TÉCNICO LISBOA

Learning and Decision Making 2017-2018

MSc in Computer Science and Engineering

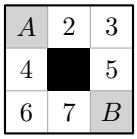
Examination – Part 1 – July 7, 2018

Instructions

- You have 90 minutes to complete this part of the examination.
- Make sure that the handout has a total of 7 pages and is not missing any sheets, then write your full name and student n. on this page (and all others if you want to be safe).
- This part has a total of 3 questions, with a maximum score of 20 points. The questions have different levels of difficulty. The point value of each question is provided next to the question number.
- *If you get stuck in a question, move on.* You should start with the easier questions to secure those points, before moving on to the harder questions.
- *No interaction with the faculty is allowed during the exam.* If you are unclear about a question, clearly indicate it and answer to the best of your ability.
- Please provide your answer in the space below each question. If you make a mess, clearly indicate your answer.
- The exam is open book and open notes. You may use a calculator, but any other type of electronic or communication equipment is not allowed.
- Good luck.

Question 1. (**3 pts.**)

Consider a surveillance robot moving in the following grid environment.



The robot moves around the environment in surveillance rounds. In each round, it must visit the rooms of interest (the shadowed cells marked with the letters A and B) *exactly once*. At each time step, the robot is able to observe with perfect certainty its current location. It must then take one of four possible actions: "up" (u), "down" (d), "left" (l) and "right" (r).

Describe the decision problem of the robot using an adequate model. Note that, as soon as a round of surveillance ends (i.e., as soon as the robot visited both rooms A and B), another surveillance round starts. You should indicate the type of model, the state-space, action-space, observation-space and a suitable cost function. You *do not* need to concern about transition or observation probability matrices.

Solution 1.

Let us start by defining the state space for the decision problem of the robot. Since the goal of the robot is to complete surveillance rounds, its actions will depend on which of the rooms of interest (A and B) it has already visited in the present round. The robot also needs to know its position. Therefore, we adopt the following representation for the state: a state $x \in \mathcal{X}$ consists of a triplet $(x_{\text{pos}}, x_A, x_B)$, where

- x_{pos} takes values in $\{A, 2, 3, \dots, 7, B\}$ and corresponds to the position of the robot in the grid;
- x_A is a bit that indicates whether room A has been visited in the current round;
- x_B is a bit that indicates whether room B has been visited in the current round.

The state space thus becomes

$$\mathcal{X} = \left\{(2, \bar{A}, \bar{B}), (2, A, \bar{B}), \dots, (7, \bar{A}, B), (7, A, B).(A, A, \bar{B}), (A, A, B), (B, \bar{A}, B), (B, A, B)\right\},$$

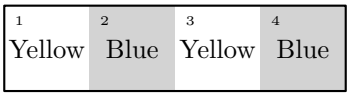
where $\bar{A}$ and $\bar{B}$ indicate that rooms A and B have not been visited in the current round, respectively, while A and B denote the opposite. We do not need to consider the triplets $(A, \bar{A}, \bar{B})$ or $(A, \bar{A}, B)$, since it is not possible for the robot to be in room A without having visited A in the current round. Similarly, we do not consider the triplets $(B, \bar{A}, \bar{B})$ or $(B, A, \bar{B})$.

The action space is directly given by $\mathcal{A} = \{u, d, l, r\}$. As for the observation space, since the robot can observe its location perfectly, we have that $\mathcal{Z} = \mathcal{X}$ and the observation probability matrices are all identities. This indicates that the model is, in fact, a (fully observable) Markov decision problem.

Finally, the cost function should translated the goal of the robot: completing surveillance rounds. Therefore, we assign a cost of 0 every time that the robot has visited both rooms of interest, and 1 otherwise, leading to:

$$c(x, a) = \begin{cases} 0 & \text{if } x \in \{(A, A, B), (B, A, B)\} \\ 1 & \text{otherwise.} \end{cases}$$

Question 2. (**12 pts.**)
Consider a robot moving in the following environment.



The environment comprises several blue (b) and yellow (y) rooms, as indicated. The robot has two actions available: left (l) and (r), and each action succeeds with probability 0.8, moving the robot in the corresponding direction. However, each action fails with a probability 0.2, leaving the position of the robot unchanged. Similarly, the robot observes the correct color of the room with a probability of 0.6 and the wrong color with a probability 0.4. The goal of the robot is to reach one of the rooms at the ends of the corridor.

The decision problem of the robot can be described as a POMDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathcal{Z}, \{\mathbf{P}_a\}, \{\mathbf{O}_a\}, c, \gamma)$, where

- $\mathcal{X} = \{1, 2, 3, 4\}$;
- $\mathcal{A} = \{l, r\}$;
- $\mathcal{Z} = \{y, b\}$;
- The transition probabilities are

$$\mathbf{P}_l = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.8 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.8 & 0.2 \end{bmatrix}; \qquad \mathbf{P}_r = \begin{bmatrix} 0.2 & 0.8 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.8 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.8 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}.$$

- The observation probabilities are

$$\mathbf{O}_l = \mathbf{O}_r = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \\ 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}.$$

- The cost function can be represented as a matrix $\mathbf{C}$ given by

$$\mathbf{C} = \begin{bmatrix} 0.0 & 0.0 \\ 1.0 & 1.0 \\ 1.0 & 1.0 \\ 0.0 & 0.0 \end{bmatrix}.$$

- Finally, the discount is given by $\gamma = 0.9$.

a) **(3 pts.)** Suppose that the initial position of the robot (at time $t = 0$) is unknown. If the agent makes the observations $\boldsymbol{z}_{1:3} = \{y, b, y\}$ after taking actions $\boldsymbol{a}_{0:2} = \{r, r, l\}$, what is its most likely initial state? Indicate all relevant computations.

b) **(3 pts.)** Show that the function

$$\mathbf{Q}_{\text{MDP}} = \begin{bmatrix} 0.00 & 0.88 \\ 1.22 & 2.10 \\ 2.10 & 1.22 \\ 0.88 & 0.00 \end{bmatrix}$$

is optimal for the underlying MDP—i.e., the MDP $(\mathcal{X}, \mathcal{A}, \{\mathbf{P}_a\}, c, \gamma)$ where $\mathcal{X}$ and $\mathcal{A}$ are as defined above, $\gamma = 0.9$ and

$$\mathbf{P}_l = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.8 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.8 & 0.2 \end{bmatrix}; \qquad \mathbf{P}_r = \begin{bmatrix} 0.2 & 0.8 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.8 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.8 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}; \qquad \mathbf{C} = \begin{bmatrix} 0.0 & 0.0 \\ 1.0 & 1.0 \\ 1.0 & 1.0 \\ 0.0 & 0.0 \end{bmatrix}.$$

c) **(2 pts.)** Consider once again the Q-function

$$\mathbf{Q}_{\text{MDP}} = \begin{bmatrix} 0.00 & 0.88 \\ 1.22 & 2.10 \\ 2.10 & 1.22 \\ 0.88 & 0.00 \end{bmatrix}.$$

Suppose that, at some time-step t, the agent's belief is given by

$$\boldsymbol{b}_t = \begin{bmatrix} 0.4 & 0.0 & 0.6 & 0.0 \end{bmatrix}.$$

Compute the POMDP action prescribed by the Q-MDP heuristic given $\boldsymbol{b}_t$.

d) **(2 pts.)** Suppose now that the optimal cost-to-go for the POMDP is represented by the α-vectors

$$\Gamma = \left\{ \begin{bmatrix} 0.00 & 1.22 & 2.29 & 2.01 \end{bmatrix}^\top, \begin{bmatrix} 2.01 & 2.29 & 1.22 & 0.00 \end{bmatrix}^\top \right\},$$

associated, respectively, to the actions l and r. Compute the optimal action given the belief $\boldsymbol{b}_t$ in c).

e) **(2 pt.)** Based on the results from c) and d), briefly discuss the merits and disadvantages of the Q-MDP heuristic.

Solution 2.

a) Computing the most likely at time step $t = 0$ is a problem of *marginal smoothing*, and we should use the forward-backward algorithm. However, since we want to compute $\mu_{0|0:3}$, the forward pass consists in computing only $\boldsymbol{\alpha}_0 = \boldsymbol{\mu}_0^\top$. We thus get

$$\boldsymbol{\alpha}_0 = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}^\top.$$

As for $\boldsymbol{\beta}_0$, we run the backward pass to get:

$$\boldsymbol{\beta}_3 = \begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ 1.00 \end{bmatrix}$$

$$\boldsymbol{\beta}_2 = \mathbf{P}_l \operatorname{diag}(\mathbf{O}_{:,y})\boldsymbol{\beta}_3 = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.8 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.8 & 0.2 \end{bmatrix} \begin{bmatrix} 0.60 \\ 0.40 \\ 0.60 \\ 0.40 \end{bmatrix} = \begin{bmatrix} 0.60 \\ 0.56 \\ 0.44 \\ 0.56 \end{bmatrix}$$

$$\boldsymbol{\beta}_1 = \mathbf{P}_r \operatorname{diag}(\mathbf{O}_{:,b})\boldsymbol{\beta}_2 = \begin{bmatrix} 0.2 & 0.8 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.8 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.8 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} 0.60 \\ 0.56 \\ 0.44 \\ 0.56 \end{bmatrix} = \begin{bmatrix} 0.32 \\ 0.21 \\ 0.30 \\ 0.34 \end{bmatrix}$$

$$\boldsymbol{\beta}_0 = \mathbf{P}_r \operatorname{diag}(\mathbf{O}_{:,y})\boldsymbol{\beta}_1 = \begin{bmatrix} 0.2 & 0.8 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.8 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.8 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} 0.32 \\ 0.21 \\ 0.30 \\ 0.34 \end{bmatrix} = \begin{bmatrix} 0.10 \\ 0.16 \\ 0.14 \\ 0.13 \end{bmatrix}.$$

Finally, we get:

$$\mu_{0|0:3} = \frac{\boldsymbol{\alpha}_0 \otimes \boldsymbol{\beta}_0}{\boldsymbol{\alpha}_0^\top \boldsymbol{\beta}_0} = \begin{bmatrix} 0.19 & 0.30 & 0.26 & 0.25 \end{bmatrix}^\top,$$

and the most likely state is $x_0 = 2$.

b) We run one step of value iteration. We have that:

$$\min_{a \in \mathcal{A}} \mathbf{Q}_{\text{MDP}}(\cdot, a) = \begin{bmatrix} 0 \\ 1.22 \\ 1.22 \\ 0 \end{bmatrix},$$

where the minimum is taken row-wise. This yields

$$\mathbf{Q}_{\text{update}}(\cdot, l) = \mathbf{C}_{:,l} + \gamma \mathbf{P}_l \min_{a \in \mathcal{A}} \mathbf{Q}_{\text{MDP}}(\cdot, a) = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + 0.9 \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.8 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.8 & 0.2 \end{bmatrix} \begin{bmatrix} 0 \\ 1.22 \\ 1.22 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1.22 \\ 2.10 \\ 0.88 \end{bmatrix},$$

$$\mathbf{Q}_{\text{update}}(\cdot, r) = \mathbf{C}_{:,r} + \gamma \mathbf{P}_r \min_{a \in \mathcal{A}} \mathbf{Q}_{\text{MDP}}(\cdot, a) = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + 0.9 \begin{bmatrix} 0.2 & 0.8 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.8 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.8 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} 0 \\ 1.22 \\ 1.22 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.88 \\ 2.10 \\ 1.22 \\ 0 \end{bmatrix}.$$

The resulting Q-function is, therefore,

$$\mathbf{Q}_{\text{update}} = \begin{bmatrix} 0.00 & 0.88 \\ 1.22 & 2.10 \\ 2.10 & 1.22 \\ 0.88 & 0.00 \end{bmatrix},$$

and the conclusion follows.

c) Computing the Q-MDP heuristic, we get:

$$\mathbf{Q}(\boldsymbol{b}_t, \cdot) = \boldsymbol{b}_t \mathbf{Q}_{\text{MDP}} = \begin{bmatrix} 1.26 & 1.08 \end{bmatrix},$$

and the action selected is $a = r$.

d) Let us denote the α-vector associated with action l as α_l and the α-vector associated with action r as α_r. We have

$$\boldsymbol{b}_t \boldsymbol{\alpha}_l = 1.38, \qquad \boldsymbol{b}_t \boldsymbol{\alpha}_r = 1.54.$$

The optimal action is, therefore, $a = l$.

e) The main merit of the Q-MDP heuristic is the fact that it computed a POMDP policy from the MDP solution, not requiring the POMDP to be solved. Additionally, unlike other simpler heuristics, it is able to take into consideration difference in value between the different actions.

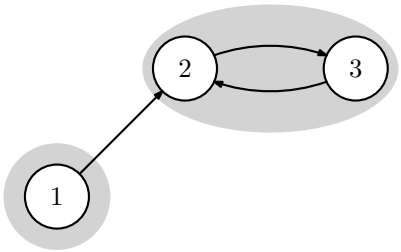
One key disadvantage of Q-MDP is the fact that it assumes that state uncertainty will disappear after the next time step, which leads Q-MDP to "ignore" information-gathering actions (since they are useless in the underlying MDP) and to often act optimistically.

As can be seen in the previous two questions, Q-MDP is optimistic in relying too much on a probability of 0.6 of being in state $x = 3$ and moving right. If uncertainty is gone, after the next step the agent will know where it is and—even if it figures out that it is in state 2, it can immediately go back to 1.

The optimal action is more cautious, and instead moves left. Then, if the next observation is "yellow", the agent will be quite certain to be in state 2 and can then move left to 1. Otherwise, most likely it is already in state 1 and action left is the best action to take.

Question 3. (**5 pts.**)

Consider the Markov chain represented below, where all transitions are deterministic.



a) **(1 pt.)** Indicate in the transition diagram the communicating classes for the chain. Is the chain irreducible? Why?

b) **(1 pt.)** Is the chain aperiodic? Why?

c) **(2 pt.)** Show that

$$\boldsymbol{\mu} = \begin{bmatrix} 0.0 & 0.5 & 0.5 \end{bmatrix}$$

is a stationary distribution for the chain.

d) **(1 pt.)** Is the chain ergodic? Why?

Solution 3.

a) The chain is not irreducible, since it has two communicating classes, indicated in the transition diagram.

b) Let us compute the period of each state in the chain.

- *State* 1*:* If $\mathrm{x}_0 = 1$, then $\mathbb{P}\left[\mathrm{x}_t = 1\right] = 0$ for all $t > 0$. In other words, if the chain departs from state 1 it will never return, so the period is infinite.
- *States* 2 *and* 3*:* The two states form a communicating class, so they have the same period. We also see that, if $\mathrm{x}_0 = 2$, $\mathbb{P}\left[\mathrm{x}_t = 2\right] > 0$ only if for even t. Therefore, the period of state 2 (and consequently 3) is 2.

It follows that the chain is not aperiodic.

c) To show that $\boldsymbol{\mu}$ is a stationary distribution, we need only to show that

$$\boldsymbol{\mu}\mathbf{P} = \boldsymbol{\mu}.$$

For the given chain, we have that

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

and

$$\boldsymbol{\mu}\mathbf{P} = \begin{bmatrix} 0.0 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0.0 & 0.5 & 0.5 \end{bmatrix}.$$

d) The chain is not ergodic. For all $t \geq 1$ the chain will oscillate between states 2 and 3 and, therefore, never converge to the stationary distribution.

TÉCNICO LISBOA

Learning and Decision Making 2017-2018

MSc in Computer Science and Engineering

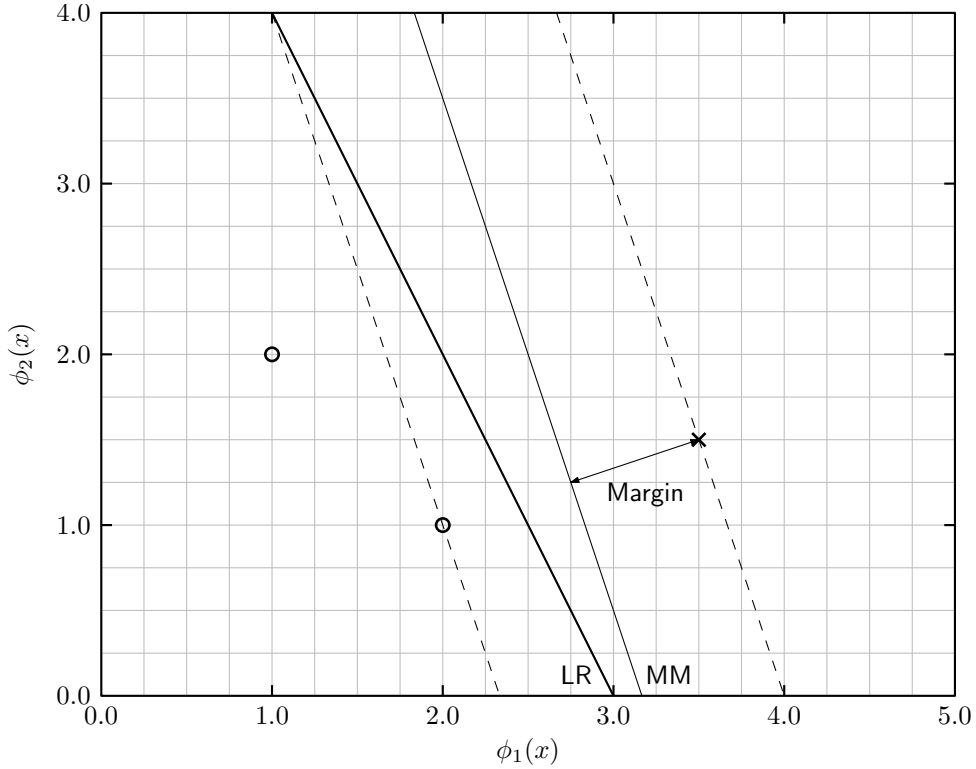
Examination – Part 2 – July 7, 2018

Instructions

- You have 90 minutes to complete this part of the examination.
- Make sure that the handout has a total of 5 pages and is not missing any sheets, then write your full name and student n. on this page (and all others if you want to be safe).
- This second part has a total of 3 questions, with a maximum score of 20 points. The questions have different levels of difficulty. The point value of each question is provided next to the question number.
- *If you get stuck in a question, move on.* You should start with the easier questions to secure those points, before moving on to the harder questions.
- *No interaction with the faculty is allowed during the exam.* If you are unclear about a question, clearly indicate it and answer to the best of your ability.
- Please provide your answer in the space below each question. If you make a mess, clearly indicate your answer.
- The exam is open book and open notes. You may use a calculator, but any other type of electronic or communication equipment is not allowed.
- Good luck.

Question 1. (**9 pts.**)

Consider the following data.



a) **(2 pts.)** Is the data linearly separable? If not, explain why. If so, draft in the plot the decision boundary for the maximum margin classifier.

b) **(2 pts.)** The logistic regression classifier is a probabilistic approach whose training consists of optimizing the parameters $\boldsymbol{w}$ that define the probability

$$\pi(a \mid x) \triangleq \mathbb{P}\left[\mathrm{a} = 1 \mid \mathrm{x} = x\right] = \frac{1}{1 + e^{-\boldsymbol{w}^\top \boldsymbol{\phi}(x)}},$$

where $\boldsymbol{\phi}(x)$ is the vector of features describing the data point x. Show that the logistic regression is a *linear classifier*.

c) **(2 pts.)** Suppose that, after some training,

$$\boldsymbol{w} \triangleq \begin{bmatrix} w_0 & w_1 & w_2 \end{bmatrix} = \begin{bmatrix} -6.0 & 2.0 & 1.0 \end{bmatrix}^\top,$$

where w_1 and w_2 are the weights associated with features ϕ_1 and ϕ_2 and w_0 is the bias term, or intercept. Compute the decision boundary for the resulting logistic regression classifier. Plot it in the diagram above.

d) **(3 pts.)** Using the weight vector $\boldsymbol{w}$ in c), compute the updated weights resulting of one iteration of gradient descent using the provided data, assuming that the algorithm is minimizing the *negative log-likelihood* of the data, i.e.,

$$\text{negative log-likelihood} = -\sum_{n=1}^{N} (a_n \log \pi(1 \mid x_n) + (1 - a_n) \log(1 - \pi(1 \mid x_n))).$$

Explicitly indicate the expression for the gradient and the relevant computations. Use a step size $\alpha = 0.1$.

Solution 1.

a) The data is linearly separable, since the dataset can be perfectly separated in two classes through a hyperplane. The maximum margin classifier is sketched in the grid and marked with "MM".

b) Being a probabilistic classifier, the decision boundary corresponds to the set of points $x \in \mathcal{X}$ such that

$$\mathbb{P}\left[\mathrm{a} = 1 \mid \mathrm{x} = x\right] = \mathbb{P}\left[\mathrm{a} = 0 \mid \mathrm{x} = x\right] = 0.5.$$

But $\mathbb{P}\left[\mathrm{a} = 1 \mid \mathrm{x} = x\right] = 0.5$ if and only if

$$1 + e^{-\boldsymbol{w}^\top \boldsymbol{\phi}(x)} = 2$$

or, equivalently,

$$e^{-\boldsymbol{w}^\top \boldsymbol{\phi}(x)} = 1.$$

Computing the logarithm on both sides finally yields

$$\boldsymbol{w}^\top \boldsymbol{\phi}(x) = 0.$$

which is the equation defining a hyperplane. The conclusion follows.

c) As seen in the previous question, the decision boundary corresponds to the set of points $x \in \mathcal{X}$ such that

$$\boldsymbol{w}^\top \boldsymbol{\phi}(x) = 0.$$

In our case, this yields

$$-6 + 2\phi_1(x) + \phi_2(x) = 0$$

or

$$\phi_2(x) = 6 - 2\phi_1(x).$$

The straight line described by such equation is plotted in the grid and marked with "LR".

d) The gradient of the negative log-likelihood is given by

$$\mathbf{grad} = \sum_{n=1}^{N} \boldsymbol{\phi}(x)(\pi(1 \mid x_n) - a_n).$$

Using the gradient above in a gradient descent update yields

$$\begin{aligned}
\boldsymbol{w}_{\text{new}} &= \boldsymbol{w} - \alpha \mathbf{grad} \\
&= \begin{bmatrix} -6.0 \\ 2.0 \\ 1.0 \end{bmatrix} - 0.1 \left(\begin{bmatrix} 1.0 \\ 2.0 \\ 1.0 \end{bmatrix} 0.119 + \begin{bmatrix} 1.0 \\ 1.0 \\ 2.0 \end{bmatrix} 0.269 - \begin{bmatrix} 1.0 \\ 3.5 \\ 1.5 \end{bmatrix} 0.076 \right) \\
&= \begin{bmatrix} -6.03 \\ 1.98 \\ 0.95 \end{bmatrix}.
\end{aligned}$$

Question 2. (**8 pts.**)

A stochastic approximation algorithm takes the general form

$$\boldsymbol{u}^{(t+1)} = \boldsymbol{u}^{(t)} + \alpha_t f(u^{(t)}, y_t),$$

where, for all $t \in \mathbb{N}$, $\boldsymbol{u}^{(t)} \in \mathbb{R}^p$, f is some real-valued function, $\{\alpha_t, t \in \mathbb{N}\}$ is a step-size sequence, and $\{y_t, t \in \mathbb{N}\}$ is a set of samples of a random variable $\mathrm{y} \in \mathbb{R}^p$ that follows some distribution P. Under mild conditions on $\{\alpha_t, t \in \mathbb{N}\}$, the function f and $\{y_t, t \in \mathbb{N}\}$, it can be shown that such algorithms converge to $\boldsymbol{u}^* \in \mathbb{R}^p$ such that $\mathbb{E}_{\mathrm{y}\sim P}\left[f(\boldsymbol{u}^*, \mathrm{y})\right] = 0$, where the subscript $\mathrm{y} \sim P$ indicates that the expectation is taken with respect to the random variable y and the distribution P.

a) **(3 pts.)** Let $\mathcal{D} = \{(x_t, a_t, c_t), t \in \mathbb{N}\}$ denote a set of transitions obtained from some MDP $(\mathcal{X}, \mathcal{A}, \{\mathbf{P}_a\}, c, \gamma)$ using a given policy π. Consider the algorithm

$$Q^{(t+1)}(x_t, a_t) = Q^{(t)}(x_t, a_t) + \alpha_t \left(c_t + \gamma \mathbb{E}_{\mathrm{a}'\sim\pi'(x_{t+1})}\left[Q^{(t)}(x_{t+1}, \mathrm{a}')\right] - Q^{(t)}(x_t, a_t)\right), \quad (1)$$

where π' is some policy (possibly different from π). Assuming that the provided algorithm converges to some function $\hat{Q}$, identify what $\hat{Q}$ will be using a stochastic approximation argument.

Suggestion: Note that, given a pair (x, a), the only random element in computing the update in (1) is the next state. You can use this fact to determine the limit of the stochastic approximation algorithm.

b) **(3 pts.)** Suppose that $\mathcal{A} = \{a, b, c, d\}$ and that π' is the uniform policy. Further assume that, after t transitions,

$$\mathbf{Q}^{(t)}_{A,:} = \begin{bmatrix} 0.25 & 0.32 & 0.32 & 0.25 \end{bmatrix}$$
$$\mathbf{Q}^{(t)}_{B,:} = \begin{bmatrix} 0.29 & 0.36 & 0.36 & 0.29 \end{bmatrix}.$$

Using the transition information

$$(x_t, a_t, c_t) = (A, a, 0.05) \qquad (x_{t+1}, a_{t+1}, c_{t+1}) = (B, b, 0.05),$$

indicate the values of $\mathbf{Q}^{(t+1)}_{A,:}$ resulting from a single update of the algorithm in (1). Use $\alpha_t = 0.1$ and $\gamma = 0.9$. Indicate all relevant computations.

c) **(2 pts.)** Is the algorithm in (1) on-policy or off-policy? Why?

Solution 2.

a) Assuming that the algorithm converges to $\hat{Q}$, we can use the provided result on the convergence of stochastic approximation by noting that the only random element in the update is the next state, x_{t+1}. Therefore, $\hat{Q}$ will be such that

$$\mathbb{E}_{\mathrm{y}\sim\mathbf{P}_a(x)}\left[c(x, a) + \gamma \mathbb{E}_{\mathrm{a}'\sim\pi'(\mathrm{y})}\left[\hat{Q}(\mathrm{y}, \mathrm{a}')\right] - \hat{Q}(x, a)\right] = 0$$

Since only the second term depends on y, we get

$$\hat{Q}(x, a) = c(x, a) + \gamma \mathbb{E}_{\mathrm{y}\sim\mathbf{P}_a(x), \mathrm{a}'\sim\pi'(\mathrm{y})}\left[\hat{Q}(\mathrm{y}, \mathrm{a}')\right],$$

which is the recursive equation verified by $Q^{\pi'}$. The algorithm will thus converge to $Q^{\pi'}$.

b) Using the provided transition information, we get

$$Q^{(t+1)}(A,a) = Q^{(t)}(A,a) + \alpha_t \left(c_t + \gamma \mathbb{E}_{\mathrm{a}' \sim \pi'(B)} \left[Q^{(t)}(B,\mathrm{a}')\right] - Q^{(t)}(A,a)\right)$$
$$= 0.25 + 0.1(0.05 + 0.9 \times 0.325 - 0.25) = 0.259,$$

resulting in

$$\mathbf{Q}^{(t)}_{A,:} = \begin{bmatrix} 0.259 & 0.32 & 0.32 & 0.25 \end{bmatrix}.$$

c) The algorithm is off-policy since it computes the Q-values for a policy π', $Q^{\pi'}$, given samples obtained from a different policy π.

Question 3. (**3 pts.**)

Let A and B represent two arbitrary multi-armed bandit algorithms, with expected regret, respectively,

$$R_T^A = c\sqrt{T}\log|\mathcal{A}|, \qquad R_T^B = cT\sqrt{\log|\mathcal{A}|},$$

where c is some positive constant. Is any of the two algorithms a *no-regret* algorithm? Justify your answer, including a definition of what a no-regret algorithm is.

Solution 3.

A no-regret algorithm is an algorithm that, asymptotically, has a zero per-step regret. In other words, if R_T is the regret of an algorithm after T time steps, a no-regret algorithm is such that

$$\lim_{T\to\infty} \frac{R_T}{T} = 0.$$

In the case of the two provided algorithms, we have that

$$\lim_{T\to\infty} \frac{R_T^A}{T} = \lim_{T\to\infty} \frac{1}{\sqrt{T}} c\log|\mathcal{A}| = 0 \qquad \lim_{T\to\infty} \frac{R_T^B}{T} = \lim_{T\to\infty} c\sqrt{\log|\mathcal{A}|} > 0.$$

Therefore, Algorithm A is a no-regret algorithm, but Algorithm B is not.