Instituto Superior Técnico

Aprendizagem 2022/21

Exam 1

November 14, 2022

I. [7v] **Regression, unsupervised learning**

Considering the following data points

	y_1	y_2	z
$\mathbf{x}_1$	0	3	5
$\mathbf{x}_2$	2	1	8
$\mathbf{x}_3$	3	1	3
$\mathbf{x}_4$	3	2	3

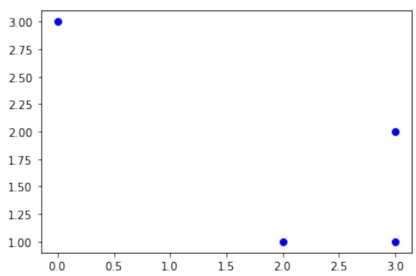
1) [2v] Considering a linear regression model, $\boldsymbol{w} = \begin{pmatrix} 2 \\ 0 \\ 0.5 \end{pmatrix}$, learnt on the data space $\phi(\mathbf{x}) = 2\mathbf{x}$, estimate its training mean absolute error.

$$\hat{\mathbf{z}} = \mathrm{X}\boldsymbol{w} = \begin{pmatrix} 1 & 0 & 6 \\ 1 & 4 & 2 \\ 1 & 6 & 2 \\ 1 & 6 & 4 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \\ 0.5 \end{pmatrix} \approx \begin{pmatrix} 5 \\ 3 \\ 3 \\ 4 \end{pmatrix}$$

$$\mathrm{MAE} = \frac{1}{4}\sum_{i=1}^{4} |z_i - \hat{z}_i| \approx \frac{0+5+0+1}{4} = \frac{3}{2}$$

2) Considering input variables (y1 and y2) only:

a) [2v] Apply k-Means using Manhattan (l_1) distance and $\{\mathbf{x}_1, \mathbf{x}_2\}$ initial centroids. Identify the centroids after each iteration. *Hint*: visualize the data space for guidance.



First iteration: $\mathbf{c}_1 = \{\mathbf{x}_1\}, \mathbf{c}_2 = \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$

$$\bar{\mathrm{c}}_1 = \begin{pmatrix} 0 \\ 3 \end{pmatrix}, \bar{\mathrm{c}}_2 = \begin{pmatrix} \frac{8}{3} \\ \frac{4}{3} \end{pmatrix}$$

Second iteration: preserved, converged.

b) [1v] For the computed solution, identify the silhouette of observation $\mathbf{x}_2$ under the same Manhattan assumption.

$$s(\mathbf{x}_2) = 1 - \frac{a(\mathbf{x}_2)}{b(\mathbf{x}_2)} = 1 - \frac{1.5}{4} = 0.625$$

c) [2v] The following eigenvectors and eigenvalues were produced from the given data:

$$v_1 = \begin{pmatrix} 0.86 \\ -0.51 \end{pmatrix}, v_2 = \begin{pmatrix} 0.51 \\ 0.86 \end{pmatrix}, \quad \lambda_1 = 2.6, \quad \lambda_2 = 0.32$$

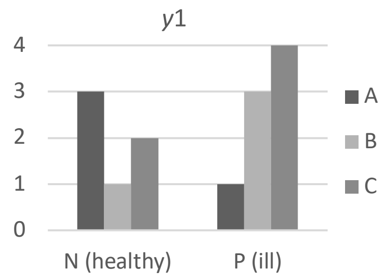
Transform the original data into the new data space by choosing one of the two criteria: i) considering the Kaiser criterion, or ii) ensuring a minimum dimensionality able to explain 95% of data variability. Identify the selected criterion in your answer.

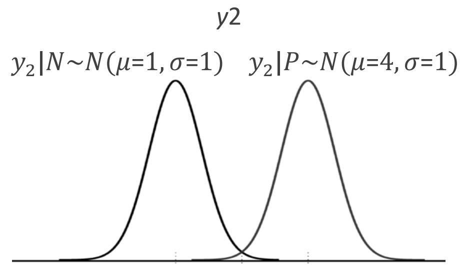
Using Kaiser criterion: only first component. Using 95% explained variance: two components.

$$X' = U^T X = \begin{pmatrix} 0.86 & 0.51 \\ -0.51 & 0.86 \end{pmatrix}^T \begin{pmatrix} 0 & 2 & 3 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} -1.53 & 1.21 & 2.07 & 1.56 \\ 2.58 & 1.88 & 2.39 & 3.25 \end{pmatrix}$$

II. [6v] **Bayes and tree learning**

Consider a dataset with a binary target and two input variables (one nominal and one numerical) with the following class-conditional frequencies:





1) [2.5v] Under a naïve Bayesian assumption, classify patient $\mathbf{x} = [\mathrm{C}, 2.5]^T$ using *MAP* estimates. Show all calculus.

$$p(z|\mathbf{x}) = \frac{p(\mathbf{x}|z) \times p(z)}{p(\mathbf{x})} \text{ where } p(\mathbf{x}|z) = p(y_1 = \mathrm{A} \mid z) p(y_2 = 2.5 \mid z)$$

$$p(N) = \frac{6}{14}, \; p(P) = \frac{8}{14}, \; p(y_1 = \mathrm{C} \mid N) = \frac{2}{6}, \; p(y_1 = \mathrm{C} \mid P) = \frac{4}{7}$$

As $p(y_2 = 2.5 \mid P) = p(y_2 = 2.5 \mid N)$ and $p(\mathbf{x})$ is invariant, we simply need to compare:

$$p(y_1 = \mathrm{C} \mid P) p(P) = \frac{4}{7} \times \frac{8}{14} = 0.33 > 0.14 = \frac{2}{6} \times \frac{6}{14} = p(y_1 = \mathrm{C} \mid N) p(N),$$

hence the patient is classified as P or ill.

2) Considering decision tree learning.

a) [1.5v] Compute the information gain of y1.

$$IG(y1) = E(z) - E(z|y1) = 0.985 - 0.857 = 0.128$$

$$E(z) = -\frac{8}{14} log \frac{8}{14} - \frac{6}{14} log \frac{6}{14} = 0.985$$

$$E(z|y1) = -\frac{6}{14} \times \left(\frac{4}{6} log \frac{4}{6} + \frac{2}{6} log \frac{2}{6}\right) - \frac{4}{14} \times \left(\frac{1}{4} log \frac{1}{4} + \frac{3}{4} log \frac{3}{4}\right) - \frac{4}{14} \times \left(\frac{1}{4} log \frac{1}{4} + \frac{3}{4} log \frac{3}{4}\right) = 0.857$$

b) [2v] Compute the confusion matrix of a decision tree given by the following two rules: $\{x_1 = A \Rightarrow P, x_1 \in \{B, C\} \Rightarrow N\}$

		True P	*True* N
Predicted	P	1	3
	N	7	3

III. [7v] **Perceptron and Neural networks**

1) [2v] Consider the linearly separable training set, $\left\{\mathbf{x}_1 = \begin{pmatrix}0\\-1\end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix}2\\2\end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix}1\\1\end{pmatrix}\right\}$, with targets $\{t_1 = N, \mathrm{t}_2 = P, \mathrm{t}_3 = P\}$. Initialize all weights to 1 (including the bias) and use a learning rate of 1. Apply *logistic regression* (sigmoid activation and cross-entropy loss) to compute the first batch gradient update and determine the separation hyperplane.

$$\Delta \mathrm{wj} = -\eta \frac{\partial \mathrm{E}}{\partial \mathrm{w}_j} = \eta \sum_{\mathrm{i}=0}^{\mathrm{n}} (\mathrm{t_i} - \hat{\mathrm{z}}_\mathrm{i}) \cdot x_j{}^{(i)}$$

$$\mathbf{w} = \mathbf{w} + \eta \times \Big(\big(t_1 - \sigma(net_1)\big)\mathbf{x}_1 + \big(t_2 - \sigma(net_2)\big)\mathbf{x}_2 + \big(t_3 - \sigma(net)\big)\mathbf{x}_3\Big)$$

$$= \begin{pmatrix}1\\1\\1\end{pmatrix} + \left(\big(0 - \sigma(0)\big)\begin{pmatrix}1\\0\\-1\end{pmatrix} + \big(1 - \sigma(5)\big)\begin{pmatrix}1\\2\\2\end{pmatrix} + \big(1 - \sigma(3)\big)\begin{pmatrix}1\\1\\1\end{pmatrix}\right) \approx \begin{pmatrix}0.56\\1.07\\1.57\end{pmatrix}$$

$$0.56 + 1.07x_1 + 1.57x_2 = 0$$

small penalization for perceptron with SSE:

$$\Delta \mathrm{wj} = \eta \sum_{\mathrm{i}=0}^{\mathrm{n}} (\mathrm{t_i} - \hat{\mathrm{z}}_\mathrm{i})\hat{\mathrm{z}}_\mathrm{i}(1 - \hat{\mathrm{z}}_\mathrm{i}) \cdot x_j{}^{(i)},$$

$$0.8775 + 1.0026x_1 + 1.1276x_2 = 0$$

2) [4v] Given the weights

$$W^{[1]} = \begin{pmatrix}1 & 1 & 1\\1 & 2 & 1\\1 & 1 & 0.4\end{pmatrix}, \quad b^{[1]} = \begin{pmatrix}0\\1\\0\end{pmatrix},$$

$$W^{[2]} = \begin{pmatrix}2 & 1 & 1\\1 & 1 & 1\end{pmatrix}, \quad b^{[2]} = \begin{pmatrix}0.2\\0\end{pmatrix},$$

the $\phi(x) = ReLU(0.4x + 0.6)$ activation function for all units/neurons, and the squared error loss,

$$E[w] = \frac{1}{2}\sum_{i=1}^{n} (\mathbf{t}_i - \mathbf{o}_i)^2.$$

Determine the new weights and biases of the *last (second) layer* only considering one stochastic gradient descent update (with learning rate of 1) using observation $\mathbf{x} = (0\ 1\ 0)^T$ and corresponding target $\mathbf{t} = (0\ 1)^T$.

$$net^{[1]} = \mathrm{W}^{[1]}\mathbf{x} + \mathrm{b}^{[1]} = \begin{pmatrix}1\\3\\1\end{pmatrix}, \quad \mathbf{x}^{[1]} = ReLU\begin{pmatrix}0.4 \times 1 + 0.6\\0.4 \times 3 + 0.6\\0.4 \times 1 + 0.6\end{pmatrix} = \begin{pmatrix}1\\1.8\\1\end{pmatrix}$$

$$net^{[2]} = \mathrm{W}^{[2]}\mathbf{x}^{[1]} + \mathrm{b}^{[2]} = \begin{pmatrix}5\\3.8\end{pmatrix}, \quad \mathbf{x}^{[2]} = ReLU\begin{pmatrix}0.4 \times 5 + 0.6\\0.4 \times 3.8 + 0.6\end{pmatrix} = \begin{pmatrix}2.6\\2.12\end{pmatrix}$$

$$\phi'(x) = \begin{cases}0.4 & \text{if } x > 0\\0 & \text{else}\end{cases}$$

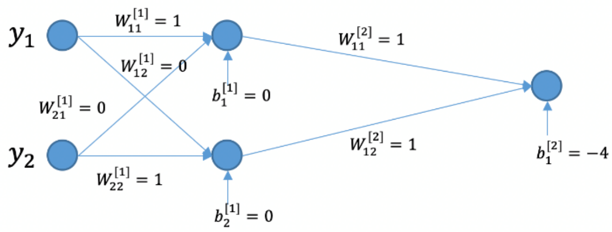
$$\delta^{[2]} = (o - t) \circ \phi'\big(\mathbf{x}^{[2]}\big) = \left(\begin{pmatrix}2.6\\2.12\end{pmatrix} - \begin{pmatrix}0\\1\end{pmatrix}\right) \circ \begin{pmatrix}0.4\\0.4\end{pmatrix} = \begin{pmatrix}1.04\\0.448\end{pmatrix}$$

$$W^{[2]} = W^{[2]} - 1\delta^{[2]}\mathbf{x}^{[1]^T} = \begin{pmatrix}2 & 1 & 1\\1 & 1 & 1\end{pmatrix} - \begin{pmatrix}1.04\\0.448\end{pmatrix}\begin{pmatrix}1 & 1.8 & 1\end{pmatrix} = \begin{pmatrix}0.96 & -0.872 & -0.04\\0.552 & 0.1936 & 0.552\end{pmatrix}$$

$$b^{[2]} = b^{[2]} - 1\delta^{[2]} = \begin{pmatrix}-0.84\\-0.448\end{pmatrix}$$

3) [1v] Consider the decision boundary given by $y_1^2 + y_2^2 = 4$.
Can a neural network with architecture 2-2-1 with net $w_0 + w_1 x_1 + w_2 x_2$ and output activation function sigmoid represent this boundary?
If so, what would be the weights, biases, and the hidden layer's activation function?

The boundary equation is given by $y_1^2 + y_2^2 = 4$ or $y_1^2 + y_2^2 - 4 = 0$
The boundary is at the place where $a^{[2]} = \sigma(net^{[2]}) = 0.5$, or, equivalently, $net^{[2]} = 0$.
To make this condition fit our problem, we must have $net^{[2]} = x_1^2 + x_2^2 - 4$
Hence $\phi^{[1]}(net^{[1]}) = (net^{[1]})^2$ and $\phi^{[2]}(net^{[2]}) = \sigma(net^{[2]})$, with the following weights and biases



END