HOMEWORK #4

APRENDIZAGEM
LEIC-A 2023/2024

Grupo #24
- Gonçalo Alves (Nº 103540)
- Daniel Nunes (Nº 103095)

I. Pen and Paper

$\{y_1\} \perp\!\!\!\perp \{y_2, y_3\}$

	y_1	y_2	y_3
x_1	1	0.6	0.1
x_2	0	-0.4	0.8
x_3	0	0.2	0.5
x_4	1	0.4	-0.1

$\pi_1 = 0.5$ $\quad$ $\pi_2 = 0.5$

$p_1 = 0.3$ $\quad$ $p_2 = 0.7$

$N_1\left(\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}2 & 0.5\\0.5 & 2\end{bmatrix}\right)$ $\quad$ $N_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1.5 & 1\\1 & 1.5\end{bmatrix}\right)$

① Para uma epoch do algoritmo EM, clustering, existem dois passos principais: o cálculo dos posteriors no E-step (Expectation) e o update dos parâmetros originais dos dois clusters no M-step (Maximization).

E-Step Expectation

Para cada observação x_i e cluster c_k, temos de calcular

$$\gamma_{ki} = p(c_k \mid x_i) = \frac{p(x_i \mid c_k)\, p(c_k)}{p(x_i)} =$$

$$p(x_i \mid c_k) = p(y_1 \mid c_k) \cdot p(y_2, y_3 \mid c_k) =$$

$$= p(y_1 \mid c_k) \cdot \mathcal{N}(y_{2,3} \mid \mu_k, \Sigma_k)$$

$$p(c_k) = \pi_k$$

$$p(x_i) = \sum_k \pi_k \cdot p(y_1 \mid c_k) \cdot \mathcal{N}(y_2, y_3 \mid \mu_k, \Sigma_k)$$

Assim, $p(C_1) = \pi_1 = 0.5$
$p(C_2) = \pi_2 = 0.5$

Para a observação x_1:

$$p(x_1 | C_1) = p(y_1 = 1 | C_1) \cdot p(q_2 = 0.6, q_3 = 0.1 | C_1)$$

$$= p_1 \cdot \mathcal{N}\left(\begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} \middle| \mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 2 \end{bmatrix}\right) =$$

$$= 0.3 \cdot \frac{1}{2\pi\sqrt{\det \begin{bmatrix} 2 & 0.5 \\ 0.5 & 2 \end{bmatrix}}} \cdot \exp\left\{ -\frac{1}{2} \left(\begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right)^T \cdot \right.$$

$$\left. \cdot \begin{bmatrix} 2 & 0.5 \\ 0.5 & 2 \end{bmatrix}^{-1} \cdot \left(\begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \right\} =$$

$$= 0.3 \cdot \frac{1}{2\pi\sqrt{3.749}} \cdot e^{-\frac{1}{2} \cdot [-0.4 \ -0.9] \begin{bmatrix} 0.533 & -0.133 \\ -0.133 & 0.533 \end{bmatrix} \begin{bmatrix} -0.4 \\ -0.9 \end{bmatrix}} =$$

$$= 0.3 \cdot 0.06658 \approx 0.01997$$

$$p(C_1 | x_1) \propto P(x_1 | C_1) P(C_1) = 0.5 \times 0.01997 = 0.00999$$

$$p(x_1 | C_2) = p(y = 1 | C_2) \cdot p(q_2 = 0.6, q_3 = 0.1 | C_2) =$$

$$= p_2 \cdot \mathcal{N}\left(\begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} \middle| \mu_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1.5 & 1 \\ 1 & 1.5 \end{bmatrix}\right) =$$

$$= 0.7 \cdot \frac{1}{2\pi\sqrt{\det \begin{bmatrix} 1.5 & 1 \\ 1 & 1.5 \end{bmatrix}}} \cdot \exp\left\{ -\frac{1}{2} \left(\begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right)^T \cdot \right.$$

$$\left. \cdot \begin{bmatrix} 1.5 & 1 \\ 1 & 1.5 \end{bmatrix}^{-1} \cdot \left(\begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right) \right\} =$$

$$= 0.7 \cdot \frac{1}{2\pi\sqrt{1.25}} \cdot e^{-\frac{1}{2} [0.6 \ 0.1] \begin{bmatrix} 1.2 & -0.8 \\ -0.8 & 1.2 \end{bmatrix} \begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix}} =$$

$= 0.7 \times 0.11962 = 0.08373$

$p(c_2|x_1) \propto p(x_1|c_2)\,p(c_2) = 0.5 \times 0.08373 = 0.04187$

$\Rightarrow \underline{p(c_1|x_1)} = \frac{0.00999}{0.00999+0.04187} = 0.19259$

$\underline{p(c_2|x_1)} = \frac{0.04187}{0.00999+0.04187} = 0.80741$

Para a observação x_2:

$$p(x_2|c_1) = p(y_1=0|c_1) \cdot \mathcal{N}\left(\begin{bmatrix}-0.4\\0.8\end{bmatrix} \middle| \mu_1 = \begin{bmatrix}1\\1\end{bmatrix}, \Sigma_1 = \begin{bmatrix}2 & 0.5\\0.5 & 2\end{bmatrix}\right) =$$

$$= (1-p_1) \cdot \frac{1}{2\pi\sqrt{3.749}}\, e^{-\frac{1}{2}\cdot\begin{bmatrix}-1.4 & -0.2\end{bmatrix}\cdot\begin{bmatrix}0.533 & -0.133\\-0.133 & 0.533\end{bmatrix}\begin{bmatrix}-1.4\\-0.2\end{bmatrix}} =$$

$$= 0.7 \cdot 0.05005 = 0.03503$$

$p(c_1|x_2) \propto p(x_2|c_1)\,p(c_1) = 0.5 \times 0.03503 = 0.01752$

$$p(x_2|c_2) = p(y_1=0|c_2) \cdot \mathcal{N}\left(\begin{bmatrix}-0.4\\0.8\end{bmatrix} \middle| \mu_2 = \begin{bmatrix}0\\0\end{bmatrix}, \Sigma_2 = \begin{bmatrix}1.5 & 1\\1 & 1.5\end{bmatrix}\right) =$$

$$= (1-p_2) \cdot \frac{1}{2\pi\sqrt{1.25}}\, e^{-\frac{1}{2}\cdot\begin{bmatrix}-0.4 & 0.8\end{bmatrix}\begin{bmatrix}1.2 & -0.8\\-0.8 & 1.2\end{bmatrix}\begin{bmatrix}-0.4\\0.8\end{bmatrix}} =$$

$$= 0.3 \cdot 0.06819 = 0.02046$$

$p(c_2|x_2) \propto p(x_2|c_2)\,p(c_2) = 0.5 \times 0.02046 = 0.01023$

$\Rightarrow \underline{p(c_1|x_2)} = \frac{0.01752}{0.01752+0.01023} = 0.63134$

$\underline{p(c_2|x_2)} = \frac{0.01023}{0.01752+0.01023} = 0.36865$

Para a observação x_3:

$$p(x_3|c_1) = p(y_1=0|c_1) \cdot \mathcal{N}\left(\begin{bmatrix}0.2\\0.5\end{bmatrix} \middle| \mu_1=\begin{bmatrix}1\\1\end{bmatrix}, \Sigma_1=\begin{bmatrix}2 & 0.5\\0.5 & 2\end{bmatrix}\right) =$$

$$= (1-p_1) \cdot \frac{1}{2\pi\sqrt{3.749}} e^{-\frac{1}{2}\begin{bmatrix}-0.8 & -0.5\end{bmatrix} \cdot \begin{bmatrix}0.533 & -0.133\\-0.133 & 0.533\end{bmatrix}\begin{bmatrix}-0.8\\-0.5\end{bmatrix}}$$

$$= 0.7 \cdot 0.06837 = 0.04786$$

$$p(c_1|x_3) \propto p(x_3|c_1)\, p(c_1) = 0.5 \times 0.04786 = 0.02393$$

$$p(x_3|c_2) = p(y_1=0|c_2) \cdot \mathcal{N}\left(\begin{bmatrix}0.2\\0.5\end{bmatrix} \middle| \mu_2=\begin{bmatrix}0\\0\end{bmatrix}, \Sigma_2=\begin{bmatrix}1.5 & 1\\1 & 1.5\end{bmatrix}\right) =$$

$$= (1-p_2) \cdot \frac{1}{2\pi\sqrt{1.25}} \cdot e^{-\frac{1}{2}\begin{bmatrix}0.2 & 0.5\end{bmatrix}\begin{bmatrix}1.2 & -0.8\\-0.8 & 1.2\end{bmatrix}\begin{bmatrix}0.2\\0.5\end{bmatrix}} =$$

$$= 0.3 \cdot 0.12958 = 0.03887$$

$$p(c_2|x_3) \propto p(x_3|c_2)\, p(c_2) = 0.5 \times 0.03887 = 0.19437$$

$$\Rightarrow p(c_1|x_3) = \frac{0.02393}{0.02393+0.19437} = 0.55181$$

$$p(c_2|x_3) = \frac{0.19437}{0.02393+0.19437} = 0.44819$$

Para a observação x_4:

$$p(x_4|c_1) = p(y_1=1|c_1) \cdot \mathcal{N}\left(\begin{bmatrix}0.4\\-0.1\end{bmatrix} \middle| \mu_1=\begin{bmatrix}1\\1\end{bmatrix}, \Sigma_1=\begin{bmatrix}2 & 0.5\\0.5 & 2\end{bmatrix}\right) =$$

$$= p_1 \cdot \frac{1}{2\pi\sqrt{3.749}} e^{-\frac{1}{2}\begin{bmatrix}-0.6 & -1.1\end{bmatrix}\begin{bmatrix}0.533 & -0.133\\-0.133 & 0.533\end{bmatrix}\begin{bmatrix}-0.6\\-1.1\end{bmatrix}} =$$

$$= 0.3 \cdot 0.05905 = 0.01771$$

$$p(c_1|x_4) \propto p(x_4|c_1)\, p(c_1) = 0.01771 \cdot 0.5 = 0.00886$$

$$p(x_4|c_2) = p(\pi_1=1|c_2) \cdot \mathcal{N}\left(\begin{bmatrix}0.4\\-0.1\end{bmatrix} \middle| \mu_2=\begin{bmatrix}0\\0\end{bmatrix}, \Sigma_2=\begin{bmatrix}1.5 & 1\\1 & 1.5\end{bmatrix}\right) =$$

$$= p_2 \cdot \frac{1}{2\pi\sqrt{1.25}} \cdot e^{-\frac{1}{2}\cdot[0.4 \;\; -0.1]\begin{bmatrix}1.5 & 1\\1 & 1.5\end{bmatrix}\begin{bmatrix}0.4\\-0.1\end{bmatrix}} =$$

$$= 0.7 \cdot 0.12450 \simeq 0.08715$$

$$p(c_2|x_4) \propto p(x_4|c_2)\,p(c_2) = 0.5 \times 0.08715 = 0.04358$$

$$\Rightarrow p(c_1|x_4) = \frac{0.00856}{0.00856+0.04358} = 0.16892$$

$$p(c_2|x_4) = \frac{0.04358}{0.00856+0.04358} = 0.83108$$

Em resumo:

	$P(c_1\|x_k)$	$P(c_2\|x_k)$
x_1	0.19259	0.80741
x_2	0.63134	0.36865
x_3	0.55181	0.44819
x_4	0.16892	0.83108

M-Step Maximization

Vamos atualizar os mixing coefficients (π_i), as probabilidades associadas às distribuições de Bernoulli e as médias e matrizes de covariância da da distribuição gaussiana multivariada, através do uso das probabilidades (posteriors) calculadas no passo anterior.

$$N_1 = \sum_i P(c_1|x_i) = 0.19259 + 0.63134 + 0.55181 + 0.16892 =$$
$$= 1.54467$$

$$N_2 = \sum_i P(c_2|x_i) = 0.80741 + 0.36865 + 0.44819 + 0.83108 =$$
$$= 2.45533$$

Os novos coeficientes de mixing serão calculados desta forma:

$$\pi_1 = \frac{N_1}{N_1 + N_2} = \frac{1.54467}{1.54467 + 2.45533} = 0.38617$$

$$\pi_2 = \frac{N_2}{N_1 + N_2} = \frac{2.45533}{1.54467 + 2.45533} = 0.61383$$

As novas probabilidades das distribuições de Bernoulli em $\{y_1\}$ serão:

$$p_1 = P(y_1 = 1) = \frac{\sum_i P(C_1 | x_i) \cdot y_{1i}}{N_1} =$$

(y_{1i} → valor de y_1 na observação x_i)

$$= \frac{0.19259 \cdot 1 + \cancel{0.63134 \cdot 0} + \cancel{0.55181 \cdot 0} + 0.16892 \cdot 1}{1.54467} = 0.23404$$

$$p_2 = P(y_1 = 1) = \frac{\sum_i P(C_2 | x_i) \cdot y_{1i}}{N_2} =$$

$$= \frac{0.80741 \cdot 1 + \cancel{0.36865 \cdot 0} + \cancel{0.44817 \cdot 0} + 0.83108}{2.45533} = 0.66732$$

Os novos parâmetros para as distribuições Gaussianas multivariadas serão:

$$\mu_1 = \frac{\sum_i P(C_1 | x_i) \cdot \begin{bmatrix} y_{2i} \\ y_{3i} \end{bmatrix}}{N_1} = \left(0.19259 \cdot \begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} + 0.63134 \cdot \begin{bmatrix} -0.4 \\ 0.8 \end{bmatrix} + \right.$$

$$\left. + 0.55181 \cdot \begin{bmatrix} 0.2 \\ 0.5 \end{bmatrix} + 0.16892 \cdot \begin{bmatrix} 0.4 \\ -0.1 \end{bmatrix}\right) / 1.54467 = \begin{bmatrix} 0.02651 \\ 0.50713 \end{bmatrix}$$

$$\Sigma_1 = \frac{\sum_i P(C_1 | x_i) \cdot \left(\begin{bmatrix} y_{2i} \\ y_{3i} \end{bmatrix} - \mu_1\right) \cdot \left(\begin{bmatrix} y_{2i} \\ y_{3i} \end{bmatrix} - \mu_1\right)^T}{N_1} =$$

$$= \left(0.19259 \cdot \left(\begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} - \begin{bmatrix} 0.02651 \\ 0.50713 \end{bmatrix}\right)\left(\begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} - \begin{bmatrix} 0.02651 \\ 0.50713 \end{bmatrix}\right)^T +\right.$$

$$+0.63134 \cdot \left(\begin{bmatrix}-0.4\\0.8\end{bmatrix}-\begin{bmatrix}0.02651\\0.50713\end{bmatrix}\right)\left(\begin{bmatrix}-0.4\\0.8\end{bmatrix}-\begin{bmatrix}0.02651\\0.50713\end{bmatrix}\right)^T +$$

$$+0.55181 \cdot \left(\begin{bmatrix}0.2\\0.5\end{bmatrix}-\begin{bmatrix}0.02651\\0.50713\end{bmatrix}\right)\left(\begin{bmatrix}0.2\\0.5\end{bmatrix}-\begin{bmatrix}0.02651\\0.50713\end{bmatrix}\right)^T +$$

$$+0.16892 \cdot \left(\begin{bmatrix}0.4\\-0.1\end{bmatrix}-\begin{bmatrix}0.02651\\0.50713\end{bmatrix}\right)\left(\begin{bmatrix}0.4\\-0.1\end{bmatrix}-\begin{bmatrix}0.02651\\0.50713\end{bmatrix}\right)^T\Big) / 1.54467 =$$

$$= \begin{bmatrix}0.14136 & -0.10540\\-0.10540 & 0.09605\end{bmatrix}$$

$$\mu_2 = \frac{\sum_i P(C_2|x_i)\cdot\begin{bmatrix}y_{2i}\\y_{3i}\end{bmatrix}}{N_2} = \Big(0.80741\cdot\begin{bmatrix}0.6\\0.1\end{bmatrix} + 0.36865\cdot\begin{bmatrix}-0.4\\0.8\end{bmatrix} +$$

$$+0.44819\cdot\begin{bmatrix}0.2\\0.5\end{bmatrix} + 0.83108\cdot\begin{bmatrix}0.4\\-0.1\end{bmatrix}\Big) / 2.45533 = \begin{bmatrix}0.30914\\0.21042\end{bmatrix}$$

$$\Sigma_2 = \frac{\sum_i P(C_2|x_i)\cdot\left(\begin{bmatrix}y_{2i}\\y_{3i}\end{bmatrix}-\mu_2\right)\cdot\left(\begin{bmatrix}y_{2i}\\y_{3i}\end{bmatrix}-\mu_2\right)^T}{N_2} =$$

$$= \Big(0.80741\cdot\left(\begin{bmatrix}0.6\\0.1\end{bmatrix}-\begin{bmatrix}0.30914\\0.21042\end{bmatrix}\right)\cdot\left(\begin{bmatrix}0.6\\0.1\end{bmatrix}-\begin{bmatrix}0.30914\\0.21042\end{bmatrix}\right)^T +$$

$$+0.36865\cdot\left(\begin{bmatrix}-0.4\\0.8\end{bmatrix}-\begin{bmatrix}0.30914\\0.21042\end{bmatrix}\right)\cdot\left(\begin{bmatrix}-0.4\\0.8\end{bmatrix}-\begin{bmatrix}0.30914\\0.21042\end{bmatrix}\right)^T +$$

$$+0.44819\cdot\left(\begin{bmatrix}0.2\\0.5\end{bmatrix}-\begin{bmatrix}0.30914\\0.21042\end{bmatrix}\right)\cdot\left(\begin{bmatrix}0.2\\0.5\end{bmatrix}-\begin{bmatrix}0.30914\\0.21042\end{bmatrix}\right)^T +$$

$$+0.83108\cdot\left(\begin{bmatrix}0.4\\0.1\end{bmatrix}-\begin{bmatrix}0.30914\\0.21042\end{bmatrix}\right)\cdot\left(\begin{bmatrix}0.4\\-0.1\end{bmatrix}-\begin{bmatrix}0.30914\\0.21042\end{bmatrix}\right)^T\Big) / 2.45533 =$$

$$= \begin{bmatrix}0.10829 & -0.08865\\-0.08865 & 0.10412\end{bmatrix}$$

Logo, a nova mixture terá os seguintes valores:

$$\pi_1 = 0.38617\,,\quad \pi_2 = 0.61383$$

$$p_1 = P(y_1 = 1) = 0.23404 \quad , \quad p_2 = P(y_1 = 1) = 0.66732$$

$$\mathcal{N}_1\left(\mu_1 = \begin{bmatrix} 0.02651 \\ 0.50713 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 0.14136 & -0.10540 \\ -0.10540 & 0.09605 \end{bmatrix}\right), \quad \mathcal{N}_2\left(\mu_2 = \begin{bmatrix} 0.30914 \\ 0.21042 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.10829 & -0.08865 \\ -0.08865 & 0.10412 \end{bmatrix}\right)$$

② $x_{new} = \begin{pmatrix} 1 \\ 0.3 \\ 0.7 \end{pmatrix}$

Vamos calcular as probabilidades $P(C_1 | x_{new})$ e $P(C_2 | x_{new})$ de modo a podermos aferir qual das classes é mais provável estar contida esta nova observação.

$$P(x_{new} | C_1) = P(y_1 = 1 | C_1) \cdot P\left(\begin{bmatrix} y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} \middle| C_1\right) =$$

$$= p_1 \cdot \mathcal{N}\left(\begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} \middle| \mu_1 = \begin{bmatrix} 0.02651 \\ 0.50713 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 0.14136 & -0.10540 \\ -0.10540 & 0.09605 \end{bmatrix}\right) =$$

$$= 0.23404 \cdot \frac{1}{2\pi\sqrt{0.00247}} e^{-\frac{1}{2}\left(\begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} - \begin{bmatrix} 0.02651 \\ 0.50713 \end{bmatrix}\right) \cdot \begin{bmatrix} 38.91660 & 42.70599 \\ 42.70599 & 57.27533 \end{bmatrix} \left(\begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} - \begin{bmatrix} 0.02651 \\ 0.50713 \end{bmatrix}\right)^T} =$$

$$= 0.23404 \cdot 0.07241 = 0.01695$$

$$\cdot P(C_1 | x_{new}) \propto P(x_{new} | C_1) P(C_1) = 0.38617 \cdot 0.01695 = 0.00654$$

$$P(x_{new} | C_2) = P(y_1 = 1 | C_2) \cdot P\left(\begin{bmatrix} y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} \middle| C_2\right) =$$

$$= p_2 \cdot \mathcal{N}\left(\begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} \middle| \mu_2 = \begin{bmatrix} 0.30914 \\ 0.21042 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.10829 & -0.08865 \\ -0.08865 & 0.10412 \end{bmatrix}\right) =$$

$$= 0.66732 \cdot \frac{1}{2\pi\sqrt{0.00342}} e^{-\frac{1}{2}\left(\begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} - \begin{bmatrix} 0.30914 \\ 0.21042 \end{bmatrix}\right) \cdot \begin{bmatrix} 30.47484 & 25.99662 \\ 25.99662 & 31.69524 \end{bmatrix} \cdot \left(\begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} - \begin{bmatrix} 0.30914 \\ 0.21042 \end{bmatrix}\right)^T} =$$

$$= 0.66732 \cdot 0.11890 = 0.07935$$

$$\cdot P(C_2 | x_{new}) \propto P(x_{new} | C_2) P(C_2) = 0.61383 \cdot 0.07935 = 0.04870$$

$$\Rightarrow P(c_1|x_{new}) = \frac{0.00654}{0.00654+0.04870} = 0.11845$$

$$P(c_2|x_{new}) = \frac{0.04870}{0.00654+0.04870} = 0.88155$$

Como $P(c_2|x_{new}) > P(c_1|x_{new})$, então podemos concluir que é bastante provável que a observação $x_{new} = [1 \ 0.3 \ 0.7]^T$ pertença ao cluster c_2.

(3) $x = \left\{ \begin{pmatrix} 1 \\ 0.6 \\ 0.1 \end{pmatrix}, \begin{pmatrix} 0 \\ -0.4 \\ 0.8 \end{pmatrix}, \begin{pmatrix} 0 \\ 0.2 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 1 \\ 0.4 \\ -0.1 \end{pmatrix} \right\}$

• atualizados

$p(c_1|x_1) = \frac{0.08939}{0.08939+0.58286} = 0.13297$

$p(c_2|x_1) = \frac{0.58286}{0.08939+0.58286} = 0.86703$ → a observação x_1 pertence ao cluster c_2

$p(c_1|x_2) = \frac{0.48902}{0.48902+0.05447} = 0.89978$

$p(c_2|x_2) = \frac{0.05447}{0.48902+0.05447} = 0.100224$ → a observação x_2 pertence ao cluster c_1

$p(c_1|x_3) = \frac{0.555352}{0.555352+0.27879} = 0.665779$

$p(c_2|x_3) = \frac{0.27879}{0.555352+0.27879} = 0.33422$ → a observação x_3 pertence ao cluster c_1

$p(c_1|x_4) = \frac{0.0080189}{0.0080189+0.44399} = 0.017740$

$p(c_2|x_4) = \frac{0.44399}{0.0080189+0.44399} = 0.982259$ → a observação x_4 pertence ao cluster c_2

- $a(i) = \frac{1}{n} \sum_{j \in C} dMan(i,j)$

- $b(i) = \min_{k \neq C} \left(\frac{1}{m} \sum_{j \in k} dMan(i,j) \right)$

• x_1 pertence ao cluster c_2, logo: $a(x_1)$ = Distância média de x_1 pra todos os outros pontos no c_2

$b(x_1)$ = Dist. média de x_1 pra todos os pontos no c_1

- $d(1,2) = |0-1| + |-0.4-0.6| + |0.8-0.1| = 2.7$
- $d(1,3) = |0-1| + |0.2-0.6| + |0.5-0.1| = 1.8$
- $d(1,4) = |1-1| + |0.4-0.6| + |-0.1-0.1| = 0.4$

$\rightarrow a(x_1) = \frac{1}{1}\left(d(1,4)\right) = 0.4$

$\rightarrow b(x_1) = \frac{1}{2}\left(d(1,2) + d(1,3)\right) = \frac{1}{2}\left(2.7 + 1.8\right) = 2.25$

$\rightarrow s(x_1) = \frac{b(x_1) - a(x_1)}{\max\{a(x_1), b(x_1)\}} = \frac{2.25 - 0.4}{2.25} \simeq 0.82$

- x_2 pertence ao cluster C_1, logo: $a(x_2)$ = Distância média de x_2 pra todos os outros pontos no C_1

  $b(x_2)$ = Dist. média de x_2 pra todos os pontos no C_2

- $d(2,1) = |1-0| + |0.6+0.4| + |0.1-0.8| = 2.7$
- $d(2,3) = |0-0| + |0.2+0.4| + |0.5-0.8| = 0.9$
- $d(2,4) = |1-0| + |0.4+0.4| + |-0.1-0.8| = 2.7$

$\rightarrow a(x_2) = \frac{1}{1}\left(d(2,3)\right) = 0.9$

$\rightarrow b(x_2) = \frac{1}{2}\left(d(2,1) + d(2,4)\right) = \frac{1}{2}\left(2.7 + 2.7\right) = 2.2$

$\rightarrow s(x_2) = \frac{b(x_2) - a(x_2)}{\max\{a(x_2), b(x_2)\}} = \frac{2.7 - 0.9}{2.7} \simeq 0.667$

- x_3 pertence ao cluster C_1, logo: $a(x_3)$ = Distância média de x_3 pra todos os outros pontos no C_1

  $b(x_3)$ = Dist. média de x_3 pra todos os pontos no C_2

- $d(3,1) = |1 - 0| + |0.6 - 0.2| + |0.1 - 0.5| = 1.8$
  $d(3,2) = |0 - 0| + |-0.4 - 0.2| + |0.8 - 0.5| = 0.9$
  $d(3,4) = |1 - 0| + |0.4 - 0.2| + |-0.1 - 0.5| = 1.8$

$\rightarrow a(x_3) = \frac{1}{1}(d(3,2)) = 0.9$

$\rightarrow b(x_3) = \frac{1}{2}(d(3,1) + d(3,4)) = \frac{1}{2}(1.8 + 1.8) = 1.8$

$\rightarrow s(x_3) = \frac{b(x_3) - a(x_3)}{\max\{a(x_3), b(x_3)\}} = \frac{1.8 - 0.9}{1.8} = \underline{0.5}$

- x_4 pertence ao cluster C_2, logo: $a(x_4)$ = Distância média de x_4 pra todos os outros pontos no C_2

  $b(x_4)$ = Dist. média de x_4 pra todos os pontos no C_1

$d(4,1) = |1 - 1| + |0.6 - 0.4| + |0.1 + 0.1| = 0.4$
$d(4,2) = |0 - 1| + |-0.4 - 0.4| + |0.8 + 0.1| = 2.7$
$d(4,3) = |0 - 1| + |0.2 - 0.4| + |0.5 + 0.1| = 1.8$

$\rightarrow a(x_4) = \frac{1}{1}(d(4,1)) = 0.4$

$\rightarrow b(x_4) = \frac{1}{2}(d(4,2) + d(4,3)) = \frac{1}{2}(2.7 + 1.8) = 2.25$

$\rightarrow s(x_4) = \frac{b(x_4) - a(x_4)}{\max\{a(x_4), b(x_4)\}} = \frac{2.25 - 0.4}{2.25} \simeq \underline{0.82}$

- Silhouette "for both clusters":
- $S(C_1) = \frac{0.667 + 0.5}{2} = 0.5835$
- $S(C_2) = \frac{0.82 + 0.82}{2} = 0.82$

(4) - $\text{purity}(C, L) = \frac{1}{n} \sum_{k=1}^{K} \max_j (|C_k \cap L_j|) =$

- $C_1 \{x_2, x_3\}$
- $C_2 \{x_1, x_4\}$

$$0.75 = \frac{1}{4} \left(\max_j (|C_1 \cap L_j|) + \max_j (|C_2 \cap L_j|) \right)$$

$$3 = \left(\max_j (|C_1 \cap L_j|) + \max_j (|C_2 \cap L_j|) \right)$$

- Temos 4 observações. Cada classe tem 1 ou mais observações, logo se tivermos 3 classes uma delas tem de ter 2 obs. e as outras 1 cada. Mas podemos também ter 2 classes em que 1 delas tem 3 obs. e a outra 1.

Homework 4 (Part II)

Aprendizagem 2023/2024 - LEIC @ IST

Group #24

- Daniel Nunes (Nº 103095)
- Gonçalo Alves (Nº 103540)

Data importing and normalization

In [ ]:
```
import pandas as pd, numpy as np
from scipy.io.arff import loadarff

original_data = loadarff('column_diagnosis.arff')
df = pd.DataFrame(original_data[0])
df['class'] = df['class'].str.decode('utf-8')

X = df.drop("class", axis=1)
y = df["class"]

display(df)
```

	pelvic_incidence	pelvic_tilt	lumbar_lordosis_angle	sacral_slope	pelvic_radius	degree_spondylol
0	63.027817	22.552586	39.609117	40.475232	98.672917	-0.
1	39.056951	10.060991	25.015378	28.995960	114.405425	4.
2	68.832021	22.218482	50.092194	46.613539	105.985135	-3
3	69.297008	24.652878	44.311238	44.644130	101.868495	1
4	49.712859	9.652075	28.317406	40.060784	108.168725	7
...	...	...	...	...	...	
305	47.903565	13.616688	36.000000	34.286877	117.449062	-4.
306	53.936748	20.721496	29.220534	33.215251	114.365845	-0
307	61.446597	22.694968	46.170347	38.751628	125.670725	-2.
308	45.252792	8.693157	41.583126	36.559635	118.545842	0
309	33.841641	5.073991	36.641233	28.767649	123.945244	-0

310 rows × 7 columns

In [ ]:
```
import numpy as np
from sklearn.preprocessing import MinMaxScaler

# Initialize the MinMaxScaler
scaler = MinMaxScaler()

# Fit and transform the data using the scaler
```

```
normalized_data = scaler.fit_transform(X)

display(normalized_data)
```

```
array([[0.35568788, 0.51989984, 0.22917997, 0.2508573 , 0.30746116,
        0.02514839],
       [0.12450104, 0.2967831 , 0.09857833, 0.14462935, 0.47664891,
        0.03636497],
       [0.41166648, 0.51393229, 0.32299466, 0.30766054, 0.38609692,
        0.0175229 ],
       ...,
       [0.34043781, 0.52244298, 0.28789745, 0.23490726, 0.59779618,
        0.01943732],
       [0.18425678, 0.27235174, 0.24684569, 0.21462279, 0.52117504,
        0.02624045],
       [0.07420202, 0.20770855, 0.20261992, 0.14251659, 0.57924032,
        0.02527676]])
```

Exercise 1

Using sklearn, apply k-means clustering fully unsupervisedly on the normalized data with $k \in \{2,3,4,5\}$ (random=0 and remaining parameters as default). Assess the silhouette and purity of the produced solutions

In [ ]:
```
from sklearn.metrics import confusion_matrix

# Define the purity function
def purity_score(y_true, y_pred):
    conf_matrix = confusion_matrix(y_true, y_pred)

    total_only_cluster = 0
    for i in range(len(conf_matrix)):
        # For each column, get its biggest value and add it onto the total
        max_column = 0
        for j in range(len(conf_matrix)):
            if(conf_matrix[j][i] > max_column):
                max_column = conf_matrix[j][i]

        total_only_cluster = total_only_cluster + max_column

    return total_only_cluster / np.sum(conf_matrix)
```

In [ ]:
```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import LabelEncoder

# Define the range of k values to try
k_values = [2, 3, 4, 5]

# Perform k-means clustering for different values of k
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=0)
    cluster_labels = kmeans.fit_predict(normalized_data)

    # For exercise 3, we will use the solution for k=3
    if (k == 3):
        k3_cluster_solution = cluster_labels.copy()
```

```
    silhouette_avg = silhouette_score(normalized_data, cluster_labels,
                                      random_state=0)
    print(f"Silhouette score for k={k}: {silhouette_avg}")

    # Assess purity
    # We need to know the ground truth labels in order to calculate purity
    le = LabelEncoder()
    ground_truth_labels_encoded = le.fit_transform(y)

    accuracy = purity_score(ground_truth_labels_encoded, cluster_labels)
    print(f"Accuracy (Purity) for k={k}: {accuracy}\n")
```

```
Silhouette score for k=2: 0.36044124340441114
Accuracy (Purity) for k=2: 0.632258064516129

Silhouette score for k=3: 0.29579055730002257
Accuracy (Purity) for k=3: 0.667741935483871

Silhouette score for k=4: 0.2744240212234018
Accuracy (Purity) for k=4: 0.6612903225806451

Silhouette score for k=5: 0.23823928397844849
Accuracy (Purity) for k=5: 0.6774193548387096

/usr/lib/python3.11/site-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: Th
e default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/usr/lib/python3.11/site-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: Th
e default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/usr/lib/python3.11/site-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: Th
e default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
/usr/lib/python3.11/site-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: Th
e default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

Exercise 2

Consider the application of PCA after the data normalization:

i. Identify the variability explained by the top two principal components.

In [ ]:

```
from sklearn.decomposition import PCA

num_top_components = 2

pca = PCA(n_components=num_top_components)
principal_components = pca.fit_transform(normalized_data)

explained_var_ratio = pca.explained_variance_ratio_
print(f"Percentage of variability explained by " +
      f"the top two principal components:")
print("Component 1:", explained_var_ratio[0] * 100, "%")
```

```
print("Component 2:", explained_var_ratio[1] * 100, "%")
```

```
Percentage of variability explained by the top two principal components:
Component 1: 56.181444842992114 %
Component 2: 20.955952591361882 %
```

ii. For each one of these two components, sort the input variables by relevance by inspecting the absolute weights of the linear projection.

In [ ]:
```
weights_abs = abs(pca.components_)

sorted_features_by_component = []
for component_idx in range(num_top_components):
    feature_weights = list(enumerate(weights_abs[component_idx]))
    sorted_features = sorted(feature_weights, key=lambda x: x[1], reverse=True)
    sorted_features_by_component.append(sorted_features)

# Print the sorted input variables for each of the top two components
features = list(df.columns.values)
for component_idx in range(num_top_components):
    print(f"Top features (Component {component_idx + 1})  Weight")
    i = 1
    for feature_idx, weight in sorted_features_by_component[component_idx]:
        feature_name = features[feature_idx]

        # Prnt the results in a neatly arranged table
        print("{}.{:<25} {:<20}".format(i, feature_name, weight))
        i = i+1
    print()
```

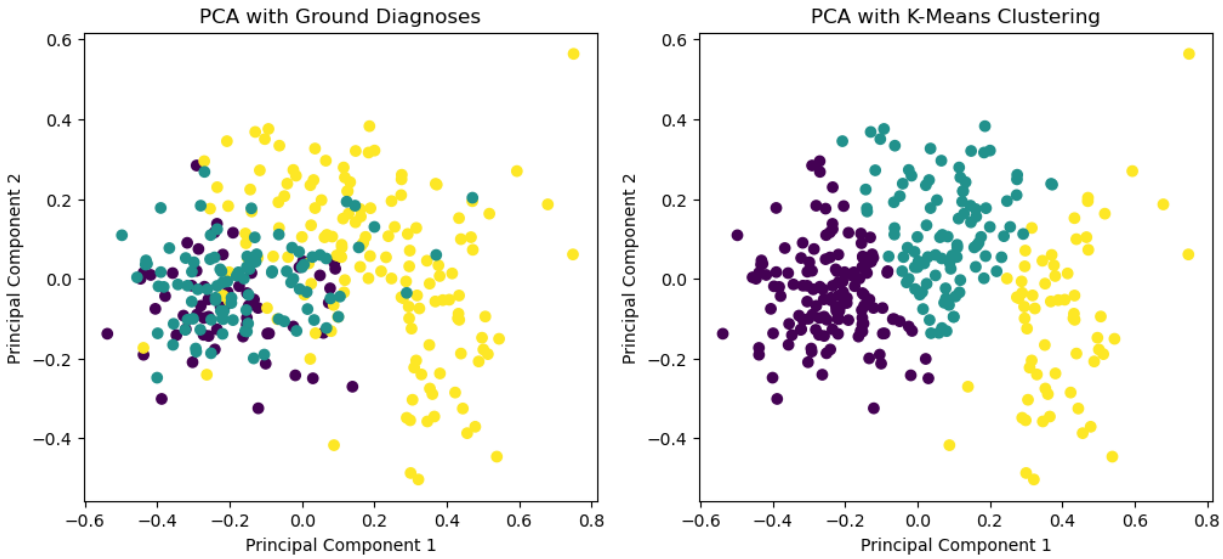
```
Top features (Component 1)  Weight
1.pelvic_incidence          0.5916206177372231
2.lumbar_lordosis_angle     0.515084762073092
3.pelvic_tilt               0.4670394389672722
4.sacral_slope              0.32568886255691937
5.degree_spondylolisthesis  0.21692963450485375
6.pelvic_radius             0.11582397626328875

Top features (Component 2)  Weight
1.pelvic_tilt               0.6703727595553639
2.pelvic_radius             0.5810738370953586
3.sacral_slope              0.44330299494707504
4.pelvic_incidence          0.10003707489152272
5.lumbar_lordosis_angle     0.08004745059088425
6.degree_spondylolisthesis  0.004582909709400146
```

Exercise 3

Visualize side-by-side the data using: i) the ground diagnoses, and ii) the previously learned k=3 clustering solution. To this end, projected the normalized data onto a 2-dimensional data space using PCA and then color observations using the reference and cluster annotations.

In [ ]:
```
import matplotlib.pyplot as plt

# Use a Label Encocoder to translate each ground diagnose (target label)
# into a color code
label_encoder = LabelEncoder()
```

```
y_numeric = label_encoder.fit_transform(y)

# Create a scatter plot for the ground diagnoses
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.scatter(principal_components[:, 0], principal_components[:, 1],
            c=y_numeric, cmap='viridis')
plt.title("PCA with Ground Diagnoses")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")

# Create a scatter plot for the cluster annotations (k-means clustering)
plt.subplot(1, 2, 2)
plt.scatter(principal_components[:, 0], principal_components[:, 1],
            c=k3_cluster_solution, cmap='viridis')
plt.title("PCA with K-Means Clustering")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")

plt.show()
```



Exercise 4

Considering the results from questions (1) and (3), identify two ways on how clustering can be used to characterize the population of ill and healthy individuals.

Our answer

Clustering can have various advantages when caracterizing ill and healthy individuals in a population.

One of them is in the early detection of diseases on patients, where we can extrapolate a set of health conditions based on the characteristics of the population of one cluster, and thus, making it faster to diagnose new patients, long before they start having symptoms.

Another one is in personalizing the treatments of different types of patients. If there is a group of patients that share similar symptoms, they're more likely to end in the same cluster, hence we can diagnose them similarly and make treatments that are more specific to their health profiles.