TÉCNICO LISBOA

Planning, Learning and Decision Making

MSc in Computer Science and Engineering

Repetition T1 – February 10, 2021

Instructions

- You have 90 minutes to complete the test.
- The test has a total of 5 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 answers next to each question number, within the provided square brackets. For multiple-choice questions, provide only the letter corresponding to your option.
- For open answer questions, provide your answer succinctly the square brackets. Make sure to indicate all relevant calculations. If you feel the need to use math, avoid complicated notation, and write down the names of complicated symbols. You can also use programming-like notation to indicate computations.
- **Make sure that your answer file contains only ASCII symbols.**
- The exam is open book and open notes, and you may consult any written or printed material. You can also use a calculator. The consultation or use of any other type of electronic or communication equipment during the test is not allowed.
- Good luck.

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

Pong is one of the games included in the Atari 2600 bundle in which the player controls a paddle and must avoid a ball to go off its end of the screen. In this question you will model the simplified version of *Pong* depicted in Fig. 1.

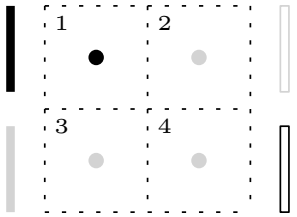


Figure 1: Simplified Pong game. The diagram depicts all posible positions for the paddles and ball.

The player controls the paddle on the right, by selecting one of two actions: "move up", or "move down". If the paddle is in the bottom position, the action "move up" moves the paddle to the top position; otherwise, the position of the paddle remains unchanged. Conversely, if the paddle is in the top position, the action "move down" moves the paddle to the bottom position; otherwise, the position of the paddle remains unchanged. The "oponent's" paddle moves randomly: at every step, it moves to the adjacent position with a probability 0.5, and remains in the same position with probability 0.5.

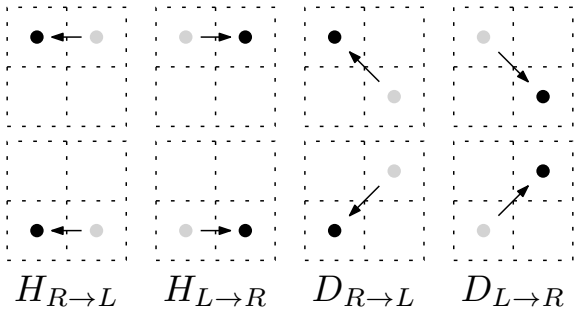


Figure 2: Possible ball movements. H and D indicate "horizontal" and "diagonal" movement, respectively. The subscript $a \to b$ indicates the direction of the movement.

The ball can be in any of the 4 numbered cells, and moves deterministically in one of four possible ways: horizontally, from left to right; horizontally, from right to left; diagonally, from left to right; or diagonally, from right to left. The four types of movement are depicted in the diagram above. If the ball is moving in a direction $A \to B$ ($A, B \in \{L, R\}$) and reaches one of the two cells on the B side of the field, one of two things may happen:

- If the paddle on that side of the field is next to the ball, the ball will "bounce back": it will remain in the same cell, but the movement will change type and direction (i.e., H changes to D and vice-versa; $R \to L$ will change to $L \to R$ and vice-versa). The agent incurs a cost of 0.1.
- If the paddle on that side of the field is *not* next to the ball, the ball will also bounce back, remaining in the same cell but changing only the direction of the movement, not the type. In that case, the agent incurs a cost of 1 if B is the agent's side of the field, or 0 otherwise.

You will now model the simplified version of the Pong game using an MDP.

(a) **(2 pts.)** Indicate the state space, action space and cost function for the game.

(b) **(1 pt.)** Suppose that the game is in the state depicted in Fig. 1 (consider the elements in black only), where the ball is moving diagonally from right to left. Indicate the transition probabilities out of this state, if the agent selects the action "move up".

Solution 1.

(a) The state space must contain information about: the players paddle (P_1), the other paddle (P_2), the position of the ball (P_b) and the movement of the ball (M_b). We thus have:

- $P_1 \in \{T, B\}$, where T corresponds to the top position and B to the bottom.
- Similarly, $P_2 \in \{T, B\}$, where T corresponds to the top position and B to the bottom.
- $P_b \in \{1, 2, 3, 4\}$, where $P_b = i$ indicates that the ball is in cell i in the grid of Fig. 1.
- $M_b \in \{H_{L\to R}, H_{R\to L}, D_{L\to R}, D_{R\to L}\}$.

The state space is thus $\mathcal{X} = \{T, B\} \times \{T, B\} \times \{1, 2, 3, 4\} \times \{H_{L\to R}, H_{R\to L}, D_{L\to R}, D_{R\to L}\}$, containing a total of 64 states.

The action space contains only the two actions "move up" and "move down", yielding $\mathcal{A} = \{U, D\}$, where U stands for the action "move up" and D stands for the action "move down".

Finally, the cost function will penalize states where the ball is in states 2 or 4, moving left to right, and the player's paddle is not next to the ball; conversely, the cost will be minimal when the ball is in states 1 or 3, moving right to left, and the opponent's paddle is not next to the ball. All other states have a cost of 0.1. Summarizing, if a state x is a tuple (P_1, P_2, P_b, M_b), we get

$$c(x, a) = \begin{cases} 1 & \text{if } P1 = T,\, P_b = 4,\, M_b \in \{H_{L\to R}, D_{L\to R}\}; \\ 1 & \text{if } P1 = B,\, P_b = 2,\, M_b \in \{H_{L\to R}, D_{L\to R}\}; \\ 0 & \text{if } P2 = T,\, P_b = 3,\, M_b \in \{H_{R\to L}, D_{R\to L}\}; \\ 0 & \text{if } P2 = B,\, P_b = 1,\, M_b \in \{H_{R\to L}, D_{R\to L}\}; \\ 0.1 & \text{otherwise.} \end{cases}$$

(b) Since the paddle is next to the ball, the ball will bounce back.Therefore, we have:

$$\mathbf{P}(y \mid (B, T, 1, D_{R\to L}), U) = \begin{cases} 0.5 & \text{if } y = (T, T, 1, H_{L\to R}) \\ 0.5 & \text{if } y = (T, B, 1, H_{L\to R}) \\ 0 & \text{otherwise.} \end{cases}$$

Question 2. (**3 pts.**)

Consider an HMM $(\mathcal{X}, \mathcal{Z}, \mathbf{P}, \mathbf{O})$ with initial distribution μ_0, and let $\boldsymbol{z}_{0:T}$ denote a sequence of T observations. For each of the following questions, select the *single* most correct answer.

(a) **(1 pt.)** The *forward mapping* is defined, for each $x \in \mathcal{X}$ and each $t = 0, \ldots, T$, as

A. $\alpha_t(x) = \mathbb{P}_{\mu_0}[\mathrm{x}_t = x \mid \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}]$.

B. $\alpha_t(x) = \mathbf{O}(z_t \mid x) \sum_{y \in \mathcal{X}} \mathbf{P}(x \mid y) \alpha_{t-1}(y)$.

C. $\beta_t(x) = \mathbb{P}_{\mu_0}[\mathbf{z}_{t+1:T} = \boldsymbol{z}_{t+1:T} \mid \mathrm{x}_t = x]$.

D. None of the above.

(b) **(1 pt.)** The *backward mapping* is defined, for each $x \in \mathcal{X}$ and each $t = 0, \ldots, T$, as

A. $\alpha_t(x) = \mathbb{P}_{\mu_0}[\mathrm{x}_t = x, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}]$.

B. $\beta_t(x) = \sum_{y \in \mathcal{X}} \mathbf{O}(z_{t+1} \mid y) \mathbf{P}(y \mid x) \beta_{t+1}(y)$.

C. $\beta_t(x) = \mathbb{P}_{\mu_0}[\mathbf{z}_{t+1:T} = \boldsymbol{z}_{t+1:T} \mid \mathrm{x}_t = x]$.

D. None of the above.

(c) **(1 pt.)** It holds that

A. $\mathbb{P}_{\mu_0}[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}] = \sum_{x \in \mathcal{X}} \alpha_t(x) \beta_t(x)$, for any $t \in \{0, \ldots, T\}$.

B. $\mu_{t|0:T}(x) = \mathbb{P}_{\mu_0}[\mathrm{x}_t = x, \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}]$

C. $\boldsymbol{\alpha}_T^\top \boldsymbol{\beta}_0 = \boldsymbol{\alpha}_0^\top \boldsymbol{\beta}_T$.

D. None of the above.

Solution 2.

(a) D (none of the above).

(b) C ($\beta_t(x) = \mathbb{P}_{\mu_0}[\mathbf{z}_{t+1:T} = \boldsymbol{z}_{t+1:T} \mid \mathrm{x}_t = x]$).

(c) A ($\mathbb{P}_{\mu_0}[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}] = \sum_{x \in \mathcal{X}} \alpha_t(x) \beta_t(x)$, for any $t \in \{0, \ldots, T\}$).

In the remainder of the test, consider the POMDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathcal{Z}, \{\mathbf{P}_a\}, \{\mathbf{O}_a\}, c, \gamma)$ where

- $\mathcal{X} = \{1, 2, 3\}$;
- $\mathcal{A} = \{a, b\}$;
- $\mathcal{Z} = \{u, v\}$;
- The transition probabilities are

$$\mathbf{P}_a = \begin{bmatrix} 0.4 & 0.6 & 0.0 \\ 0.0 & 0.4 & 0.6 \\ 0.6 & 0.0 & 0.4 \end{bmatrix}; \qquad \mathbf{P}_b = \begin{bmatrix} 0.2 & 0.0 & 0.8 \\ 0.8 & 0.2 & 0.0 \\ 0.0 & 0.8 & 0.2 \end{bmatrix}.$$

- The observation probabilities are

$$\mathbf{O}_a = \begin{bmatrix} 0.7 & 0.3 \\ 0.7 & 0.3 \\ 0.7 & 0.3 \end{bmatrix}; \qquad \mathbf{O}_b = \begin{bmatrix} 0.1 & 0.9 \\ 0.1 & 0.9 \\ 0.1 & 0.9 \end{bmatrix}.$$

- The cost function c is given by

$$\mathbf{C} = \begin{bmatrix} 0.0 & 1.0 \\ 0.8 & 1.0 \\ 0.8 & 1.0 \end{bmatrix}.$$

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

Question 3. (**7 pts.**)
Consider the MDP $\mathcal{M}_{\text{MDP}} = (\mathcal{X}, \mathcal{A}, \{\mathbf{P}_a\}, c, \gamma)$, where $\mathcal{X}$, $\mathcal{A}$, $\{\mathbf{P}_a, a \in \mathcal{A}\}$, c and γ are as defined above. Consider also the uniform random policy π.

(a) **(2 pts.)** Perform the *first iteration* necessary to compute J^π using value iteration.

(b) **(1.5 pts.)** Suppose that, for the policy π above,

$$Q^\pi = \begin{bmatrix} 6.9 & 8.0 \\ 7.8 & 7.7 \\ 7.6 & 8.0 \end{bmatrix}.$$

Compute J^π.

(c) **(1.5 pts.)** Suppose that, for the MDP $\mathcal{M}_{\text{MDP}}$, the optimal policy is given by

$$\pi^* = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

How many iterations would policy iteration take to converge, if $\pi^{(0)} = \pi$? Justify your answer.

(d) **(1 pt.)** Consider the policy π^* in (c). The transition probabilities for the Markov chain induced by π^* are (select the *single* correct option):

A. $\begin{bmatrix} 0.4 & 0.6 & 0.0 \\ 0.0 & 0.4 & 0.6 \\ 0.6 & 0.0 & 0.4 \end{bmatrix}$; B. $\begin{bmatrix} 0.4 & 0.6 \\ 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix}$; C. $\begin{bmatrix} 0.4 & 0.6 & 0.0 \\ 0.8 & 0.2 & 0.0 \\ 0.6 & 0. & 0.4 \end{bmatrix}$; D. None of the previous.

(e) **(1 pt.)** Consider once again the Markov chain induced by π^*. Select the *single* most correct option:

A. The chain is irreducible.

B. The chain is aperiodic.

C. Both A. and B. are true.

D. None of the above.

Solution 3.

(a) We have

$$\mathbf{P}_\pi = \begin{bmatrix} 0.3 & 0.3 & 0.4 \\ 0.4 & 0.3 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix}, \qquad \boldsymbol{c}_\pi = \begin{bmatrix} 0.5 \\ 0.9 \\ 0.9 \end{bmatrix}.$$

Then, since $\boldsymbol{J}^{(1)} = \boldsymbol{c}_\pi + \gamma \mathbf{P}_\pi \boldsymbol{J}^{(0)}$, we get

$$\boldsymbol{J}^{(1)} = \begin{bmatrix} 0.5 \\ 0.9 \\ 0.9 \end{bmatrix} + 0.9 \begin{bmatrix} 0.3 & 0.3 & 0.4 \\ 0.4 & 0.3 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.9 \\ 0.9 \end{bmatrix}.$$

(b) We have that

$$J^\pi(x) = \sum_{a \in \mathcal{A}} \pi(a \mid x) Q^\pi(x, a).$$

Since π is the random uniform policy, we get

$$\boldsymbol{J}^\pi = \begin{bmatrix} 7.45 \\ 7.75 \\ 7.80 \end{bmatrix}.$$

(c) We note that the greedy policy w.r.t. Q^π, provided in (b), is the same as π^*. This means that, by running policy iteration, we would get $\pi^{(1)} = \pi^*$, $\pi^{(2)} = \pi^{(1)}$, and the algorithm would stop after 2 iterations.

(d) C since the transition probabilities for the chain are

$$\mathbf{P}_{\pi^*} = \begin{bmatrix} 0.4 & 0.6 & 0.0 \\ 0.8 & 0.2 & 0.0 \\ 0.6 & 0. & 0.4 \end{bmatrix}.$$

(e) D (None of the above.)

Question 4. (**4 pts.**)
Consider the POMDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathcal{Z}, \{\mathbf{P}_a\}, \{\mathbf{O}_a\}, c, \gamma)$ defined above.

(a) **(2 pts.)** Explain the difference between the Q-MDP and FIB heuristics, in terms of what each computes, what each assumes, and the relative advantages and disadvantages.

(b) **(1 pts.)** Suppose that, for the FIB heuristic,

$$\boldsymbol{Q}_{\text{FIB}} = \begin{bmatrix} 4.9 & 5.7 \\ 5.8 & 5.5 \\ 5.3 & 6.1 \end{bmatrix}.$$

Compute the action prescribed by FIB at time step t if $\boldsymbol{b}_t = [0.1 \quad 0.1 \quad 0.8]$.

(c) **(1 pt.)** Suppose, once again, that $\boldsymbol{b}_t = [0.1 \quad 0.1 \quad 0.8]$. Further suppose that, at time step t, the agent selects the action prescribed by FIB and observes $z_{t+1} = u$. Compute $\boldsymbol{b}_{t+1}$.

Solution 4.

(a) Both Q-MDP and FIB are MDP heuristics. Q-MDP uses the optimal Q-function from the MDP, which is equivalent to assuming that partial observability will cease at the next time step and the agent will be able to follow the optimal policy for the MDP afterwards. For that reason, Q-MDP disregards partial observability from the next time step on, which causes the agent to be overly optimistic.

FIB, on the other hand, provides a tighter approximation to the POMDP Q-function that (to some extent) takes into consideration partial observability. This involves computing an alternative Q-function, Q_{FIB}, through a value-iteration-like algorithm. This algorithm is polynomial in the size of the POMDP, but more time-consuming than standard value-iteration. The advantage is that—by not discarding completely partial observability—FIB provides better performance than Q-MDP.

(b) The computation of the action for FIB is similar to that for Q-MDP, but using Q_{FIB} instead of Q^*_{MDP}. Therefore, we get

$$a_{\text{FIB}} = \operatorname{argmin} \boldsymbol{b}^\top \boldsymbol{Q}_{\text{FIB}} = \operatorname{argmin} \begin{bmatrix} 0.1 & 0.1 & 0.8 \end{bmatrix} \begin{bmatrix} 4.9 & 5.7 \\ 5.8 & 5.5 \\ 5.3 & 6.1 \end{bmatrix} = \operatorname{argmin} \begin{bmatrix} 5.3 & 6.0 \end{bmatrix} = a.$$

(c) We have

$$\boldsymbol{b}_{t+1} = \frac{1}{Z} \boldsymbol{b}_t \mathbf{P}_a \operatorname{diag}(\mathbf{O}_a(u)) = \frac{1}{Z} \begin{bmatrix} 0.1 & 0.1 & 0.8 \end{bmatrix} \begin{bmatrix} 0.4 & 0.6 & 0.0 \\ 0.0 & 0.4 & 0.6 \\ 0.6 & 0.0 & 0.4 \end{bmatrix} \times 0.7 = \frac{1}{Z} \begin{bmatrix} 0.37, 0.07, 0.27 \end{bmatrix},$$

where Z is a normalization constant. Normalizing, finally yields

$$\boldsymbol{b}_{t+1} = \begin{bmatrix} 0.52 & 0.1 & 0.38 \end{bmatrix}.$$

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

In each of the following statements, indicate whether $\succ$ is a valid preference relation and, if not, why.

(a) **(1 pt.)** Given 3 outcomes x, y, z, $x \succ y$, $y \succ z$, $x \succ z$.

(b) **(1 pt.)** Given 4 outcomes x, y, z, w, $x \succ y$, $x \succ z$, $y \succ z$, $z \succ w$, $w \succ x$.

(c) **(1 pt.)** Given 4 outcomes x, y, z, w, $x \succ y$, $x \succ z$, $z \succ y$, $z \succ w$, $w \succ x$.

Solution 5.

(a) It is a valid preference relation.

(b) It is not a valid preference relation, since it is not negative transitive. For the relation to be negative transitive, if $x \succ z$, then either $x \succ w$ or $w \succ z$. However, we see that none of the two holds, so the relation is not a valid preference.

(c) It is not a valid preference relation for the same reason. The only difference between this relation and the one in (b) is the relation between z and y. The relations between x, z, and w remain unchanged, so the relation is also not negative transitive.

TÉCNICO LISBOA

Planning, Learning and Decision Making

MSc in Computer Science and Engineering

Repetition T2 – February 10, 2021

Instructions

- You have 90 minutes to complete the test.
- The test has a total of 5 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 answers next to each question number, within the provided square brackets. For multiple-choice questions, provide only the letter corresponding to your option.
- For open answer questions, provide your answer succinctly the square brackets. Make sure to indicate all relevant calculations. If you feel the need to use math, avoid complicated notation, and write down the names of complicated symbols. You can also use programming-like notation to indicate computations.
- **Make sure that your answer file contains only ASCII symbols.**
- The exam is open book and open notes, and you may consult any written or printed material. You can also use a calculator. The consultation or use of any other type of electronic or communication equipment during the test is not allowed.
- Good luck.

Question 1. (**6 pts.**)

Consider a binary classification problem, where $\mathcal{A} = \{-1, 1\}$ and each point x is described by four binary features $\phi_1, \ldots, \phi_4$. We are given the training dataset in Table 1.

Table 1: Training dataset.

Point	$\phi_1(x_n)$	$\phi_2(x_n)$	$\phi_3(x_n)$	$\phi_4(x_n)$	a_n
x_1	0	1	0	0	1
x_2	0	0	1	0	1
x_3	0	0	1	1	1
x_4	1	0	0	1	1
x_5	0	0	1	0	-1

Suppose that we want to compute a Naïve Bayes classifier using the data in Table 1.

(a) **(2.5 pts.)** Compute the parameters of the Naïve Bayes classifier.

(b) **(2.5 pts.)** Supose now that we are given the test set in Table 2. Classify the points in the test set and compute the accuracy of the Naïve Bayes classifier.

Table 2: Test dataset.

Point	$\phi_1(x_n)$	$\phi_2(x_n)$	$\phi_3(x_n)$	$\phi_4(x_n)$	a_n
x_6	0	0	1	1	-1
x_7	1	0	0	1	-1
x_8	1	0	1	1	-1

(c) **(1 pt.)** Do you believe that your Naïve Bayes classifier is suffering from overfitting? Justify your opinion.

Solution 1.

(a) The Naïve Bayes parameters include the prior probabilities, $\mathbb{P}[\mathrm{a} = 1]$ and $\mathbb{P}[\mathrm{a} = -1]$, as well as the the class-conditional distributions $\mathbb{P}[\phi_k(\mathrm{x}) = 1 \mid \mathrm{a} = a]$, $k = 1, \ldots 4$, $a = -1, 1$. Since all features are binary, The prior probabilities immediately come:

$$\mathbb{P}[\mathrm{a} = 1] = \frac{N_1}{N} = 0.8, \qquad \mathbb{P}[\mathrm{a} = -1] = 1 - \mathbb{P}[\mathrm{a} = 1] = 0.2.$$

The conditional probabilities for class 1 become

$$\mathbb{P}[\phi_1(\mathrm{x}) = 1 \mid \mathrm{a} = 1] = \frac{N_{11}}{N_1} = 0.25, \quad \mathbb{P}[\phi_1(\mathrm{x}) = -1 \mid \mathrm{a} = 1] = 1 - \mathbb{P}[\phi_1(\mathrm{x}) = 1 \mid \mathrm{a} = 1] = 0.75,$$
$$\mathbb{P}[\phi_2(\mathrm{x}) = 1 \mid \mathrm{a} = 1] = \frac{N_{21}}{N_1} = 0.25, \quad \mathbb{P}[\phi_2(\mathrm{x}) = -1 \mid \mathrm{a} = 1] = 1 - \mathbb{P}[\phi_2(\mathrm{x}) = 1 \mid \mathrm{a} = 1] = 0.75,$$
$$\mathbb{P}[\phi_3(\mathrm{x}) = 1 \mid \mathrm{a} = 1] = \frac{N_{31}}{N_1} = 0.5, \quad \mathbb{P}[\phi_3(\mathrm{x}) = -1 \mid \mathrm{a} = 1] = 1 - \mathbb{P}[\phi_3(\mathrm{x}) = 1 \mid \mathrm{a} = 1] = 0.5,$$
$$\mathbb{P}[\phi_4(\mathrm{x}) = 1 \mid \mathrm{a} = 1] = \frac{N_{41}}{N_1} = 0.5, \quad \mathbb{P}[\phi_4(\mathrm{x}) = -1 \mid \mathrm{a} = 1] = 1 - \mathbb{P}[\phi_4(\mathrm{x}) = 1 \mid \mathrm{a} = 1] = 0.5,$$

and for class -1,

$$\mathbb{P}\left[\phi_1(\mathrm{x}) = 1 \mid \mathrm{a} = 1\right] = \frac{N_{11}}{N_1} = 0.0, \quad \mathbb{P}\left[\phi_1(\mathrm{x}) = -1 \mid \mathrm{a} = 1\right] = 1 - \mathbb{P}\left[\phi_1(\mathrm{x}) = 1 \mid \mathrm{a} = 1\right] = 1.0,$$
$$\mathbb{P}\left[\phi_2(\mathrm{x}) = 1 \mid \mathrm{a} = 1\right] = \frac{N_{21}}{N_1} = 0.0, \quad \mathbb{P}\left[\phi_2(\mathrm{x}) = -1 \mid \mathrm{a} = 1\right] = 1 - \mathbb{P}\left[\phi_2(\mathrm{x}) = 1 \mid \mathrm{a} = 1\right] = 1.0,$$
$$\mathbb{P}\left[\phi_3(\mathrm{x}) = 1 \mid \mathrm{a} = 1\right] = \frac{N_{31}}{N_1} = 1.0, \quad \mathbb{P}\left[\phi_3(\mathrm{x}) = -1 \mid \mathrm{a} = 1\right] = 1 - \mathbb{P}\left[\phi_3(\mathrm{x}) = 1 \mid \mathrm{a} = 1\right] = 0.0,$$
$$\mathbb{P}\left[\phi_4(\mathrm{x}) = 1 \mid \mathrm{a} = 1\right] = \frac{N_{41}}{N_1} = 0.0, \quad \mathbb{P}\left[\phi_4(\mathrm{x}) = -1 \mid \mathrm{a} = 1\right] = 1 - \mathbb{P}\left[\phi_4(\mathrm{x}) = 1 \mid \mathrm{a} = 1\right] = 1.0.$$

(b) The classification of each point is computed using the Naïve Bayes assumption, i.e., that the probability of each feature given the class is independent of the other features. Denoting by $f_{k,n}$ the value of $\phi_k(x_n)$, we get

$$\mathbb{P}\left[\mathrm{a}_6 = 1 \mid \boldsymbol{\phi}(x_6)\right] \propto \mathbb{P}\left[\mathrm{a}_6 = 1\right] \prod_{k=1}^{4} \mathbb{P}\left[\phi_k(x_6) = f_{k,6} \mid \mathrm{a}_6 = 1\right] = 0.1125,$$
$$\mathbb{P}\left[\mathrm{a}_7 = 1 \mid \boldsymbol{\phi}(x_7)\right] \propto \mathbb{P}\left[\mathrm{a}_7 = 1\right] \prod_{k=1}^{4} \mathbb{P}\left[\phi_k(x_7) = f_{k,7} \mid \mathrm{a}_7 = 1\right] = 0.0375,$$
$$\mathbb{P}\left[\mathrm{a}_8 = 1 \mid \boldsymbol{\phi}(x_8)\right] \propto \mathbb{P}\left[\mathrm{a}_8 = 1\right] \prod_{k=1}^{4} \mathbb{P}\left[\phi_k(x_8) = f_{k,8} \mid \mathrm{a}_8 = 1\right] = 0.0375,$$

and

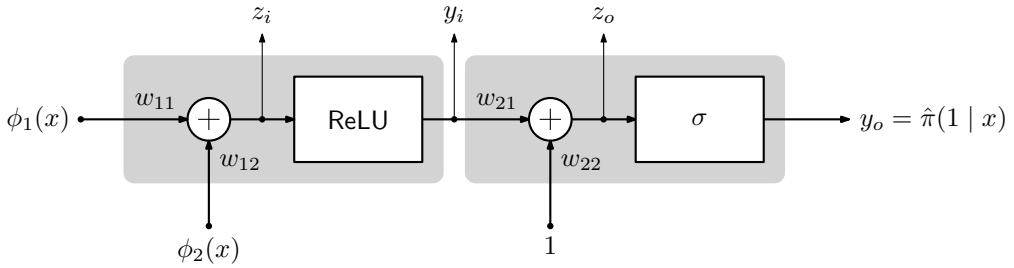
$$\mathbb{P}\left[\mathrm{a}_6 = 0 \mid \boldsymbol{\phi}(x_6)\right] \propto \mathbb{P}\left[\mathrm{a}_6 = 0\right] \prod_{k=1}^{4} \mathbb{P}\left[\phi_k(x_6) = f_{k,6} \mid \mathrm{a}_6 = 1\right] = 0.0,$$
$$\mathbb{P}\left[\mathrm{a}_7 = 0 \mid \boldsymbol{\phi}(x_7)\right] \propto \mathbb{P}\left[\mathrm{a}_7 = 0\right] \prod_{k=1}^{4} \mathbb{P}\left[\phi_k(x_7) = f_{k,7} \mid \mathrm{a}_7 = 1\right] = 0.0,$$
$$\mathbb{P}\left[\mathrm{a}_8 = 0 \mid \boldsymbol{\phi}(x_8)\right] \propto \mathbb{P}\left[\mathrm{a}_8 = 0\right] \prod_{k=1}^{4} \mathbb{P}\left[\phi_k(x_8) = f_{k,8} \mid \mathrm{a}_8 = 1\right] = 0.0.$$

Therefore, all points are classified as belonging to class 1, and the accuracy of the classifier is 0.

(c) The classifier is clearly suffering from overfitting, due to the fact that (i) the dataset is very small (5 points for 9 parameters) and, moreover, quite unbalanced—since there is only one point from class -1. This implies that the only point classified in class -1 is the one point in the training dataset.

Question 2. (**3 pts.**)

Consider the following neural network, comprising one hidden layer with a single unit and a single sigmoid output.



Note that z_i, y_i, and z_o *are not network outputs*, but merely auxiliary variables included for ease of reference.

(a) **(2.5 pts.)** Suppose that the network weights are

$$w_{11} = a, \qquad w_{12} = b, \qquad w_{21} = 1, \qquad w_{22} = -1,$$

for some constants $a, b > 0$. Compute the decision boundary for the neural network.

(b) **(0.5 pts.)** Is this neural network a linear classifier? Justify your answer.

Solution 2.

(a) The decision boundary is the space of inputs x such that $\hat{\pi}(1 \mid x) = 0.5$ or, equivalently, $z_o = 0$. Since

$$z_o = y_i - 1,$$

equating to 0 yields

$$\max\{a\phi_1(x) + b\phi_2(x), 0\} = 1.$$

Since $0 \neq 1$, the previous expression reduces to

$$a\phi_1(x) + b\phi_2(x) = 1,$$

which finally yields

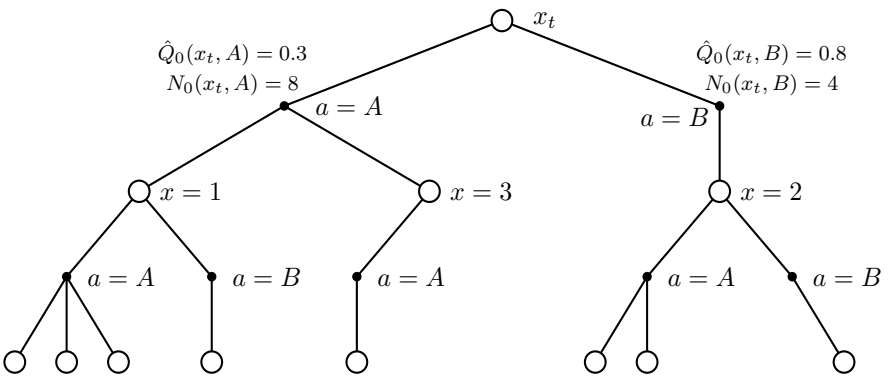
$$\phi_2(x) = -\frac{a}{b}\phi_1(x) + \frac{1}{b}.$$

(b) The neural network is a linear classifier, since the decision boundary is a hyperplane in feature space.

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

Monte-Carlo Tree Search (MCTS) is a family of online planning algorithms that can be used to solve Markov decision problems. MCTS algorithms determine the optimal action at each time step t by using "rollouts" to estimate the Q-value of the different actions available at time step t and—once planning is over—selecting the one with the smallest value. To this purpose, MCTS builds a *search tree* like the one depicted below.

In the MCTS search tree, the root node corresponds to the current state, x_t; circle notes correspond to *state nodes*, while dot notes correspond to *action nodes*. An action node a at depth d maintains an estimate $\hat{Q}_d(x, a)$ of the corresponding Q-value, where x is the corresponding parent node in the tree, as well as a counter $N_d(x, a)$ that tracks the number of times that the sequence of states-actions from the root node has been played. At each iteration of the algorithm, the tree is traversed from the root to a leaf node, where an action is selected, a new node added, and a rollout simulated.



UCT is one particular MCTS algorithm that uses the UCB heuristic to traverse the tree from the root to the leaf node to be expanded. In other words, at each state node, UCT decides which action (branch) to select using the UCB heuristic with the information available at that node.

Suppose that the tree above was built using UCT in an MDP where $\mathcal{X} = \{1, 2, 3\}$ and $\mathcal{A} = \{A, B\}$. A missing action node indicates that the corresponding action has never been selected. For example, the missing action node $a = B$ in state node $x = 3$ means that the agent never selected action B at that node. Finally, for succinctness, we show only the information stored in the nodes at level 0 of the tree.

(a) **(2 pts.)** Indicate which action UCT would select in the next iteration, at the root node? Show all relevant computations. Note that, in UCT, $\hat{Q}_0(x_t, a)$ plays the role of $\hat{c}(a)$ and $N_0(x_t, A) + N_0(x_t, B)$ plays the role of t in standard UCB.

(b) **(1 pt.)** Indicate which action UCT would select in its next visit to node $x = 3$? Justify your answer.

Solution 3.

(a) We use UCB in node x_t, and select

$$a^* = \operatorname{argmin} \hat{Q}_0(x_t, a) - \sqrt{\frac{2 \log t}{N_0(x_t, a)}},$$

where $t = N_0(x_t, A) + N_0(x_t, B)$. We get, for action A,

$$0.3 - \sqrt{\frac{2 \log(12)}{8}} = 0.3 - 0.79 = -0.49$$

and for action B,

$$0.8 - \sqrt{\frac{2 \log(12)}{4}} = 0.8 - 1.11 = -0.31.$$

The action selected would be $a = A$.

(b) At node $x = 3$, only action A has been selected. Therefore, since UCB starts by selecting each action once, the next action to be selected would be $a = B$.

Question 4. (**4 pts.**)

Expected SARSA is a variation of the standard SARSA algorithm that, given a transition (x_t, a_t, c_t, x_{t+1}), uses the update

$$\hat{Q}(x_t, a_t) \leftarrow \hat{Q}(x_t, a_t) + \alpha \left(c_t + \gamma \mathbb{E}_{\mathrm{a}_{t+1} \sim \pi(x_{t+1})} \left[\hat{Q}(x_{t+1}, \mathrm{a}_{t+1}) \right] - \hat{Q}(x_t, a_t) \right),$$

where π is the policy used by the agent to interact with the environment. Consider an MDP $(\mathcal{X}, \mathcal{A}, \{\mathbf{P}_a\}, c, \gamma)$, where $\mathcal{X} = \{1, 2, 3\}$, $\mathcal{A} = \{a, b\}$, and $\gamma = 0.9$ and suppose that, after a number of iterations of Expected SARSA,

$$\hat{Q} = \begin{bmatrix} 1.39 & 2.74 \\ 2.19 & 1.84 \\ 2.10 & 2.67 \end{bmatrix}.$$

(a) **(2.5 pts.)** Suppose that the agent observes the transition $(2, b, 1.0, 1)$. Perform one update of Expected SARSA using $\alpha = 1$, assuming that the agent is following an ε-greedy policy with $\varepsilon = 0.1$. Indicate the resulting Q-function.

(b) **(1.5 pts.)** "*One advantage of Expected SARSA is that it exhibits less variance than standard SARSA.* Do you agree with this statement? Justify your opinion.

Solution 4.

(a) Since the agent follows an ε-greedy policy, with $\varepsilon = 0.1$, we have that, in state 1,

$$\pi(\cdot \mid 1) = \begin{bmatrix} 0.95 & 0.05 \end{bmatrix}.$$

This leads to the update

$$\begin{aligned} \hat{Q}(2,b) &\leftarrow \hat{Q}(2,b) + \alpha \Big(1.0 + \gamma(0.95\, \hat{Q}(1,a) + 0.05\, \hat{Q}(1,b)) - \hat{Q}(2,b)\Big) \\ &= 1.84 + 1 \times (1 + 0.9 \times 1.46 - 1.84) \\ &= 2.31. \end{aligned}$$

The resulting Q-function is, thus,

$$\hat{Q} = \begin{bmatrix} 1.39 & 2.74 \\ 2.19 & 2.31 \\ 2.10 & 2.67 \end{bmatrix}.$$

(b) I agree with the statement. The Expected SARSA update, by computing the expected value of $\hat{Q}(x_{t+1}, \mathrm{a}_{t+1})$ over the possible next actions, a_{t+1}, eliminates the variability arising from the potential randomness introduced by the agents action selection. For this reason, the updates exhibit less variance.

Question 5. (**4 pts.**)

For each of the following statements, indicate whether it is true or false, and why.

(a) **(1 pt.)** Reinforcement learning algorithms can be used to optimally solve Markov decision problems.

(b) **(1 pt.)** In reinforcement learning with function approximation, using soft state aggregation is equivalent to introducing partial observability.

(c) **(1 pt.)** In TD(λ), the value of λ introduces a bias-variance tradeoff in the TD update: values of λ close to 1 imply larger bias but lower variance, while values of λ closer to 0 imply larger variance but smaller bias.

(d) **(1 pt.)** Policy-gradient algorithms should not be used with function approximation, since they lose their convergence properties.

Solution 5.

(a) The statement is true. RL is used to solve MDPs for which the cost function c and/or the transition probabilities $\{\mathbf{P}_a, a \in \mathcal{A}\}$ are unknown.

(b) The statement is true. Soft state aggregation groups the states into "meta-states", which depend only on the state. These meta-states lost state information and, as such, can be seen as the equivalent to observations in POMDPs. In particular, they are not Markov and are not sufficient to predict the next (meta-)state.

(c) The statement is false. Although the value of λ indeed induces a bias-variance tradeoff in the TD updates, a λ close to 1 implies smaller bias and higher variance, not the other way around. In contrast, a λ close to 0 implies larger bias and a smaller variance.

(d) The statement is false. Policy-gradient algorithms can be used with function approximation, since they are, in their essence, gradient descent algorithms. Moreover, the policy gradient theorem ensures that, even if the Q-function estimates used to perform the gradient updates is only approximate, an adequate choice of approximation architecture ensures that an unbiased gradient estimate can still be computed.