Planning, Learning and Decision Making

MSc in Computer Science and Engineering

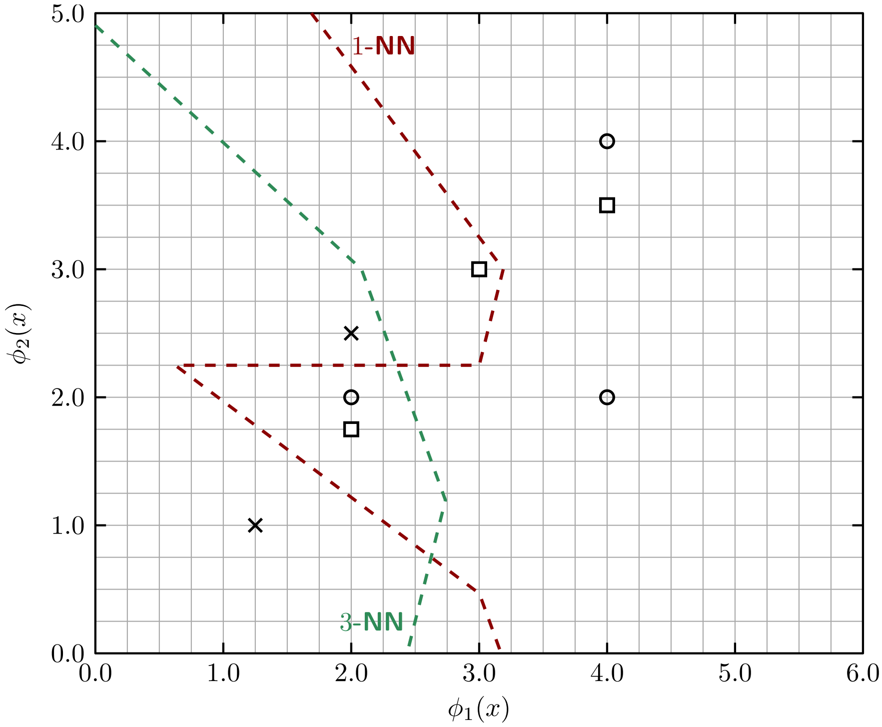
Second test – January 20, 2020

Instructions

- You have 90 minutes to complete the test.
- Make sure that your test has a total of 7 pages and is not missing any sheets, then write your full name and student n. on this page (and your number in all others).
- The test has a total of 4 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. (**7 pts.**)

Consider the dataset depicted in the grid below, where each data-point is described by two features, ϕ_1 and ϕ_2. The points marked with "$\times$" correspond to class 0 and the points marked with "$\circ$" correspond to class 1.



(a) **(2 pts.)** Indicate in the grid the decision boundary for a 1-NN classifier.

(b) **(2 pts.)** Consider now the test set

	x_1	x_2	x_3
$\phi_1(x)$	2.0	3.0	4.0
$\phi_2(x)$	1.75	3.0	3.5
Class	0	1	1

Indicate the test points in the grid, and compute the accuracy of the 1-NN classifier. You can use geometric arguments/reasoning.

(c) **(2 pts.)** Using the same test data, compute the accuracy of the 3-NN classifier. You can use geometric arguments/reasoning.

(d) **(1 pts.)** Based on the previous results, comment the following statement: *"One approach to control overfitting in the k-NN classifier is to consider larger values for k."*

Solution 1.

(a) See grid.

(b) The three test points are marked in the grid with the symbol "□". As can easily be seen,

$$\pi(x_1) = 1 \qquad \text{(correct class: 0);}$$
$$\pi(x_2) = 0 \qquad \text{(correct class: 1);}$$
$$\pi(x_3) = 1 \qquad \text{(correct class: 1).}$$

Only x_3 is correctly classified, implying that the 1-NN classifier has an accuracy of 33%, according to the provided test set.

(c) In the grid, we indicate the 3-NN boundary. It follows directly that

$$\pi(x_1) = 0 \qquad \text{(correct class: 0);}$$
$$\pi(x_2) = 1 \qquad \text{(correct class: 1);}$$
$$\pi(x_3) = 1 \qquad \text{(correct class: 1),}$$

corresponding to an accuracy of 100%.

(d) As we can observe in the boundary of the 1-NN classifier, the boundary is quite complex, seeking to "accommodate" the point in $(2.0, 2.0)$—which is most likely an outlier from class 1. This causes overfitting—for example, the 1-NN classifier classifies x_1 as belonging to class 1 precisely because of this outlier.

As we increase k, the classifier takes into consideration the label of multiple points from the training set. This makes the classification less sensitive to outliers and, therefore, less prone to overfitting. This can be seen by observing the decision boundary for the 3-NN classifier, which is significantly smoother than the previous one.

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

Consider a binary classification problem, where the input data is described by two features, ϕ_1 and ϕ_2. Suppose that you want to use the Naive Bayes classifier in this problem. According to the Naive Bayes assumption, the probability of class $a \in \{0, 1\}$ can be computed as

$$\pi(a \mid x) = \frac{p(a)p(\phi_1(x) \mid a)p(\phi_2(x) \mid a)}{Z},$$

where

- $p(a)$ is the class prior, computed from the data;
- Each $p(\phi_k(x) \mid a)$ is the class-conditional probability of $\phi_k(x)$, which corresponds to a Gaussian distribution with mean $\mu_{k,a}$ and variance $\sigma^2_{k,a}$, also computed from data;
- Z is a class-independent normalization term.

Assume that the feature variances are the same for both classes, i.e., $\sigma^2_{k,0} = \sigma_{k,1}$ for $k = 1, 2$. Show that, in this case, Naive Bayes is a *linear classifier*.

Note: It may be useful to consider that the decision boundary is such that

$$\log \pi(0 \mid x) = \log \pi(1 \mid x).$$

Also, recall that a Gaussian distribution with mean μ and variance σ^2 is given by

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Solution 2.

The NB classifier is linear if the decision boundary is given by a hyperplane (in this case, a straight line). The decision boundary for NB is given by the points x such that

$$\pi(0 \mid x) = \pi(1 \mid x)$$

or, equivalently,

$$\log \pi(0 \mid x) = \log \pi(1 \mid x).$$

Rewriting the $\pi(a \mid x)$ explicitly, we get

$$\log p(0) + \log p(\phi_1(x) \mid 0) + \log p(\phi_2(x) \mid 0) = \log p(1) + \log p(\phi_1(x) \mid 1) + \log p(\phi_2(x) \mid 1), \qquad (1)$$

where the terms involving Z have cancelled out. Denoting as σ_1^2 and σ_2^2 the (common) feature variances, we have

$$\begin{aligned}
\log p(\phi_1(x) \mid 0) &= -\log(2\pi) - \log(\sigma_1) - \frac{(\phi_1(x) - \mu_{1,0})^2}{2\sigma_1^2} \\
\log p(\phi_2(x) \mid 0) &= -\log(2\pi) - \log(\sigma_2) - \frac{(\phi_2(x) - \mu_{2,0})^2}{2\sigma_2^2} \\
\log p(\phi_1(x) \mid 1) &= -\log(2\pi) - \log(\sigma_1) - \frac{(\phi_1(x) - \mu_{1,1})^2}{2\sigma_1^2} \\
\log p(\phi_2(x) \mid 1) &= -\log(2\pi) - \log(\sigma_2) - \frac{(\phi_2(x) - \mu_{2,1})^2}{2\sigma_2^2}.
\end{aligned}$$

Replacing in (1), we get, after cancelling out duplicate terms,

$$\log p(0) - \frac{(\phi_1(x) - \mu_{1,0})^2}{2\sigma_1^2} - \frac{(\phi_2(x) - \mu_{2,0})^2}{2\sigma_2^2} = \log p(1) - \frac{(\phi_1(x) - \mu_{1,1})^2}{2\sigma_1^2} - \frac{(\phi_2(x) - \mu_{2,1})^2}{2\sigma_2^2}.$$

Expanding the squares and again cancelling out duplicate terms yields

$$\log p(0) + \frac{\mu_{1,0}}{2\sigma_1^2}(2\phi_1(x) - \mu_{1,0}) + \frac{\mu_{2,0}}{2\sigma_2^2}(2\phi_2(x) - \mu_{2,0}) = \log p(1) + \frac{\mu_{1,1}}{2\sigma_1^2}(2\phi_1(x) - \mu_{1,1}) + \frac{\mu_{2,1}}{2\sigma_2^2}(2\phi_2(x) - \mu_{2,1}),$$

which is the equation for a straight line. Thus, NB is a linear classifier.

Question 3. (**8 pts.**)

Consider a RL agent moving in an MDP $(\mathcal{X}, \mathcal{A}, \{\mathbf{P}_a\}, c, \gamma)$, where $\mathcal{A} = \{A, B\}$ and the transition probabilities $\{\mathbf{P}_a, a \in \mathcal{A}\}$ and the cost c are unknown. Consider $\gamma = 0.9$.

Further suppose that the agent follows a parameterized policy π_θ such that

$$\pi_\theta(a \mid x) = \frac{e^{\phi^\top(x,a)\theta}}{\sum_{a' \in \mathcal{A}} e^{\phi^\top(x,a')\theta}},$$

where $\phi(x, a)$ is a vector of features. At each step, the agent updates the parameters θ of the policy according to the REINFORCE algorithm. Assume that, at time step t,

$$\theta = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}^\top.$$

(a) **(2 pts.)** Suppose that, at time step $t = 2$, the agent is in some state x_t such that

$$\phi(x_t, A) = \begin{bmatrix} 0.5 \\ 0.0 \end{bmatrix}, \qquad \phi(x_t, B) = \begin{bmatrix} 0.2 \\ 0.6 \end{bmatrix}.$$

Compute $\pi(a \mid x_t)$ for $a \in \{A, B\}$.

(b) **(2 pts.)** Show that

$$\nabla_\theta \log \pi_\theta(a \mid x) = \phi(x, a) - \sum_{a' \in \mathcal{A}} \pi_\theta(a' \mid x)\phi(x, a').$$

(c) **(2 pts.)** Suppose now that, starting at time $t = 2$, the agent collects the following trajectory

$$x_2 = x, a_2 = B, c_2 = 1, \quad x_3 = x, a_3 = A, c_3 = 0.2, \quad x_4 = y, a_4 = A, c_4 = 0.2.$$

Using the trajectory above and the results from (a) and (b), perform a single REINFORCE update to θ using a step-size $\alpha = 0.3$.

Note: Recall that, given a trajectory $\{(x_t, a_t, c_t), \ldots, (x_T, a_T, c_T)\}$, REINFORCE updates the policy parameters as

$$\theta \leftarrow \theta - \alpha \nabla \log \pi_\theta(a_t \mid x_t) \sum_{\tau=t}^{T} \gamma^\tau c_\tau.$$

(d) **(2 pts.)** Explain what is the *actor-critic architecture*.

Solution 3.

(a) For the given the vector θ, e have that

$$\phi^\top(x_t, A) = 0.25, \qquad \phi^\top(x_t, B) = 0.4.$$

Then,

$$\pi(A \mid x_t) = \frac{e^{0.25}}{Z} = \frac{1.28}{Z}, \qquad \pi(B \mid x_t) = \frac{e^{0.4}}{Z} = \frac{1.49}{Z},$$

where Z is a normalization constant. In particular, $Z = 1.28 + 1.49 = 2.774$, yielding

$$\pi(A \mid x_t) = \frac{1.28}{2.774} = 0.46, \qquad \pi(B \mid x_t) = \frac{1.49}{2.774} = 0.54.$$

(b) From the policy definition,

$$\log \pi_\theta(a \mid x) = \phi^\top(x, a)\theta - \log \sum_{a' \in \mathcal{A}} e^{\phi^\top(x, a')\theta}.$$

Deriving with respect to θ immediately yields

$$\begin{aligned}
\nabla_\theta \log \pi_\theta(a \mid x) &= \phi(x, a) - \sum_{a' \in \mathcal{A}} \frac{e^{\phi^\top(x, a')\theta}\phi(x, a')}{\sum_{a'' \in \mathcal{A}} e^{\phi^\top(x, a'')\theta}} \\
&= \phi(x, a) - \sum_{a' \in \mathcal{A}} \pi(a' \mid x)\phi(x, a').
\end{aligned}$$

(c) Since, according to the trajectory provided, $a_2 = B$, using the result from (b) we get

$$\begin{aligned}\nabla_\theta \log \pi_\theta(a_t \mid x_t) &= \nabla_\theta \log \pi_\theta(B \mid x_2) \\ &= \phi(x_2, B) - \sum_{a' \in \mathcal{A}} \pi(a' \mid x_t)\phi(x_t, a').\end{aligned}$$

Using the computations from (a), yields

$$\nabla_\theta \log \pi_\theta(a_t \mid x_t) = \begin{bmatrix} 0.2 \\ 0.6 \end{bmatrix} - \begin{bmatrix} 0.5 & 0.2 \\ 0.0 & 0.6 \end{bmatrix} \begin{bmatrix} 0.46 \\ 0.54 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.6 \end{bmatrix} - \begin{bmatrix} 0.338 \\ 0.324 \end{bmatrix} = \begin{bmatrix} -0.138 \\ 0.276 \end{bmatrix}.$$

From the trajectory, we get

$$\sum_{\tau=t}^{T} \gamma^\tau c_\tau = \gamma^2 c_2 + \gamma^3 c_3 + \gamma^4 c_4 = 1.087.$$

Finally, we get

$$\theta = \theta - \alpha \nabla \log \pi_\theta(a_t \mid x_t) \sum_{\tau=t}^{T} \gamma^\tau c_\tau = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} - 0.3 \begin{bmatrix} -0.138 \\ 0.276 \end{bmatrix} 1.087 = \begin{bmatrix} 0.545 \\ 0.41 \end{bmatrix}.$$

(d) The actor-critic architecture combines two modules:

- The *actor module*, responsible for selecting the actions performed in the environment and for updating the policy using the information from the critic (usually using a policy gradient approach or a variation thereof);
- The *critic module*, uses the data collected from the environment to evaluate the current policy by computing the corresponding Q-values (or, alternatively, the advantage function).

The objective of an actor-critic architecture, when compared with a "pure" policy-gradient approach such as REINFORCE, is to decrease the variance in the estimation of the gradient observed in such "pure" PG algorithms.

Question 4. (3 pts.)

Consider an agent interacting in a sequential prediction problem, where $\mathcal{A} = \{a, b, c\}$. The agent follows the EWA algorithm and, at some time step t, $\boldsymbol{w}_t = \begin{bmatrix} 0.1 & 0.3 & 0.4 \end{bmatrix}$.

(a) **(1.5 pts.)** Compute the probability of each action a, b and c at time step t.

(b) **(1.5 pts.)** Suppose that the cost function at time step t is

$$\boldsymbol{c}_t = \begin{bmatrix} 0.3 & 0.3 & 0.0 \end{bmatrix}. \tag{2}$$

Compute the resulting weights after observing the above cost. Use $\eta = 1.0$.

Solution 4.

(a) According to EWA,

$$p_t(a) = \frac{w_t(a)}{\sum_{a' \in \mathcal{A}} w_t(a')}.$$

In our case, this leads immediately to

$$\begin{aligned} p_t(a) &= \frac{0.1}{0.8} = 0.125, \\ p_t(b) &= \frac{0.3}{0.8} = 0.375, \\ p_t(c) &= \frac{0.4}{0.8} = 0.5. \end{aligned}$$

(b) The weights in EWA are updating as

$$w_{t+1}(a) = w_t(a) e^{-\eta c_t(a)}.$$

In our case, this leads immediately to

$$\begin{aligned} w_t(a) &= 0.1 e^{-0.3} = 0.074, \\ w_t(b) &= 0.3 e^{-0.3} = 0.222, \\ w_t(c) &= 0.4 e^{-0.0} = 0.4. \end{aligned}$$