EL2805 Reinforcement Learning

Exam – April 9, 2021

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

Exam (tentamen), April 9, 2021, kl 8.00 - 13.00

Aids. Slides of the lectures (**not exercises**), lecture notes.

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.
You will **stop writing at 12:40** to ensure you have enough time to upload your solution on canvas.

The exam consists in 5 problems. The grading policy will be adapted to the fact that the zoom exam is 20 mins shorter than the on-site exam.

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 28, 2021

Good luck!

Problem 1

a) What does it mean that "TD-learning methods *boostrap*"? [1 pt]

b) Consider the DQN algorithm. Explain if it is a good idea to increase the learning rate when we increase the batch size. *Hint: think of the variance of the gradients.* [1 pt]

c) Give two reasons why you would use function approximation in RL, each with an example to illustrate your point? [2 pts]

d) Consider Q-learning or SARSA, implemented both with an ϵ-greedy policy. Give a reason why you would prefer Q-learning rather than SARSA, and another reason why you would prefer SARSA rather than Q-learning? [1 pt]

e) Consider an infinite time-horizon discounted MDP with finite state and action spaces $\mathcal{S}$ and $\mathcal{A}$. Denote by $p(\cdot|s,a)$ the distribution of the next state starting in state s and selecting action a. Assume that the reward collected in step t is $r(s_t, a_t, s_{t+1})$ where s_t and a_t denote the state and the selected action at time t (in the course, we assumed that this reward was not a function of s_{t+1}, i.e., it was $r(s_t, a_t)$). Write Bellman's equations for this kind of scenario. How is the Q-function defined? [2 pts]

Polya's Urn Suppose that you are given a large urn with two balls in it, one red and one blue (at $t = 1$). In each round thereafter, you choose a ball uniformly at random from the urn, and inspect its color. You put the ball back in the urn, and add in the urn another ball of the same color as the one you just picked up. Denote by B_t the number of blue balls at time t and by R_t the number of red balls at time t. Note that $\{B_t\}_{t\geq 1}$ and $\{R_t\}_{t\geq 1}$ are Markov chains.

f) What is the state space of $\{B_t\}_{t\geq 1}$? [1pt]

g) What are the transition probabilities for $\{B_t\}_{t\geq 1}$? [1pt]

h) Specify the transient and recurrent classes of $\{B_t\}_{t\geq 1}$. [1pt]

Problem 1 - Answers

a) TD learning methods bootstrap means that the target value depends on the current estimate of the value function.

b) In general it is a good idea. If we increase the batch size it means that the gradients will be more accurate. Therefore, we can afford using larger learning rates thanks to this improved accuracy in the gradients. As a matter of fact, consider the case where the batch is small, and consists of just 1 sample. The gradients would have a lot of variance. Therefore, in this case, it does not make sense to use a larger learning rate (since we are not really sure of the gradient!).

However, even if we use a large batch size, it does not imply that we can use a large learning rate, since the landscape of the loss function is not convex in deep learning (and large learning rates may lead to oscillations).

c) Here are the two reasons with examples:

Reason 1. We use function approximation to deal with computational challenges where the dynamics and rewards may be known but the state and/or action spaces are extremely large. For example, in the strategy game Go, there are roughly 10^{170} board configurations[1]. This renders solving the game using brute force impossible, eventhough the outcome of any strategy is perfectly predictable.

Reason 2. We can also use function approximation to deal with continuous state and/or action spaces with uncertainties about the environement. Such problems arize in control, for example, teaching an agent how to fly and maneouvre a helicopter under challenging weather conditions.

d) A good reason for prefering Q-learning rather than SARSA is that Q-learning converges to the optimal Q-function while SARSA dose not. A good reason for prefering SARSA rather than Q-learning is that SARSA can be safer in comparison with Q-learning since it will learn the best policy in light of a prescribed behaviour policy.

e) Bellman's equations characterize the value: for all $s \in \mathcal{S}$,

$$V(s) = \max_{a \in \mathcal{A}} \sum_{s'} p(s'|s,a) \left(r(s,a,s') + \lambda V(s') \right).$$

The Q-function is defined by: for all (s,a),

$$Q(s,a) = \sum_{s'} p(s'|s,a) \left(r(s,a,s') + \lambda V(s') \right).$$

f) $B_0 = 1$ and it is possible to reach every integer greater than 2 from this, so $S = \mathbb{N}$.

g) If there are j blue balls at stage t, where t balls in total have been added (yielding a total of $t+2$ balls), then the probability of selecting a blue ball is

$$\frac{j}{t+2}.$$

Hence

$$P(B_{t+1} = j+1 | B_t = j) = \frac{j}{t+2}$$
$$P(B_{t+1} = j | B_t = j) = 1 - \frac{j}{t+2}$$

and the transition probabilities are zero otherwise.

h) All states are in their own class, and transient.

[1] that's more than the number of atoms in the universe

Problem 2 – Modelling using MDPs

Input design for linear system identification. Consider the following scenario. Time is discrete. In each round $t \geq 1$, you select an input $a_t \in \mathbb{R}^d$ that is one of the vectors composing the canonical basis $\{e_i, i = 1, \ldots, d\}$ (where e_i is the vector whose coordinates are all 0 except for the i-th, equal to 1). As a consequence of your choice, you observe the scalar y_t given by

$$y_t = \theta^\top a_t + w_t,$$

where $\theta \in \mathbb{R}^d$ and $w_t \in \mathbb{R}$ has distribution $p(\cdot)$ whose support is finite and equal to $\{-N, \ldots, N\}$. The random variables $\{w_t\}_{t\geq 1}$ are independent. Your goal is to estimate θ using the noisy observations y_t only. To this aim, you use the (regularized) mean squares estimator, which after T observations is given by:

$$\hat{\theta}_T = \left(\mu I_d + \sum_{t=1}^{T} a_t a_t^\top\right)^{-1} \sum_{t=1}^{T} a_t y_t$$

for some fixed parameter $\mu > 0$ and I_d is the identity matrix of dimension d. Your goal is to choose the successive inputs $a_t \in \mathbb{R}^d$ according to some policy π to minimize the least square error:

$$\mathbb{E}^\pi (\theta - \hat{\theta}_T)^\top (\theta - \hat{\theta}_T).$$

a) Model the above problem as an MDP. Do not try to solve it. [5pts]

A time-varying communication channel. Alice wishes to communicate to Bob over a noisy communication channel. Alice is allowed to send N bits in total over the time horizon of T rounds. During round t, a bit sent to Bob is received successfully with probability P_t. P_t evolves as follows: $P_1 = 0.4$, and for all $t \geq 1$[2],

$$10P_{t+1} = (10P_t + 2\rho_t) \mod 10$$

where $\{\rho_t\}_{t\geq 1}$ is a sequence of i.i.d. Rademacher random variables, taking values 1 and -1 with equal probability.

At the beginning of each round t, Alice observes the channel quality represented by P_t, and decides to send in this round up to $M(\leq N)$ bits. At the end of the round, Alice gets to know how many bits sent in the round have been successfully received by Bob. Alice wishes to design a transmission strategy maximizing the total expected number of bits received by Bob in the T rounds. Remember that this number is less than N since Alice cannot send more than N bits overall.

b) Model the above problem as an MDP. Do not try to solve it. [5pts]

[2]For an integer $p \geq 0$, $(p \mod 10)$ is an integer in $\{0, \ldots, 9\}$ which is remaining of the Euclidian division of p by 10, i.e., $r = (p \mod 10)$ if $p = 10k + r$, where k is an integer.

Problem 2 – Answers

a) The possible actions are simply the possible design vectors at each stage; that is, $A = \{e_i : i = 1, \ldots, d\}$ where, as before, e_i are standard basis vectors.

The state can then be chosen as the history over all actions and outputs y_t. Since each y_t has support on $Y = \{\theta^\top e_i + j : i = 1, \ldots, d \text{ and } j \in \{-N, \ldots, N\}\}$ the state space can be chosen as

$$S = \{(\underbrace{Y}_{\text{observation } y_t} \times \underbrace{A}_{\text{action } a_t}) \cup \underbrace{(0,0)}_{t \text{ has not yet occured}} \}^T$$

The only part of the state which changes at time t is the t:th coordinate of s_t, denoted s_t^t. This has dynamics

$$p(s_{t+1}^{t+1} = (s', a') | s_t = s, a_t = a) = \mathbf{1}_{a'=a} p(s' - \theta^\top a).$$

As for the rewards, they are given by

$$r_t = -\mathbf{1}_{t=T} (\theta - \hat{\theta}_T)^\top (\theta - \hat{\theta}_T)$$

which depends on s_t through the definition of $\hat{\theta}_T$. The objective is to maximize the cumulative expected reward

$$\mathbb{E}^\pi \sum_{t=1}^{T} r_t$$

which of course is equivalent to simply minimizing $\mathbb{E}^\pi (\theta - \hat{\theta}_T)^\top (\theta - \hat{\theta}_T)$.

b) The state-space is

$$S = \underbrace{\{0, 1, \ldots, N\}}_{\#\text{ bits Alice has left}} \times \underbrace{\{0, 0.2, 0.4, 0.6, 0.8\}}_{\text{Channel transmission rate}}$$

The possible actions are

$$A_s = \underbrace{\{0, 1, \ldots, s\}}_{\#\text{ bits Alice attemps to send}}$$

Let $s_t = (i_t, P_t)$. The transition probabilities are

$$p((i_{t+1}, P_{t+1}) = (i, p) | (i_t, P_t) = (j, r), a_t = a) = \mathbf{1}_{j-a=i, |r-p|=0.2} \times 0.5.$$

and the rewards, for $s = (i, p)$ are given by

$$r_t(s, a) = Pa$$

where P denotes the channel state coordinate of s. The goal is to maximize the expected total reward

$$\mathbb{E}^\pi \sum_{t=1}^{T} r_t.$$

Problem 3 – Research funding

You have become the prime minister of Sweden, and will remain at this position for T consecutive years. One of your duties is to define the research funding policy, in particular in bio-technology. At the beginning of each year, and more precisely January 1, you decide or not to invest in research. This investment, if decided, has a fixed cost c, that you pay immediately. At the beginning of each year, and more precisely January 2, a pandemic starts with probability p. If a pandemic starts, the remaining of your mandate is devoted to cope with it, you cannot invest in research, and experience a cost equal to $g(x)$ per year until the end of your mandate, where x is the number of years (the investment you did the day before does not count in x) where you invested in research so far. g is a function that indicates the benefit of investing in research (e.g., being able to come up with a vaccin). g is decreasing and concave. Your objective is to identify the investment policy that minimize the expected cumulative cost.

a) Model the problem as an MDP. You may use reward functions that depend on the initial state s, the action a, and the next state s' (in the course, the last dependence was absent).[3 pts]

b) Write Bellman's equations and at the beginning of the t-th year, provide a condition (depending on the value function at year $t+1$) under which it is optimal to invest. [1 pt]

c) Establish that the optimal policy is threshold-based. This means that at the beginning of year $t \in \{1, \dots, T\}$, if no pandemic has started yet, you can decide to invest in research by only comparing the number of years x you invested in research in the past to a threshold (that depends on t).*[Hint: since g is concave, we have $x \mapsto g(x) - g(x+1)$ is increasing. You may wish to establish such a property for the state-value functions (the minimum cost starting at some year as a function of the number of years you invested so far). To this aim, proceed by induction, and use the facts that if f, g are concave, then $f+g$ and $\min\{f, g\}$ are also concave.]* [3 pts]

d) What can we say about this optimal policy when g is a linear function? [1 pt]

e) Consider the policy that invests only the X first year without pandemic. Compute its average cost. [2 pts]

Problem 3 – Answers

a) This is a finite time horizon MDP, with horizon T. The state x at the beginning of year t (just before January 1) represents the number of years where you invested so far and the absence of a pandemic. In addition we add the state $\emptyset$ representing the presence of the pandemic. The actions available in state $x \neq \emptyset$ are I (investment) and N (no investment), and there is no action available in state $\emptyset$. The transition probabilities are as follows:

$$\forall x \neq \emptyset,\ \mathbb{P}[x+1|x,I] = (1-p),\ \mathbb{P}[x|x,N] = (1-p),$$
$$\mathbb{P}[\emptyset|x,I] = p,\ \mathbb{P}[\emptyset|x,N] = p,$$
$$\mathbb{P}[\emptyset|\emptyset,\cdot] = 1.$$

The reward functions are for all t: $r_t(x,I,x+1) = -c$, $r_t(x,\cdot,\emptyset) = -(T-t+1)g(x)$, and $r(\emptyset,\cdot,\emptyset) = 0$.

b) Bellman's equations provide recursive equations characterizing the value functions $u_t^\star$ mapping the state at time t to the minimum expected cost of the best policy. We have:
$u_T^\star(x) = pg(x)$ (this is the expected cost for the last year given that no pandemic has started earlier), and $u_T^\star(\emptyset) = 0$.
For all $t < T$, $u_t^\star(\emptyset) = 0$, and

$$u_t^\star(x) = pg(x)(T-t+1) + \min\{c + (1-p)u_{t+1}^\star(x+1), (1-p)u_{t+1}^\star(x)\}.$$

c) We prove by a backward induction on t that the function $x \mapsto u_t^\star(x)$ is concave in x.
For $t = T$, this is true since g is concave.
Assume that the result holds for $t+1$. Now from Bellman's equations above, $u_t^\star$ is the sum of two functions. The first one involves g only and is concave. The second one is written as the minimum of two concave functions defined from $u_{t+1}^\star$ and is also concave. Hence $u_t^\star$ is concave. This implies that the function $h_t(x) = u_t^\star(x) - u_t^\star(x+1)$ is increasing in x.

Now from Bellamn's equations, it is optimal to invest if and only if:

$$c + (1-p)u_{t+1}^\star(x+1) \leq (1-p)u_{t+1}^\star(x),$$

which is equivalent to $h_{t+1}(x) \geq \frac{c}{(1-p)}$. Since h_{t+1} is increasing, we deduce that it is optimal to invest if and only if $x \geq \gamma_t$ where γ_t is the desired threshold.

d) When g is linear, the function $h_t(x)$ is constant, and hence the optimal policy does not depend on the state.

e) We first compute the expected cost due to investment (remember that we invest only in absence of pandemic). This cost is:

$$c\left(1 + (1-p) + (1-p)^2 + \ldots + (1-p)^{X-1}\right) = c \times \frac{1-(1-p)^X}{p}.$$

Next we compute the average cost due to the pandemic. This cost is:

$$pg(0)T + p(1-p)g(1)(T-1) + p(1-p)^2 g(2)(T-1) + \ldots + p(1-p)^{X-1}g(X-1)(T-X+1)$$
$$+ p(1-p)^X g(X)(T-X) + \ldots + p(1-p)^{T-1}g(X)$$

In summary, the cost of the policy is:

$$c \times \frac{1-(1-p)^X}{p} + p\sum_{i=0}^{T-1}(1-p)^i g(\min\{i,X\})(T-i).$$

Problem 4 - Boltzmann Policies

In this exercise, we will investigate the so-called *Boltzmann* policies. Consider a stationary MDP with finite state and action spaces S and A, with terminal state (that we denote $\emptyset$) and discounted rewards. Let $|A|$ be the number of actions. Assume that you have an estimate at time t of the Q-values $Q_t(s,a)$. From this estimated Q-function, consider the following policy described by the probability $\pi_t(s,a)$ to take action a in state s:

$$\pi_t(s,a) = \eta_t(s)(1-\zeta)e^{Q_t(s,a)} + \frac{\zeta}{|A|},$$

where $\zeta \in (0,1)$ and $\eta_t(s)$ is a normalization factor (so that $\pi_t(s,\cdot)$ is a probability distribution over the set of actions). π_t is a policy where the probability of taking a certain action a in state s depends on the weight $Q_t(s,a)$ (the higher, the more probable).

a) What is the value of $\eta_t(s)$? Motivate your answer. [1 pt]

b) Is π_t an ε-soft policy? Motivate your answer. [1 pt]

Consider now the following algorithm.

Algorithm 1

Input Episodes N; discount factor $\gamma \in (0,1)$; learning rate $\alpha_n = 1/n$

Initialize matrices $Q_0(s,a)$ and $n(s,a)$ to 0 for all (s,a)
for episodes $k = 1, 2, \ldots, N$ **do**
  $t \leftarrow 1$
  Observe initial state s_1 drawn uniformly at random
  while Episode k is not finished **do**
    Choose action a_t from the policy $\pi_t(s,\cdot)$
    Perform action a_t: observe reward r_t and next state s_{t+1}
    Compute target value: if the episode is terminal then $y_t = r_t$ otherwise

$$y_t = r_t + \gamma \max_{a'} Q_t(s_{t+1}, a')$$

    $n(s_t, a_t) \leftarrow n(s_t, a_t) + 1$
    Update Q function: $Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha_{n(s_t,a_t)}(y_t - Q_t(s_t, a_t))$
    $t \leftarrow t + 1$
  end while
end for

c) We let $N \to \infty$. To which Q-values does Alg. 1 converge? Motivate your answer. [2 pts]

d) Suppose the state space is $S = \{1,2,3\}$ and the action space is $A = \{a,b,c\}$.

d.1) You observe a trajectory $(2,c,4);(1,a,\delta);(1,b,1);(3,b,1);(1,a,0);(2,a,1);(\emptyset)$ where $\delta < 1$ and each triplet represents the state, the selected action and the corresponding reward (and $\emptyset$ is the terminal state). Assume that $\gamma = 1$ and that the learning rate is constant $\alpha = 0.5$. What are the estimated Q-values after the observed trajectory? Assume your initial Q values are 0. [2 pts]

d.2) What is the greedy policy with respect to the Q-values you found in (d.1)? [1 pt]

e) Suppose we now change the target value in Alg. 1 to $y_t = \dfrac{r_t + \gamma \max_{a'} Q_t(s_{t+1}, a')}{\sqrt{2}}$. To which Q-values does the new algorithm converge? What is the corresponding value function? [1 pt]

f) Let $a_s^\star = \text{argmax}_a\, Q^\star(s,a)$ be the optimal action in s ($Q^\star$ is the true Q-function). For some $t \geq 0$, assume that $\max_{a \neq a_s^\star}\{Q_t(s,a) - Q_t(s,a_s^\star)\} \leq -\varepsilon_s$, where $\varepsilon_s > 0$. When $\zeta = 0$, show that $\pi_t(s,a_s^\star) \geq \frac{1}{1+(|A|-1)e^{-\varepsilon_s}}$. [2 pts]

Problem 4 – Answers

a) Since π is a policy, we need $\sum_a \pi(a|s) = 1$. It follows that

$$\sum_a \eta_t(s)(1-\zeta)e^{Q_t(s,a)} + \frac{\zeta}{|A|} = \zeta + (1-\zeta)\eta_t(s)\sum_a e^{Q_t(s,a)} = 1$$

and thus

$$\eta_t(s) = \frac{1-\zeta}{(1-\zeta)\sum_a e^{Q_t(s,a)}} = \frac{1}{\sum_a e^{Q_t(s,a)}}$$

b) A ε-soft policy is any policy that satisfies $\pi(a|s) \geq \varepsilon/|A|$ for any $(s,a) \in S \times A$, where $|A|$ is the number of actions. It suffices to see that there is always a probability of $\zeta/|A|$ to take an action a (since the policy π is the convex combination of a softmax policy and a uniform policy).

c) They converge to the true Q-values. The algorithm is off-policy, and the exploration policy is ε-soft (thus we keep exploring all state/action pairs). Since the Robbins-Monro conditions are met, then, the algorithm converges to the true Q values.

d.1) $(2,c,4);(1,a,\delta);(1,b,1);(3,b,1);(1,a,0);(2,a,1);(\emptyset)$ with $\gamma = 1, \alpha = 1/2$. The sequence of Q values is as follows

$$Q^{(0)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}.$$

$Q^{(1)}(2,c) = 2$

$$Q^{(1)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix} \end{array}.$$

$Q^{(2)}(1,a) = \delta/2$

$$Q^{(2)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} \frac{\delta}{2} & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix} \end{array}.$$

$Q^{(3)}(1,b) = \frac{1}{2}$

$$Q^{(3)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} \frac{\delta}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix} \end{array}.$$

$Q^{(4)}(3,b) = \frac{3}{4}$

$$Q^{(4)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} \frac{\delta}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 2 \\ 0 & \frac{3}{4} & 0 \end{bmatrix} \end{array}.$$

$Q^{(5)}(1,a) = \frac{\delta}{2} + \frac{1}{2}(0 + 2 - \frac{\delta}{2}) = \frac{\delta}{2} + \frac{2-\delta/2}{2} = 1 + \frac{\delta}{4}$

$$Q^{(5)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 1+\frac{\delta}{4} & \frac{1}{2} & 0 \\ 0 & 0 & 2 \\ 0 & \frac{3}{4} & 0 \end{bmatrix} \end{array}.$$

For the last step, since the episode is terminal, the target value is just $y = 1$. Thereby $Q^{(6)}(2, a) = \frac{1}{2}$

$$Q^{(6)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 1 + \frac{\delta}{4} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 2 \\ 0 & \frac{3}{4} & 0 \end{bmatrix} \end{array} .$$

d.2) We have that the optimal action in state in state 2 is c and in state 3 is b. In state 1 the optimal action is a if $\delta > -2$, otherwise it's b if $\delta < -2$ (both are optimal actions for $\delta = -2$).

e) Note that the new target value is equivalent to dividing the reward and the discount factor by $\sqrt{2}$. Consequently, we are trying to minimize the following value function

$$V(s) = \frac{1}{\sqrt{2}} \mathbb{E}\left[\sum_{t \geq 0} \left(\frac{\gamma}{\sqrt{2}} \right)^t r(s_t, a_t) \Big| s_0 = s \right]$$

f) Just observe that

$$\pi_t(a_s^\star | s) = \frac{e^{Q_t(s, a_s^\star)}}{\sum_a e^{Q_t(s,a)}} = \frac{1}{1 + \sum_{a \neq a_s^\star} e^{Q_t(s,a) - Q_t(s, a_s^\star)}}$$

Using the inequality $\max_{a \neq a_s^\star} \{Q_t(s, a) - Q_t(s, a_s^\star)\} \leq -\varepsilon_s$ we can write

$$\pi_t(a_s^\star | s) \geq \frac{1}{1 + \sum_{a \neq a_s^\star} e^{-\varepsilon_s}} = \frac{1}{1 + (|A| - 1) e^{-\varepsilon_s}}$$

Note: The condition simply says that the gap between the optimal arm and the second best arm is always greater, or equal, to ε_s. Using this fact we have that

$$e^{Q_t(s,a) - Q_t(s, a_s^\star)} \leq e^{-\varepsilon_s}$$

and thus

$$1 + \sum_{a \neq a_s^\star} e^{Q_t(s,a) - Q_t(s, a_s^\star)} \leq 1 + \sum_{a \neq a_s^\star} e^{-\varepsilon_s}$$

from which follows the proof.

Problem 5

In this problem, we will derive a variant of policy gradient methods known as entropy regularized policy gradient methods. The goal of introducing entropy regularization is to encourage exploration and reduce the impact of an aggressive plain gradient ascent update. These variants of policy gradients can sometimes perform much better in practice.

Soft-max policies. Let $\mathcal{A}$ denote the action space, and $\mathcal{S}$ denote the state space. Consider policies of the form

$$\pi_\theta(s,a) = \exp(\theta^\top f(s,a) - g(s,\theta))$$

parametrized by $\theta \in \mathbb{R}^d$. The function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is given and fixed, and $g : \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}$ acts as a normalizing factor.

a) Express $g(s,\theta)$ as a function of θ and f. [1 pt]

b) Compute $\nabla_\theta \log \pi_\theta(s,a)$ and verify that $\mathbb{E}_{a\sim\pi_\theta(s,\cdot)}\left[\nabla_\theta \log \pi_\theta(s,a)\right] = 0$. [1 pt]

Entropy regularization for policy gradient methods. Consider a finite time horizon MDP. We wish to identify the policy parametrized by θ maximizing the objective function $J(\theta)$. The idea of entropy regularization is to penalize $J(\theta)$ by an entropy term $H(\theta)$. The resulting objective is

$$J(\theta) + \gamma H(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{T} r(s_t,a_t) - \gamma \log \pi_\theta(s_t,a_t)\right],$$

where $\gamma > 0$ is some regularization parameter, and here $\mathbb{E}_{\pi_\theta}$ means that the sequence $s_1, a_1, \ldots, s_T, a_T$ are generated under the policy π_θ[3].

c) Here we consider generic parametrized policies π_θ (not necessarily the soft-max policies investigated above). [3 pts]
Show that

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(s_t,a_t)\right)\left(\sum_{t=1}^{T} r(s_t,a_t) - \gamma \log \pi_\theta(s_t,a_t)\right)\right].$$

d) Propose a modification of the REINFORCE algorithm that seeks to maximize the objective $J(\theta) + \gamma H(\theta)$, and runs for n episodes. Provide the update step for the parameter θ in this algorithm for the case of soft-max policies. [2 pts]

e) Establish the following *causality principle*

$$\mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s_t,a_t)(r(s_u,a_u) - \log \pi_\theta(s_u,a_u))] = 0 \quad \text{for } u < t.$$

Propose a modification of your algorithm that can reduce its variance. Again, provide the corresponding gradient update for the case of soft-max policies. [3 pts]

[3] Here, you may assume that s_1 is either generated by some distribution $p(\cdot)$, or is fixed.

Problem 5 – Answers

a) For all $s \in \mathcal{S}$, $\pi_\theta(s, \cdot)$ must be a distribution, thus $\sum_{a\in\mathcal{A}} \pi_\theta(s, a) = 1$, which yields

$$g(s, \theta) = \log\left(\sum_{a\in\mathcal{A}} \exp(\theta^\top f(s, a))\right)$$

and we may write

$$\pi_\theta(s, a) = \frac{\exp(\theta^\top f(s, a))}{\sum_{a'\in\mathcal{A}} \exp(\theta^\top f(s, a'))}$$

b) Straightforward computations give

$$\begin{aligned}\nabla_\theta \log \pi_\theta(s, a) &= f(s, a) - \sum_{a'\in\mathcal{A}} f(s, a')\pi_\theta(s, a') \\ &= f(s, a) - \mathbb{E}_{a\sim\pi_\theta(s,\cdot)}\left[f(s, a)\right]\end{aligned}$$

we can see from the last equality that $\mathbb{E}_{a\sim\pi_\theta(s,\cdot)}\left[\nabla_\theta \log \pi_\theta(s, a)\right] = 0$

c) To fix ideas, let $\tau = (s_1, a_1, \ldots, s_T, a_T)$, and denote

$$\begin{aligned}\mathbb{P}_{\pi_\theta}(\tau) &= \pi_\theta\left(s_1, a_1\right)\mathbb{P}(s_2|s_1, a_1) \times \cdots \times \pi_\theta(s_{T-1}, a_{T-1})\mathbb{P}(s_T|s_{T-1}, a_{T-1})\pi_\theta(s_T, a_T) \\ R(\tau) &= \sum_{t=1}^{T} r(s_t, a_t) \\ C(\tau) &= \sum_{t=1}^{T-1} \log(\mathbb{P}(s_{t+1}|s_t, a_t)) = \log \mathbb{P}_{\pi_\theta}(\tau) - \sum_{t=1}^{T} \log \pi_\theta(s_t, a_t)\end{aligned}$$

and we note that

$$\begin{aligned}\nabla_\theta \log \mathbb{P}_\theta\left(\tau\right) &= \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(s_t, a_t) \\ \nabla_\theta R(\tau) &= 0 \\ \nabla_\theta C(\tau) &= 0\end{aligned}$$

we have

$$J(\theta) + \gamma H(\theta) = \sum_\tau \mathbb{P}_{\pi_\theta}(\tau)\left(R(\tau) - \gamma \log \mathbb{P}_{\pi_\theta}(\tau) + \gamma C(\tau)\right)$$

thus

$$\begin{aligned}\nabla_\theta(J(\theta) + \gamma H(\theta)) &= \sum_\tau \nabla_\theta \mathbb{P}_{\pi_\theta}(R(\tau) + \gamma C(\tau)) - \gamma \nabla_\theta \mathbb{P}_{\pi_\theta}(\tau) \log \mathbb{P}_{\pi_\theta}(\tau) - \gamma \nabla_\theta \mathbb{P}_{\pi_\theta}(\tau) \\ &= \sum_\tau \nabla_\theta \mathbb{P}_{\pi_\theta}\left(R(\tau) - \gamma \log \mathbb{P}_{\pi_\theta}(\tau) + \gamma C(\tau)\right) - \gamma \nabla_\theta \sum_\tau \mathbb{P}_{\pi_\theta}(\tau) \\ &= \sum_\tau \nabla_\theta \mathbb{P}_{\pi_\theta}\left(R(\tau) - \gamma \log \mathbb{P}_{\pi_\theta}(\tau) + \gamma C(\tau)\right) - \gamma \nabla_\theta 1 \\ &= \sum_\tau \mathbb{P}_{\pi_\theta}(\tau) \nabla_\theta \log \mathbb{P}_{\pi_\theta}(\tau)\left(R(\tau) - \gamma \log \mathbb{P}_{\pi_\theta}(\tau) + \gamma C(\tau)\right) \\ &= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \mathbb{P}_{\pi_\theta}(\tau)\left(R(\tau) - \gamma \log \mathbb{P}_{\pi_\theta}(\tau) + \gamma C(\tau)\right)\right] \\ &= \mathbb{E}_{\pi_\theta}\left[\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(s_t, a_t)\right)\left(\sum_{t=1}^{T} r(s_t, a_t) - \gamma \log \pi_\theta(s_t, a_t)\right)\right]\end{aligned}$$

d) The gradient update at each episode is

$$\theta \leftarrow \theta + \alpha \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(s_t, a_t) \right) \left(\sum_{t=1}^{T} r(s_t, a_t) - \gamma \log \pi_\theta(s_t, a_t) \right)$$

we have in the case of soft-max policies.

$$\theta \leftarrow \theta + \alpha \left(\sum_{t=1}^{T} f(s_t, a_t) - \sum_{a' \in \mathcal{A}} f(s_t, a') \pi_\theta(s_t, a') \right) \left(\sum_{t=1}^{T} r(s_t, a_t) - \gamma \log \pi_\theta(s_t, a_t) \right)$$

e) For ease of notation let $z(s_u, a_u) = r(s_u, a_u) - \gamma \log \pi_\theta(s_u, a_u)$

$$\begin{aligned}
&\mathbb{E}\Big[\nabla_\theta \log \pi_\theta(s_t, a_t) z(s_u, a_u) \Big] \\
&\quad = \sum_{a,s,a',s'} \mathbb{P}\left(s_t = s, a_t = a, s_u = s', a_u = a'\right) z(a', s') \nabla_\theta \log \pi_\theta(s, a) \\
&\quad = \sum_{s,a,s',a'} \pi_\theta(s, a) \mathbb{P}\left(s_t = s, s_u = s', a_u = a'\right) z(a', s') \nabla_\theta \log \pi_\theta(s, a) \\
&\quad = \sum_{s,s',a'} \mathbb{P}\left(s_t = s, s_u = s', a_u = a'\right) z(a', s') \sum_a \pi_\theta(s, a) \nabla_\theta \log \pi_\theta(s, a) \\
&\quad = \sum_{s,s',a'} \mathbb{P}\left(s_t = s, s_u = s', a_u = a'\right) z(a', s') \sum_a \pi_\theta(s, a) \nabla_\theta \log \pi_\theta(s, a) \\
&\quad = \sum_{s,s',a'} \mathbb{P}\left(s_t = s, s_u = s', a_u = a'\right) z(a', s') \sum_a \nabla_\theta \pi_\theta(s, a) \\
&\quad = \sum_{s,s',a'} \mathbb{P}\left(s_t = s, s_u = s', a_u = a'\right) z(a', s') \underbrace{\nabla_\theta \big(\sum_a \pi_\theta(s, a) \big)}_{=0}
\end{aligned}$$

where in the second equality we use the markovian property (since $u < t$, so a_t only depends on s_t) and in the last equality we use the fact that $\sum_a \pi_\theta(s, a) = 1$. Now using the causality principle we obtain a new unbiased estimate of $\nabla_\theta J(\theta)$ which has less terms and thus may potentially have a much lower variance.

$$\nabla_\theta J(\theta) = \mathbb{E}\left[\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(s_t, a_t) \right) \left(\sum_{u=t}^{T} r(s_u, a_u) - \gamma \log \pi_\theta(s_u, a_u) \right) \right]$$

Thus we may simply modify the gradient update in the REINFORCE algorithm as follows

$$\theta \leftarrow \theta + \alpha \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(s_t, a_t) \right) \left(\sum_{u=t}^{T} r(s_u, a_u) - \gamma \log \pi_\theta(s_u, a_u) \right)$$

and in the case of soft-max policies we have

$$\theta \leftarrow \theta + \alpha \left(\sum_{t=1}^{T} f(s_t, a_t) - \sum_{a' \in \mathcal{A}} f(s_t, a') \pi_\theta(s_t, a') \right) \left(\sum_{u=t}^{T} r(s_u, a_u) - \gamma \log \pi_\theta(s_u, a_u) \right)$$