Aprendizagem 2021/22

Homework IV – Group 114

I. Pen-and-paper

1)

	y_1	y_2
x_1	2	4
x_2	-1	-4
x_3	-1	2
x_4	4	0

$\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

$\Sigma_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$

$\pi_1 = p(c_1 = 1) = 0.7$

$\pi_2 = p(c_2 = 1) = 0.3$

E-Step:

$$P(Class = c \mid x^{(n)}) = \frac{P(x^{(n)} \mid Class = c) \cdot P(Class = c)}{P(x^{(n)})} =$$

(pelo Teorema de Bayes)

$$P(c_1 = 1 \mid x_1) = \frac{P(x_1 \mid c_1 = 1) \cdot P(c_1 = 1)}{P(x_1)} = \frac{P(x_1 \mid c_1 = 1) \cdot 0,7}{P(x_1 \mid c_1 = 1) + P(x_1 \mid c_2 = 1)} =$$

$$= \frac{N\left(\mu = \begin{bmatrix} 2 \\ 4 \end{bmatrix}; \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \cdot 0,7}{N\left(\mu = \begin{bmatrix} 2 \\ 4 \end{bmatrix}; \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) + N\left(\mu = \begin{bmatrix} -1 \\ -4 \end{bmatrix}; \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\right)}$$

$= 0,7$

$P(c_2 = 1 \mid x_1) = 1,78 \times 10^{-9}$

$P(c_1 = 1 \mid x_2) = 7,64 \times 10^{-18}$

$P(c_2 = 1 \mid x_2) = 0,3$

$P(c_1 = 1 \mid x_3) = 0,67$

$P(c_2 = 1 \mid x_3) = 0,012$

$P(c_1 = 1 \mid x_4) = 0,504$

$P(c_2 = 1 \mid x_4) = 0,084$

Cluster 1: x_1 ; x_3 ; x_4

Cluster 2: x_2

HW4

1

	y_1	y_2
x_1	2	4
x_2	-1	-4
x_3	-1	2
x_4	4	0

$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ $\pi_1 = p(c_1 = 1) = 0,7$

$\pi_2 = p(c_2 = 1) = 0,3$

$x^{(1)} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$, $x^{(2)} = \begin{bmatrix} -1 \\ -4 \end{bmatrix}$, $x^{(3)} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$, $x^{(4)} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$

$\mu^1 = x^{(1)} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$

$\mu^2 = x^{(2)} = \begin{bmatrix} -1 \\ -4 \end{bmatrix}$

M-step:

$P(c_1 = 1) = 0,7$

$P(c_2 = 1) = 0,3$

$P(c_1 = 1 \mid x_1) = 0,7$

$P(c_2 = 1 \mid x_1) = 1,79 \times 10^{-9}$

$P(c_1 = 1 \mid x_2) = 7,64 \times 10^{-18}$

$P(c_2 = 1 \mid x_2) = 0,3$

$P(c_1 = 1 \mid x_3) = 0,67$

$P(c_2 = 1 \mid x_3) = 0,012$

$P(c_1 = 1 \mid x_4) = 0,504$

$P(c_2 = 1 \mid x_4) = 0,084$

$\mu_1 = \begin{bmatrix} 1,461 \\ 10,469 \end{bmatrix}$ $\mu_2 = \begin{bmatrix} -0,147 \\ -2,971 \end{bmatrix}$

$$\Sigma_1 = \begin{bmatrix} \Sigma_1^{(1,1)} = 4,008 & \Sigma_1^{(1,2)} = -0,992 \\ \Sigma_1^{(2,1)} = -0,992 & \Sigma_1^{(2,2)} = 70,753 \end{bmatrix}$$

$$\Sigma_c^{(i,j)} = \frac{\sum_{k=1}^{4} p(c \mid x_k)(a_{ki} - \mu_{ci})(a_{kj} - \mu_{cj})}{\sum_{k=1}^{4} p(c \mid x_k)}$$

$$\pi_k = p(c = k) = \sum_{i=1}^{4} \frac{p(c = k \mid x_i)}{\sum_{j=1}^{2} \sum_{i=1}^{4} p(c = j \mid x_i)}$$

$$P(c = 1) = \frac{0,7 + 7,64 \times 10^{-18} + 0,67 + 0,507}{} = \frac{1,87}{} = 0,825$$

$$P(c = 2) = \frac{1,78 \times 10^{-9} + 0,3 + 0,012 + 0,084}{} = \frac{0,396}{} = 0,175$$

$$\Sigma_1^{(1,1)} = \frac{0,7\,(2 - 1,461)(4 - 1,461) + 0,67(-1 - 1,461)(2 - 1,461) + 0,504(4 - 1,461)(0 - 1,461)}{1,874}$$

$$\Sigma_2 = \begin{bmatrix} 6,271 & 3,929 \\ 3,929 & 3,423 \end{bmatrix}$$

Para $c = 1$, $N = \left(\mu_1 = \begin{bmatrix} 1,461 \\ 10,469 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 4,008 & -0,992 \\ -0,992 & 70,753 \end{bmatrix}\right)$

Para $c = 2$, $N = \left(\mu_2 = \begin{bmatrix} -0,147 \\ -2,971 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 6,271 & 3,929 \\ 3,929 & 3,423 \end{bmatrix}\right)$

2) Como o resultado obtido a partir do cálculo do "silhouette" é relativamente próximo de 1 concluímos que os clusters usados no algoritmo "EM" estão bem separados.

3) a) Para N = 5

i)102 $3 \cdot \left(N^2 + N\right) + (2 \cdot N + 2)$

ii) 243 3^N

iii) 76 $1 + 3N + N^2$

b) Para a "VC dimension", o melhor classificador é o **Bayesiano** porque o crescimento de "features" é limitado, como se pode observar no gráfico. O classificador que recorre a árvores de decisão é a pior escolha, porque quantos mais "bins" existirem para cada "feature", mais classificações diferentes vão existir. Assim d_{VC}(Bayesian)< d_{VC}(MLP)< < d_{VC}(Decision Tree)

[**O classificador de Árvores de decisão cresce exponencialmente, não permitindo uma análise correta dos classificadores MLP e Bayesiano. Assim, foi "cortado" do plot**]

(2)

Silhouette

$s(x) = 1 - \frac{a(x)}{b(x)}$

$C_1 : x_1 ; x_3 ; x_4$

$C_2 = x_2$

$s(x_1) : \frac{\|x_1 - x_3\| + \|x_1 - x_4\|}{\|x_1 - x_2\|} =$

$x1: \quad a = \frac{\|x_1 - x_3\| + \|x_1 - x_4\|}{2} = \frac{\sqrt{(2+1)^2 + (4-2)^2} + \sqrt{(2-4)^2 + (4^2)}}{2} = \frac{\sqrt{13} + \sqrt{20}}{2} = 4{,}04$

$b = \|x_1 - x_2\| = \sqrt{(2+1)^2 + (4+4)^2} = \sqrt{9+64} = \sqrt{73} = 8{,}54$

$s = 1 - \frac{a}{b}$

$s = 1 - \frac{\frac{\sqrt{13} + \sqrt{20}}{2}}{\sqrt{73}} \Leftrightarrow s = 1 - \frac{4{,}04}{8{,}54} = 0{,}53$

$x_3: \quad a = \frac{\|x_3 - x_1\| + \|x_3 - x_4\|}{2} = \frac{\sqrt{13} + \sqrt{(-1-4)^2 + (2-0)^2}}{2}$

$= \frac{\sqrt{13} + \sqrt{29}}{2} = 4{,}50$

$b = \|x_3 - x_2\| = \sqrt{(-1+\cancel{1})^2 + (2+4)^2} = \sqrt{36} = 6$

$s(x_3) = 1 - \frac{4{,}50}{6} = 0{,}25$

$x_4: \quad a = \frac{\|x_4 - x_1\| + \|x_4 - x_3\|}{2} = \frac{\sqrt{20} + \sqrt{29}}{2} = 4{,}93$

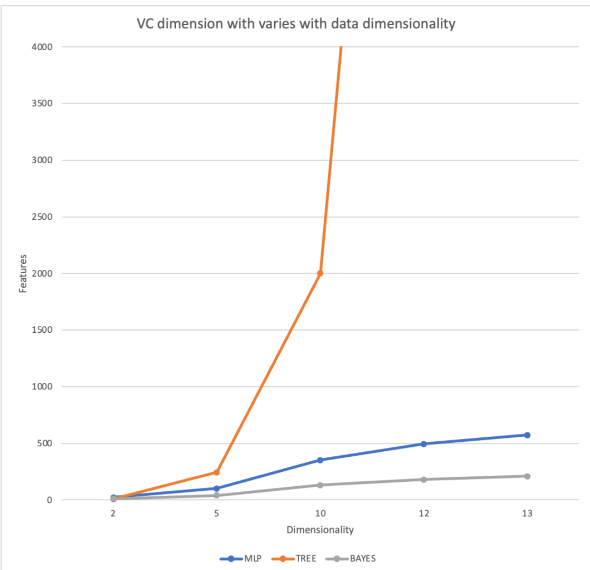
$b = \|x_4 - x_2\| = \sqrt{25 + 16} = \sqrt{41} = 6{,}40$

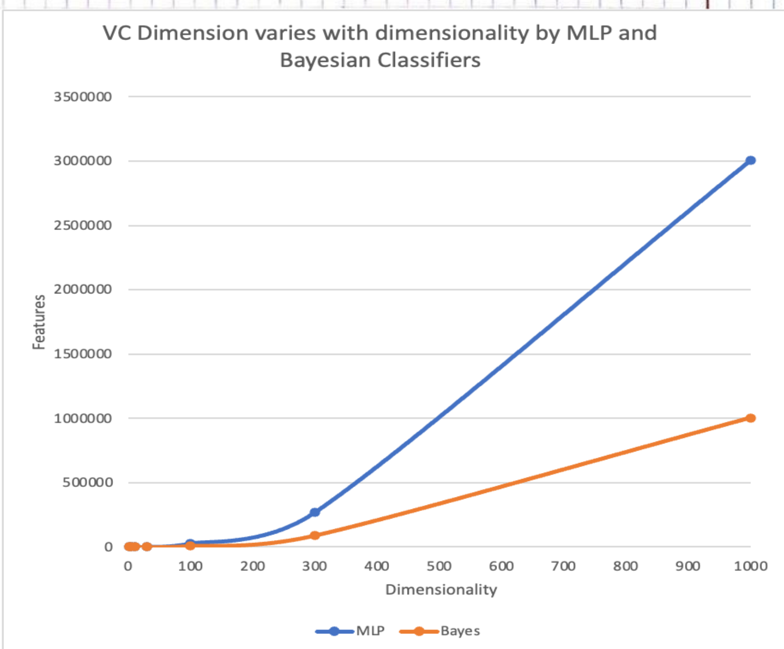
$1 - 0{,}77 = 0{,}23$

$s(c_1) = \frac{s(x_1) + s(x_4) + s(x_3)}{3} = \frac{0{,}53 + 0{,}25 + 0{,}23}{3} = 0{,}33$

$s(x_2) = s(c_2) = 1$

$\text{Silhouette}(C) = \frac{s(c_1) + s(c_2)}{2} = 0{,}6665$





c) Recorrendo a elevadas dimensões, o **Bayesiano** é a melhor opção, estando o seu crescimento limitado, gerando metade das "features" do classificador **MLP**. Assim d(MLP)> d(Bayesian)

II. Programming and critical analysis

4) Answer 5

$C = \{c_1, c_2, \dots, c_K\}$ is the set of clusters
$L = \{L_1, L_2, \dots, L_G\}$ is the set of reference classes

$$\varphi_L(c_k) = \max_{j=1..G}(|c_k \cap L_j|)$$

$$ECR = \frac{1}{K}\sum_{k=1}^{K}(|c_k| - \varphi_L(c_k))$$

a)

K = 2 -> Cluster 1 -> Classe 0 (malignant): 9 ## Classe 1 (benign): 220

K = 2 -> Cluster 2 -> Classe 0 (malignant): 435 ## Classe 1 (benign): 18

ECR K=2 -> : 13.500

K = 3 -> Cluster 1 -> Classe 0 (malignant): 433 ## Classe 1 (benign): 9

K = 3 -> Cluster 2 -> Classe 0 (malignant): 11 ## Classe 1 (benign): 104

K = 3 -> Cluster 3 -> Classe 0 (malignant): 0 ## Classe 1 (benign): 125

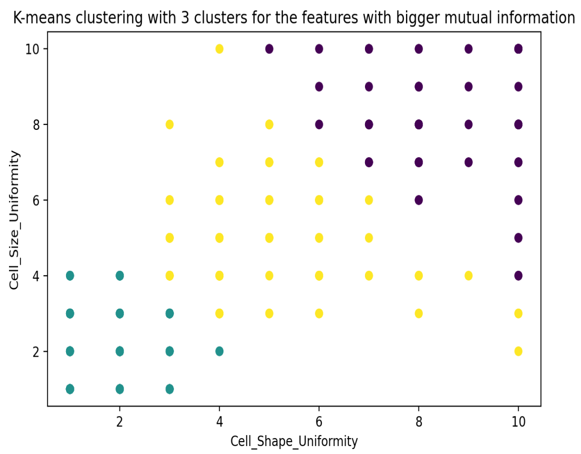
ECR K = 3 -> : 6.667

b)

Silhouette Score K = 2 : 0.597

Silhouette Score K = 3 : 0.526

5) 2-Top Features : Cell_Size_Uniformity Cell_Shape_Uniformity



6) Com 3 clusters e com as 2 melhores "features" (Cell_Size_Uniformity, Cell_Shape_Uniformity) concluímos que a classificação é mais uniforme, aglomerada.

III. APPENDIX

```
from ctypes import c_int32
import numpy as np
from numpy.lib.function_base import cov
from scipy.stats import multivariate_normal
from scipy.io.arff import loadarff
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.metrics import mutual_info_score
from sklearn import *
from sklearn import datasets
import pandas as pd
from sklearn.metrics import v_measure_score
import matplotlib.pyplot as plt

if __name__ == '__main__':
    raw_data = loadarff('breast.w.arff')
    X = pd.DataFrame(raw_data[0])
    classe = X.pop('Class')
    Y = classe.values
    Y = Y.astype(int)

#Exercise 4
for s in [2,3]:
    c_1 = [0,0]
    c_2 = [0,0]
    c_3 = [0,0]
    kmeans = KMeans(n_clusters=s,random_state = 0).fit(X)
    pred = kmeans.predict(X)
    for k in range(0,len(pred)-1):
        if pred[k] == 0: #cluster 1
            if Y[k]==1:
                c_1[1] += 1
            else:
                c_1[0] +=1
        elif pred[k]== 1 : #cluster 2
            if Y[k]==1:
                c_2[1] +=1
            else:
                c_2[0] +=1
        elif pred[k] == 2:#cluster 3
            if Y[k]==1:
                c_3[1] += 1
            else:
                c_3[0] +=1
    if s == 2:
        print("K = 2 -> Cluster 1 -> Classe 0 (malignant): %d ## Classe 1 (benign): %d " %
(c_1[0],c_1[1]))
        print("K = 2 -> Cluster 2 -> Classe 0 (malignant): %d ## Classe 1 (benign): %d " %
(c_2[0],c_2[1]))
        print("ECR K=2  -> : %.3f " % ((1/2) * (min(c_1[0],c_1[1]) + min(c_2[0],c_2[1]))))
    elif s == 3:
        print("K = 3 -> Cluster 1 -> Classe 0 (malignant): %d ## Classe 1 (benign): %d " %
(c_1[0],c_1[1]))
        print("K = 3 -> Cluster 2 -> Classe 0 (malignant): %d ## Classe 1 (benign): %d " %
(c_2[0],c_2[1]))
        print("K = 3 -> Cluster 3 -> Classe 0 (malignant): %d ## Classe 1 (benign): %d " %
(c_3[0],c_3[1]))
        print("ECR K = 3 -> : %.3f" % ((1/3) * (min(c_1[0],c_1[1]) + min(c_2[0],c_2[1]) +
min(c_3[0],c_3[1]))))

for i in [2,3]:
    kmeans = KMeans(n_clusters=i)
```

```
    kmeans.fit_predict(X)
    score = silhouette_score(X, kmeans.labels_, metric='euclidean')
    print('Silhouette Score K = %d : %.3f' % (i,score))
#Exercise 5
for i in X:
    print(mutual_info_score(Y,X[i].values), i )
print("2-Top Features : ","Cell_Size_Uniformity", "Cell_Shape_Uniformity")
X.pop("Clump_Thickness")
X.pop("Marginal_Adhesion")
X.pop("Single_Epi_Cell_Size")
X.pop("Bare_Nuclei")
X.pop("Bland_Chromatin")
X.pop("Normal_Nucleoli")
X.pop("Mitoses")
kmeans = KMeans(n_clusters=3,random_state = 0).fit(X)
pred = kmeans.predict(X)
plt.figure(figsize=(3, 3))
plt.scatter(X.iloc[:, 0], X.iloc[:, 1], c=pred)
plt.title("K-means = 3 -> Cell_Size_Uniformity ,Cell_Shape_Uniformity")
plt.show()
```

END