Aprendizagem 2023
Homework IV – Group 28

Gonçalo Bárias (ist1103124) & Raquel Braunschweig (ist1102624)

Part I: Pen and Paper

Given the following observations, $\left\{\begin{pmatrix}1\\0.6\\0.1\end{pmatrix}, \begin{pmatrix}0\\-0.4\\0.8\end{pmatrix}, \begin{pmatrix}0\\0.2\\0.5\end{pmatrix}, \begin{pmatrix}1\\0.4\\-0.1\end{pmatrix}\right\}$.

Consider a Bayesian clustering that assumes $\{y_1\} \perp\!\!\!\perp \{y_2, y_3\}$, two clusters following a Bernoulli distribution on y_1 (p_1 and p_2), a multivariate Gaussian on $\{y_2, y_3\}$ (N_1 and N_2), and the following initial mixture:

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

$$p_1 = P(y_1 = 1) = 0.3 \quad , \quad p_2 = P(y_1 = 1) = 0.7$$

$$\mathcal{N}_1\left(\boldsymbol{\mu}_1 = \begin{pmatrix}1\\1\end{pmatrix}, \boldsymbol{\Sigma}_1 = \begin{pmatrix}2 & 0.5\\0.5 & 2\end{pmatrix}\right) \quad , \quad \mathcal{N}_2\left(\boldsymbol{\mu}_2 = \begin{pmatrix}0\\0\end{pmatrix}, \boldsymbol{\Sigma}_2 = \begin{pmatrix}1.5 & 1\\1 & 1.5\end{pmatrix}\right)$$

1. **Perform one epoch of the EM clustering algorithm and determine the new parameters.**
   Hint: **we suggest you to use numpy and scipy, however disclose the intermediary results step by step.**

   The EM (Expectation-Maximization) algorithm has four major steps: Initialization, Expectation, Maximization and Evaluate.

1. Initialization

We'll start by labeling each observation:

$$x_1 = \begin{pmatrix}1\\0.6\\0.1\end{pmatrix} \quad , \quad x_2 = \begin{pmatrix}0\\-0.4\\0.8\end{pmatrix} \quad , \quad x_3 = \begin{pmatrix}0\\0.2\\0.5\end{pmatrix} \quad , \quad x_4 = \begin{pmatrix}1\\0.4\\-0.1\end{pmatrix}$$

From the statement we have the following initial parameters, p_1, p_2, μ_1, μ_2, Σ_1, Σ_2, π_1 and π_2:

Cluster	p	μ	Σ	π
Cluster 1	0.3	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}2 & 0.5\\0.5 & 2\end{pmatrix}$	0.5
Cluster 2	0.7	$\begin{pmatrix}0\\0\end{pmatrix}$	$\begin{pmatrix}1.5 & 1\\1 & 1.5\end{pmatrix}$	0.5

Table 1: Initial parameters for the two clusters

2. Expectation (E-step)

Considering $\{y_1\} \perp\!\!\!\perp \{y_2, y_3\}$ we know the posterior probability, $P(c_k|x_i)$, is given by Baye's rule:

$$P(c_k|x_i) = \frac{P(y_1, y_2, y_3|c_k)P(c_k)}{P(y_1, y_2, y_3)} = \frac{P(y_1|c_k)P(y_2, y_3|c_k)P(c_k)}{P(y_1)P(y_2, y_3)} \tag{1}$$

Since we know that $\sum_j P(c_j|x_i)$ must be equal to 1, we need to normalize the values given by equation (1). Therefore, we get these new normalized values for the posteriors represented by $\gamma_{k,i}$:

$$\gamma_{k,i} = \frac{P(c_k|x_i)}{\sum_j P(c_j|x_i)} = \frac{P(y_1|c_k)P(y_2, y_3|c_k)P(c_k)}{\sum_j P(y_1|c_j)P(y_2, y_3|c_j)P(c_j)} \tag{2}$$

The variable y_1 follows a Bernoulli distribution ($y_1 \sim \text{Bern}\,(p = p_k)$), and so the likelihoods, $P(y_1 = 0|c_k)$ and $P(y_1 = 1|c_k)$, can be calculated for each cluster:

$$\begin{aligned} P(y_1 = 0|c_1) &= 1 - p_1 = 1 - 0.3 = 0.7 \qquad & P(y_1 = 0|c_2) &= 1 - p_2 = 1 - 0.7 = 0.3 \\ P(y_1 = 1|c_1) &= p_1 = 0.3 \qquad & P(y_1 = 1|c_2) &= p_2 = 0.7 \end{aligned}$$

We know the likelihood, $P(y_2, y_3|c_k)$, follows a multivariate Gaussian, and so it is given by (considering $d = 2$, since we are working in two dimensions):

$$P(y_2 = a, y_3 = b|c_k) = \mathcal{N}_k(y_2 = a, y_3 = b|\mu_k, \Sigma_k) = \frac{\exp\left(-\frac{1}{2}\left(\begin{bmatrix} a \\ b \end{bmatrix} - \mu_k\right)^T \Sigma_k^{-1} \left(\begin{bmatrix} a \\ b \end{bmatrix} - \mu_k\right)\right)}{(2\pi)^{d/2} \times |\Sigma_k|^{1/2}} \tag{3}$$

We now have all the building blocks to calculate the posterior probabilities for each combination of observation, x_i and cluster, c_k.

We'll start off by calculating the multivariate likelihood by employing equation (3), for each pair of observation and cluster:

Cluster 1 Multivariate Likelihoods

$$P(y_2 = 0.6, y_3 = 0.1|c_1) = \mathcal{N}_1(y_2 = 0.6, y_3 = 0.1|\mu_1, \Sigma_1) \approx 0.06658$$

$$P(y_2 = -0.4, y_3 = 0.8|c_1) = \mathcal{N}_1(y_2 = -0.4, y_3 = 0.8|\mu_1, \Sigma_1) \approx 0.05005$$

$$P(y_2 = 0.2, y_3 = 0.5|c_1) = \mathcal{N}_1(y_2 = 0.2, y_3 = 0.5|\mu_1, \Sigma_1) \approx 0.06837$$

$$P(y_2 = 0.4, y_3 = -0.1|c_1) = \mathcal{N}_1(y_2 = 0.4, y_3 = -0.1|\mu_1, \Sigma_1) \approx 0.05905$$

Cluster 2 Multivariate Likelihood

$$P(y_2 = 0.6, y_3 = 0.1|c_2) = \mathcal{N}_2(y_2 = 0.6, y_3 = 0.1|\mu_2, \Sigma_2) \approx 0.11962$$

$$P(y_2 = -0.4, y_3 = 0.8|c_2) = \mathcal{N}_2(y_2 = -0.4, y_3 = 0.8|\mu_2, \Sigma_2) \approx 0.06819$$

$$P(y_2 = 0.2, y_3 = 0.5|c_2) = \mathcal{N}_2(y_2 = 0.2, y_3 = 0.5|\mu_2, \Sigma_2) \approx 0.12958$$

$$P(y_2 = 0.4, y_3 = -0.1|c_2) = \mathcal{N}_2(y_2 = 0.4, y_3 = -0.1|\mu_2, \Sigma_2) \approx 0.12450$$

Finally, we can employ equation (2) to calculate the normalized posteriors with the previously calculated values, for each pair of observation and cluster:

Cluster 1 Posteriors

$$\gamma_{1,1} = \frac{P(y_1 = 1|c_1)P(y_2 = 0.6, y_3 = 0.1|c_1)P(c_1)}{P(y_1 = 1|c_1)P(y_2 = 0.6, y_3 = 0.1|c_1)P(c_1) + P(y_1 = 1|c_2)P(y_2 = 0.6, y_3 = 0.1|c_2)P(c_2)}$$

$$= \frac{0.3 \times 0.06658 \times 0.5}{0.3 \times 0.06658 \times 0.5 + 0.7 \times 0.11962 \times 0.5} \approx 0.19259$$

$$\gamma_{1,2} = \frac{P(y_1 = 0|c_1)P(y_2 = -0.4, y_3 = 0.8|c_1)P(c_1)}{P(y_1 = 0|c_1)P(y_2 = -0.4, y_3 = 0.8|c_1)P(c_1) + P(y_1 = 0|c_2)P(y_2 = -0.4, y_3 = 0.8|c_2)P(c_2)}$$

$$= \frac{0.7 \times 0.05005 \times 0.5}{0.7 \times 0.05005 \times 0.5 + 0.3 \times 0.06819 \times 0.5} \approx 0.63135$$

$$\gamma_{1,3} = \frac{P(y_1 = 0|c_1)P(y_2 = 0.2, y_3 = 0.5|c_1)P(c_1)}{P(y_1 = 0|c_1)P(y_2 = 0.2, y_3 = 0.5|c_1)P(c_1) + P(y_1 = 0|c_2)P(y_2 = 0.2, y_3 = 0.5|c_2)P(c_2)}$$

$$= \frac{0.7 \times 0.06837 \times 0.5}{0.7 \times 0.06837 \times 0.5 + 0.3 \times 0.12958 \times 0.5} \approx 0.55181$$

$$\gamma_{1,4} = \frac{P(y_1 = 1|c_1)P(y_2 = 0.4, y_3 = -0.1|c_1)P(c_1)}{P(y_1 = 1|c_1)P(y_2 = 0.4, y_3 = -0.1|c_1)P(c_1) + P(y_1 = 1|c_2)P(y_2 = 0.4, y_3 = -0.1|c_2)P(c_2)}$$

$$= \frac{0.3 \times 0.05905 \times 0.5}{0.3 \times 0.05905 \times 0.5 + 0.7 \times 0.12450 \times 0.5} \approx 0.16892$$

Cluster 2 Posteriors

$$\gamma_{2,1} = \frac{P(y_1 = 1|c_2)P(y_2 = 0.6, y_3 = 0.1|c_2)P(c_2)}{P(y_1 = 1|c_1)P(y_2 = 0.6, y_3 = 0.1|c_1)P(c_1) + P(y_1 = 1|c_2)P(y_2 = 0.6, y_3 = 0.1|c_2)P(c_2)}$$

$$= \frac{0.7 \times 0.11962 \times 0.5}{0.3 \times 0.06658 \times 0.5 + 0.7 \times 0.11962 \times 0.5} \approx 0.80741$$

$$\gamma_{2,2} = \frac{P(y_1 = 0|c_2)P(y_2 = -0.4, y_3 = 0.8|c_2)P(c_2)}{P(y_1 = 0|c_1)P(y_2 = -0.4, y_3 = 0.8|c_1)P(c_1) + P(y_1 = 0|c_2)P(y_2 = -0.4, y_3 = 0.8|c_2)P(c_2)}$$

$$= \frac{0.3 \times 0.06819 \times 0.5}{0.7 \times 0.05005 \times 0.5 + 0.3 \times 0.06819 \times 0.5} \approx 0.36865$$

$$\gamma_{2,3} = \frac{P(y_1 = 0|c_2)P(y_2 = 0.2, y_3 = 0.5|c_2)P(c_2)}{P(y_1 = 0|c_1)P(y_2 = 0.2, y_3 = 0.5|c_1)P(c_1) + P(y_1 = 0|c_2)P(y_2 = 0.2, y_3 = 0.5|c_2)P(c_2)}$$

$$= \frac{0.3 \times 0.12958 \times 0.5}{0.7 \times 0.06837 \times 0.5 + 0.3 \times 0.12958 \times 0.5} \approx 0.44819$$

$$\gamma_{2,4} = \frac{P(y_1 = 1|c_2)P(y_2 = 0.4, y_3 = -0.1|c_2)P(c_2)}{P(y_1 = 1|c_1)P(y_2 = 0.4, y_3 = -0.1|c_1)P(c_1) + P(y_1 = 1|c_2)P(y_2 = 0.4, y_3 = -0.1|c_2)P(c_2)}$$

$$= \frac{0.7 \times 0.12450 \times 0.5}{0.3 \times 0.05905 \times 0.5 + 0.7 \times 0.12450 \times 0.5} \approx 0.83108$$

3. Maximization (M-step)

Below we will use $x_{i[y_1]}$ to refer to the variable $\{y_1\}$ of observation x_i and $x_{i[y_2 \wedge y_3]}$ to refer to the variables $\{y_2, y_3\}$ of observation x_i.

For each cluster, c_k, we will calculate the following in order to update the parameters:

$$N_k = \sum_i \gamma_{k,i} \tag{4}$$

$$p'_k = \frac{1}{N_k} \sum_i \gamma_{k,i} \cdot x_{i[y_1]} \tag{5}$$

$$\mu'_k = \frac{1}{N_k} \sum_i \gamma_{k,i} \cdot x_{i[y_2 \wedge y_3]} \tag{6}$$

$$\Sigma'_k = \frac{1}{N_k} \sum_i \gamma_{k,i} \cdot \left(x_{i[y_2 \wedge y_3]} - \mu'_k\right) \cdot (x_{i[y_2 \wedge y_3]} - \mu'_k)^T \tag{7}$$

Considering $N = \sum_k N_k$, we can also update the priors:

$$\pi'_k = \frac{N_k}{N} \tag{8}$$

We can now update the values for both clusters using the previous equations.

We start off by calculating the sum of the weights, for each cluster, N_k, by employing equation (4):

$$N_1 = \sum_i \gamma_{1,i} = \gamma_{1,1} + \gamma_{1,2} + \gamma_{1,3} + \gamma_{1,4} = 1.54467$$

$$N_2 = \sum_i \gamma_{2,i} = \gamma_{2,1} + \gamma_{2,2} + \gamma_{2,3} + \gamma_{2,4} = 2.45533$$

Cluster 1 Updates

Now for cluster 1 we can update p_1, μ_1, Σ_1 and π_1, by employing, (5), (6), (7) and (8), respectively:

$$p'_1 = \frac{1}{N_1} \sum_i \gamma_{1,i} \cdot x_{i[y_1]} = \frac{\gamma_{1,1} \cdot 1 + \gamma_{1,2} \cdot 0 + \gamma_{1,3} \cdot 0 + \gamma_{1,4} \cdot 1}{1.54467} = 0.23404$$

$$\mu'_1 = \frac{1}{N_1} \sum_i \gamma_{1,i} \cdot x_{i[y_2 \wedge y_3]} = \frac{\gamma_{1,1} \cdot \begin{pmatrix}0.6\\0.1\end{pmatrix} + \gamma_{1,2} \cdot \begin{pmatrix}-0.4\\0.8\end{pmatrix} + \gamma_{1,3} \cdot \begin{pmatrix}0.2\\0.5\end{pmatrix} + \gamma_{1,4} \cdot \begin{pmatrix}0.4\\-0.1\end{pmatrix}}{1.54467} = \begin{pmatrix}0.02651\\0.50713\end{pmatrix}$$

$$\Sigma'_1 = \frac{1}{N_1} \sum_i \gamma_{1,i} \cdot \left(x_{i[y_2 \wedge y_3]} - \mu'_1\right) \cdot (x_{i[y_2 \wedge y_3]} - \mu'_1)^T$$

$$= \frac{1}{1.54467} \times \Bigg[\gamma_{1,1} \cdot \left(\begin{pmatrix} 0.6 \\ 0.1 \end{pmatrix} - \begin{pmatrix} 0.02651 \\ 0.50713 \end{pmatrix} \right) \cdot \left(\begin{pmatrix} 0.6 \\ 0.1 \end{pmatrix} - \begin{pmatrix} 0.02651 \\ 0.50713 \end{pmatrix} \right)^T$$

$$+ \gamma_{1,2} \cdot \left(\begin{pmatrix} -0.4 \\ 0.8 \end{pmatrix} - \begin{pmatrix} 0.02651 \\ 0.50713 \end{pmatrix} \right) \cdot \left(\begin{pmatrix} -0.4 \\ 0.8 \end{pmatrix} - \begin{pmatrix} 0.02651 \\ 0.50713 \end{pmatrix} \right)^T$$

$$+ \gamma_{1,3} \cdot \left(\begin{pmatrix} 0.2 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 0.02651 \\ 0.50713 \end{pmatrix} \right) \cdot \left(\begin{pmatrix} 0.2 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 0.02651 \\ 0.50713 \end{pmatrix} \right)^T$$

$$+ \gamma_{1,4} \cdot \left(\begin{pmatrix} 0.4 \\ -0.1 \end{pmatrix} - \begin{pmatrix} 0.02651 \\ 0.50713 \end{pmatrix} \right) \cdot \left(\begin{pmatrix} 0.4 \\ -0.1 \end{pmatrix} - \begin{pmatrix} 0.02651 \\ 0.50713 \end{pmatrix} \right)^T \Bigg]$$

$$= \begin{pmatrix} 0.14137 & -0.10541 \\ -0.10541 & 0.09605 \end{pmatrix}$$

$$\pi'_1 = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2} = \frac{1.54467}{1.54467 + 2.45533} = 0.38617$$

Cluster 2 Updates

Finally, for cluster 2 we can update p_2, μ_2, Σ_2 and π_2, by employing, (5), (6), (7) and (8), respectively:

$$p'_2 = \frac{1}{N_2} \sum_i \gamma_{2,i} \cdot x_{i\,[y_1]} = \frac{\gamma_{2,1} \cdot 1 + \gamma_{2,2} \cdot 0 + \gamma_{2,3} \cdot 0 + \gamma_{2,4} \cdot 1}{2.45533} = 0.66732$$

$$\mu'_2 = \frac{1}{N_2} \sum_i \gamma_{2,i} \cdot x_{i\,[y_2 \wedge y_3]} = \frac{\gamma_{2,1} \cdot \begin{pmatrix} 0.6 \\ 0.1 \end{pmatrix} + \gamma_{2,2} \cdot \begin{pmatrix} -0.4 \\ 0.8 \end{pmatrix} + \gamma_{2,3} \cdot \begin{pmatrix} 0.2 \\ 0.5 \end{pmatrix} + \gamma_{2,4} \cdot \begin{pmatrix} 0.4 \\ -0.1 \end{pmatrix}}{2.45533} = \begin{pmatrix} 0.30914 \\ 0.21042 \end{pmatrix}$$

$$\Sigma'_2 = \frac{1}{N_2} \sum_i \gamma_{2,i} \cdot (x_{i\,[y_2 \wedge y_3]} - \mu'_2) \cdot (x_{i\,[y_2 \wedge y_3]} - \mu'_2)^T$$

$$= \frac{1}{2.45533} \times \Bigg[\gamma_{2,1} \cdot \left(\begin{pmatrix} 0.6 \\ 0.1 \end{pmatrix} - \begin{pmatrix} 0.30914 \\ 0.21042 \end{pmatrix} \right) \cdot \left(\begin{pmatrix} 0.6 \\ 0.1 \end{pmatrix} - \begin{pmatrix} 0.30914 \\ 0.21042 \end{pmatrix} \right)^T$$

$$+ \gamma_{2,2} \cdot \left(\begin{pmatrix} -0.4 \\ 0.8 \end{pmatrix} - \begin{pmatrix} 0.30914 \\ 0.21042 \end{pmatrix} \right) \cdot \left(\begin{pmatrix} -0.4 \\ 0.8 \end{pmatrix} - \begin{pmatrix} 0.30914 \\ 0.21042 \end{pmatrix} \right)^T$$

$$+ \gamma_{2,3} \cdot \left(\begin{pmatrix} 0.2 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 0.30914 \\ 0.21042 \end{pmatrix} \right) \cdot \left(\begin{pmatrix} 0.2 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 0.30914 \\ 0.21042 \end{pmatrix} \right)^T$$

$$+ \gamma_{2,4} \cdot \left(\begin{pmatrix} 0.4 \\ -0.1 \end{pmatrix} - \begin{pmatrix} 0.30914 \\ 0.21042 \end{pmatrix} \right) \cdot \left(\begin{pmatrix} 0.4 \\ -0.1 \end{pmatrix} - \begin{pmatrix} 0.30914 \\ 0.21042 \end{pmatrix} \right)^T \Bigg]$$

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

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

4. Verify the log likelihood

Since we are only performing one epoch of the EM clustering algorithm, we can skip this step.

5. Conclusion

After performing one epoch of the EM clustering algorithm, we end up with the following updated parameters for each cluster:

Cluster	p'	μ'	Σ'	π'
Cluster 1	0.23404	$\begin{pmatrix}0.02651\\0.50713\end{pmatrix}$	$\begin{pmatrix}0.14137 & -0.10541\\-0.10541 & 0.09605\end{pmatrix}$	0.38617
Cluster 2	0.66732	$\begin{pmatrix}0.30914\\0.21042\end{pmatrix}$	$\begin{pmatrix}0.10829 & -0.08865\\-0.08865 & 0.10412\end{pmatrix}$	0.61383

Table 2: Updated parameters for the two clusters after one epoch of the EM clustering algorithm

2. **Given the new observation, $x_{new} = \begin{bmatrix}1 & 0.3 & 0.7\end{bmatrix}^T$, determine the cluster memberships (posteriors).**

Note: As per the *FAQ*, we will be using the updated values obtained in exercise 1.

Using the equation on (3), we can compute, for each cluster, c_k, the value of $P(y_2, y_3|c_k)$ for the new observation, x_{new}:

$$P(y_2 = 0.3, y_3 = 0.7|c_1) = \mathcal{N}_1(y_2 = 0.3, y_3 = 0.7|\mu_1', \Sigma_1') \approx 0.02708$$

$$P(y_2 = 0.3, y_3 = 0.7|c_2) = \mathcal{N}_2(y_2 = 0.3, y_3 = 0.7|\mu_2', \Sigma_2') \approx 0.06843$$

Now, by using the equation on (2), we can compute the normalized posteriors:

$$\begin{aligned}
\gamma_{1,new} &= \frac{P(y_1 = 1|c_1)P(y_2 = 0.3, y_3 = 0.7|c_1)P(c_1)}{P(y_1 = 1|c_1)P(y_2 = 0.3, y_3 = 0.7|c_1)P(c_1) + P(y_1 = 1|c_2)P(y_2 = 0.3, y_3 = 0.7|c_2)P(c_2)} \\
&= \frac{p_1' \times 0.02708 \times \pi_1'}{p_1' \times 0.02708 \times \pi_1' + p_2' \times 0.06843 \times \pi_2'} \\
&= \frac{0.23404 \times 0.02708 \times 0.38617}{0.23404 \times 0.02708 \times 0.38617 + 0.66732 \times 0.06843 \times 0.61383} \approx 0.08029 \\
\gamma_{2,new} &= \frac{P(y_1 = 1|c_2)P(y_2 = 0.3, y_3 = 0.7|c_2)P(c_2)}{P(y_1 = 1|c_1)P(y_2 = 0.3, y_3 = 0.7|c_1)P(c_1) + P(y_1 = 1|c_2)P(y_2 = 0.3, y_3 = 0.7|c_2)P(c_2)} \\
&= \frac{p_2' \times 0.06843 \times \pi_2'}{p_1' \times 0.02708 \times \pi_1' + p_2' \times 0.06843 \times \pi_2'} \\
&= \frac{0.66732 \times 0.06843 \times 0.61383}{0.23404 \times 0.02708 \times 0.38617 + 0.66732 \times 0.06843 \times 0.61383} \approx 0.91971
\end{aligned}$$

3. **Performing a hard assignment of observations to clusters under a ML assumption, identify the silhouette of both clusters under a Manhattan distance.**

Note: As per the *FAQ*, we will be using the updated values obtained in exercise 1 and only show the calculus for the observation x_2, only presenting the remaining results in Table 3.

Firstly, we need to calculate the updated likelihoods. For that, we consider $\{y_1\} \perp\!\!\!\perp \{y_2, y_3\}$ and multiply $P(y_1|c_k)$ by $P(y_2, y_3|c_k)$, which is given by the equation (3):

$$\begin{aligned}
P(x_2|c_1) &= P(y_1 = 0|c_1) \times P(y_2 = -0.4, y_3 = 0.8|c_1) = (1 - p'_1) \times \mathcal{N}_1(y_2 = -0.4, y_3 = 0.8|\mu'_1, \Sigma'_1) \\
&= 0.76596 \times 1.65326 \approx 1.26633 \\
P(x_2|c_2) &= P(y_1 = 0|c_2) \times P(y_2 = -0.4, y_3 = 0.8|c_2) = (1 - p'_2) \times \mathcal{N}_2(y_2 = -0.4, y_3 = 0.8|\mu'_2, \Sigma'_2) \\
&= 0.33268 \times 0.26673 \approx 0.08874
\end{aligned}$$

Cluster	$P(x_1|c_k)$	$P(x_2|c_k)$	$P(x_3|c_k)$	$P(x_4|c_k)$
Cluster 1 (c_1)	0.23147	1.26633	1.43811	0.02077
Cluster 2 (c_2)	0.94954	0.08874	0.45417	0.72331

Table 3: Updated likelihoods for the two clusters

Based on the calculated likelihoods, we can infer that x_1 and x_4 are assigned to Cluster 2, while x_2 and x_3 are assigned to Cluster 1:

$$C_1 = \{x_2, x_3\} \qquad C_2 = \{x_1, x_4\}$$

The Manhattan distance is given by the following equation:

$$d(P, Q) = d((a_1, b_1, c_1), (a_2, b_2, c_2)) = |a_2 - a_1| + |b_2 - b_1| + |c_2 - c_1| \tag{9}$$

By employing equation (9), we can create the Table 4 that has the manhattan distances between every pair of observations. Only the upper diagonal entries are filled, because the distance function is commutative.

$d(x_i, x_j)$	x_1	x_2	x_3	x_4
x_1	0	2.7	1.8	0.4
x_2	-	0	0.9	2.7
x_3	-	-	0	1.8
x_4	-	-	-	0

Table 4: Manhattan distances between every pair of observations

And the Cohesion ($a(x_i)$), Separation ($b(x_i)$) and Silhouette ($S(x_i)$), for a given observation x_i, are given by:

$$\begin{aligned}
a(x_i) &= \text{average distance of } x_i \text{ to the other points in its cluster} \\
b(x_i) &= \min_j\{\text{average distance of } x_i \text{ to the points of cluster } C_j \text{ such that } x_i \notin C_j\}
\end{aligned}$$

$$S(x_i) = \frac{b(x_i) - a(x_i)}{\max\{b(x_i), a(x_i)\}} \tag{10}$$

The silhouette for a cluster C_k is given by:

$$S(C_k) = \frac{\sum_{x_i \in C_k} S(x_i)}{|C_k|} \tag{11}$$

By replacing the values on the equation (10), we get the following values:

$$S(x_1) = \frac{\frac{d(x_1,x_2)+d(x_1,x_3)}{2} - d(x_1, x_4)}{\max\left\{\frac{d(x_1,x_2)+d(x_1,x_3)}{2}, d(x_1, x_4)\right\}} \approx 0.82222$$

$$S(x_2) = \frac{\frac{d(x_2,x_1)+d(x_2,x_4)}{2} - d(x_2, x_3)}{\max\left\{\frac{d(x_2,x_1)+d(x_2,x_4)}{2}, d(x_2, x_3)\right\}} \approx 0.66667$$

$$S(x_3) = \frac{\frac{d(x_3,x_1)+d(x_3,x_4)}{2} - d(x_3, x_2)}{\max\left\{\frac{d(x_3,x_1)+d(x_3,x_4)}{2}, d(x_3, x_2)\right\}} \approx 0.49999$$

$$S(x_4) = \frac{\frac{d(x_4,x_2)+d(x_4,x_3)}{2} - d(x_4, x_1)}{\max\left\{\frac{d(x_4,x_2)+d(x_4,x_3)}{2}, d(x_4, x_1)\right\}} \approx 0.82222$$

Therefore the values of the silhouette for the clusters are given by (11):

$$S(C_1) = \frac{S(x_2) + S(x_3)}{2} = 0.58333 \qquad S(C_2) = \frac{S(x_1) + S(x_4)}{2} = 0.82222$$

4. **Knowing the purity of the clustering solution is 0.75, identify the number of possible classes (ground truth).**

   Given the purity score of 0.75 and the presence of four observations, we can deduce that approximately 75% of the observations ($0.75 \times 4 = 3$) were correctly assigned to their respective clusters. However, it also implies that one observation was misclassified.

   The unaccounted observation may belong to the opposing cluster, or possibly a cluster that wasn't initially considered, as our analysis began with a default assumption of two clusters.

   Therefore, the number of possible classes is either two or three.

Part II: Programming and critical analysis

Recall the `column_diagnosis.arff` dataset from previous homeworks. For the following exercises, normalize the data using sklearn's `MinMaxScaler`.

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.**

   Using `sklearn`'s `cluster.KMeans` class, we can apply a k-means clustering algorithm for each $k \in \{2, 3, 4, 5\}$ with `random` = 0 and remaining parameters as default.
   We opted for the default parameters in the `metric.silhouette_score` function.
   To calculate the purity score, we used the code in the `purity_score` function from the course's N5 (Clustering) Notebook available in Fénix.

```python
import numpy as np, pandas as pd
from scipy.io.arff import loadarff
from sklearn.preprocessing import MinMaxScaler
from sklearn import cluster, metrics

# Read the ARFF file, prepare data and normalize it
data = loadarff("./data/column_diagnosis.arff")
df = pd.DataFrame(data[0])
df["class"] = df["class"].str.decode("utf-8")
X, y = df.drop("class", axis=1), df["class"]
X_scaled = MinMaxScaler().fit_transform(X)

# Parametrize the clustering and learn the model
k_means_models = []
for n_clusters in [2, 3, 4, 5]:
    k_means = cluster.KMeans(n_clusters=n_clusters, random_state=0)
    k_means_models.append(k_means.fit(X_scaled))

for model in k_means_models:
    n_clusters = model.n_clusters
    y_pred = model.labels_

    # Calculate the silhouette
    silhouette = metrics.silhouette_score(X_scaled, y_pred)

    # Calculate the purity
    conf_matrix = metrics.cluster.contingency_matrix(y, y_pred)
    purity = np.sum(np.amax(conf_matrix, axis=0)) / np.sum(conf_matrix)

    # Print the results for each number of clusters
    print(f"Clustering with n_clusters = {n_clusters}")
    print(f"\tSilhouette = {silhouette:6.5f}")
    print(f"\tPurity = {purity:6.5f}")
    print()
```

`n_clusters`	2	3	4	5
Silhouette	0.36044	0.29579	0.27442	0.23824
Purity	0.63226	0.66774	0.66129	0.67742

Table 5: Silhouette and purity scores (rounded to 5 decimal places) for `n_clusters` $\in \{2, 3, 4, 5\}$

2. **Consider the application of PCA after the data normalization:**

   (a) **Identify the variability explained by the top two principal components.**

```python
from sklearn.decomposition import PCA

# Apply PCA to the normalized data
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# Variability explained by the top two principal components
explained_variance_ratio = pca.explained_variance_ratio_
print(f"Explained Variance Ratio for Top 2 PCs: {explained_variance_ratio}")
print(f"Total variability: {explained_variance_ratio[0] + explained_variance_ratio[1]}")
```

The explained variability for the top 2 PCs is 56.181445% and 20.955953% respectively.
And the total explained variability is 77.1374%.

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

```python
# Get the absolute weights (loadings) of the top two principal components
pc_weights = np.abs(pca.components_[:2, :])

# Get the feature names
feature_names = X.columns

# Sort the feature names by relevance for each PC
sorted_features_pc1 = [feature_names[i] for i in np.argsort(pc_weights[0])[::-1]]
sorted_features_pc2 = [feature_names[i] for i in np.argsort(pc_weights[1])[::-1]]

print(f"Top Variables for PC1: {sorted_features_pc1}")
print(f"Top Variables for PC2: {sorted_features_pc2}")
```

Top Variables for PC1:

1. pelvic_incidence
2. lumba_lordosis_angle
3. pelvic_tilt
4. sacral_slope
5. degree_spondylolisthesis
6. pelvic_radius

Top Variables for PC2:

1. pelvic_tilt
2. pelvic_radius
3. sacral_slope
4. pelvic_incidence
5. lumbar_lordosis_angle
6. degree_spondylolisthesis

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.**

```python
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from sklearn.preprocessing import LabelEncoder

# Apply PCA to the normalized data
```

```python
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# Convert labels to numerical format
le = LabelEncoder()
y_numerical = le.fit_transform(y)

# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))

# Plot the ground diagnoses
scatter1 = ax1.scatter(X_pca[:, 0], X_pca[:, 1], c=y_numerical, cmap='viridis')
ax1.set_title('Ground Diagnoses')
ax1.legend(handles=scatter1.legend_elements()[0], labels=list(set(y)))

# Plot the k-means clustering solution (k=3)
k_means = cluster.KMeans(n_clusters=3, random_state=0)
y_pred = k_means.fit_predict(X_scaled)

scatter2 = ax2.scatter(X_pca[:, 0], X_pca[:, 1], c=y_pred, cmap='viridis')
ax2.set_title('K-Means Clustering (k=3)')
ax2.legend(handles=scatter2.legend_elements()[0], labels=['Cluster 0', 'Cluster 1'
    , 'Cluster 2'])

plt.show()
```

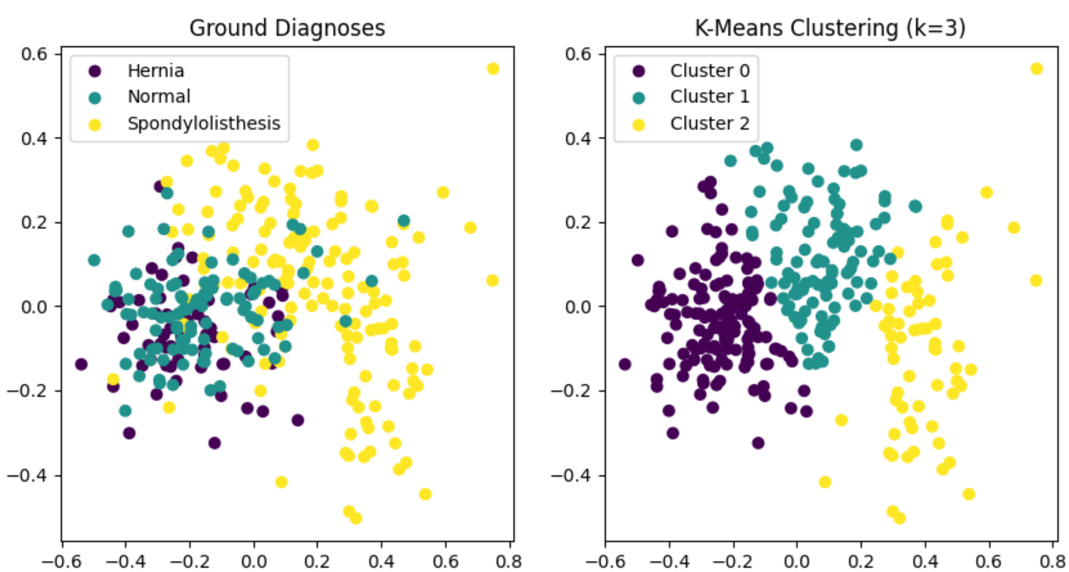

Figure 1: Projected data

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.**

Raquel