1 is selected; and if $-\theta < Z < \theta$, 0 is chosen.

a) Compute in state s, the probabilities $\pi_\theta()s, a)$ to select the three actions. [1 pt]

b) Do you think that this policy parametrization is appropriate for generic problems with three actions? [1 pt]

c) 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 [3 pts]

SARSA with function approximation. Next we consider a discounted RL problem, that we wish to solve using approximations of the (state, action) value function (i.e., parametrized by a vector θ).

d) We observe the transition $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$. Recall the Q update in the SARSA algorithm with function approximation. Why is it called a semi-gradient algorithm? [2 pts]

e) Propose another SARSA algorithm with function approximation that we could call a gradient algorithm. [3 pts]

a) We have:

$$\pi_\theta(s,1) = \pi_\theta(s,-1) = \frac{f(s)-\theta}{2f(s)}, \qquad \pi_\theta(s,0) = \frac{\theta}{f(s)}.$$

b) This is not a good parametrization since actions 1 and -1 are always selected with the same probability under these policies. Hence if e.g. 1 is the unique best action in a given state s, we would not be able to devise an algorithm converging to an approximately optimal policy.

c) The algorithm is given slide 27 Part 5. We just need to compute $\nabla \ln \pi_\theta(s,a)$.

$$\nabla \ln \pi_\theta(s,1) = \nabla \ln \pi_\theta(s,-1) = \frac{1}{\theta - f(s)}$$

$$\nabla \ln \pi_\theta(s,0) = 1/\theta.$$

d) The SARSA update is:

$$\theta \leftarrow \theta + \alpha(r_t + \lambda Q_\theta(s_{t+1},a_{t+1}) - Q_\theta(s_t,a_t))\nabla Q_\theta(s_t,a_t).$$

It is referred to a semi-gradient algorithm because half of the TD term is differentiated w.r.t. θ.

e) A gradient algorithm would be:

$$\theta \leftarrow \theta + \alpha(r_t + \lambda Q_\theta(s_{t+1},a_{t+1}) - Q_\theta(s_t,a_t))(\nabla Q_\theta(s_t,a_t) - \lambda \nabla Q_\theta(s_{t+1},a_{t+1})).$$