TÉCNICO LISBOA

Aprendizagem 2021/22

Homework IV – Group 024

I. Pen-and-paper

1) A partir do enunciado, obtemos:

$$\mathbf{x}_1 = [2 \quad 4]^T$$
$$\mathbf{x}_2 = [-1 \quad -4]^T$$
$$\mathbf{x}_3 = [-1 \quad 2]^T$$
$$\mathbf{x}_4 = [4 \quad 0]^T$$
$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$
$$\pi_1 = P(c_1 = 1) = 0.7 \qquad \pi_2 = P(c_2 = 1) = 0.3$$

Pretendemos realizar uma iteração do algoritmo *EM*, cujos centróides iniciais estão em $\mathbf{x}_1$ ($\mathbf{u}_1$) e $\mathbf{x}_2$ ($\mathbf{u}_2$). É necessário calcular o *E-Step* e o *M-Step*.

Para o *E-Step*, é preciso saber a verosimilhança (assumindo uma distribuição gaussiana multivariada) e a probabilidade conjunta, para se obter a probabilidade associada a cada *cluster*:

$$P(\mathbf{x}_i|c_k = 1) = \frac{1}{(2 \cdot \pi)^{D/2}} \cdot \frac{1}{|\Sigma_k|^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2} \cdot (\boldsymbol{x}_i - \boldsymbol{u}_k)^T {\Sigma_k}^{-1} \cdot (\boldsymbol{x}_i - \boldsymbol{u}_k)\right) \quad (verosimilhança)$$

$$P(\mathbf{x}_i, c_k = 1) = P(\mathbf{x}_i|c_k = 1) \cdot \pi_k \quad \text{(probabilidade conjunta)}$$

$$P(c_k = 1|\mathbf{x}_i) = \frac{P(\mathbf{x}_i, c_k = 1)}{\sum_{j=1}^{K} P(\mathbf{x}_i, c_j = 1)} \quad \text{(probabilidade associada ao } cluster\ k\text{)}$$

Onde $K = 2$, correspondendo ao número total de *clusters*.

No *M-Step*, calcula-se os novos centróides ($\boldsymbol{\mu}_k$) e matrizes de covariâncias ($\boldsymbol{\Sigma}_k$), e o *posterior* (π_k):

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{X} P(c_k = 1|\mathbf{x}_i) \cdot \mathbf{x}_i}{\sum_{i=1}^{X} P(c_k = 1|\mathbf{x}_i)}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{i=1}^{X} P(c_k = 1|\mathbf{x}_i) \cdot [(\mathbf{x}_i - \boldsymbol{\mu}_k) \cdot (\mathbf{x}_i - \boldsymbol{\mu}_k)^T]}{\sum_{i=1}^{X} P(c_k = 1|\mathbf{x}_i)}$$

$$\pi_k = \frac{\sum_{i=1}^{X} P(c_k = 1|\mathbf{x}_i)}{\sum_{j=1}^{K} \sum_{i=1}^{X} P(c_j = 1|\mathbf{x}_i)}$$

Sendo $X = 4$ o número total de observações.

Concretizando, para o *E-Step:*

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$
$P(\mathbf{x}_i\|c_1 = 1)$	0.1591549431	2.23908996e-17	0.0002392798	7.22562324e-06
$P(\mathbf{x}_i, c_1 = 1)$	0.1114084602	1.56736297e-17	0.0001674958	5.05793627e-06
$P(c_1 = 1\|\mathbf{x}_i)$	0.9999999975	6.56535466e-16	0.9827144049	0.85698183117248

Aprendizagem 2021/22

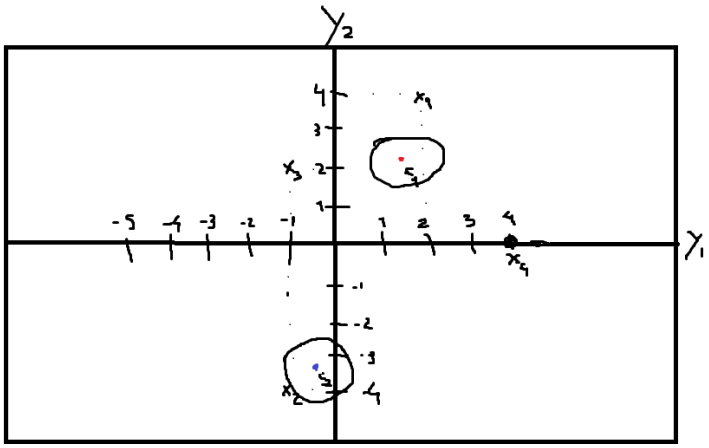
Homework IV – Group 024

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$
$P(\mathbf{x}_i \mid c_2 = 1)$	9.438779514e-10	0.07957747155	9.820640173e-06	2.813660518e-06
$P(\mathbf{x}_i, c_2 = 1)$	2.831633854e-10	0.02387324146	2.946192052e-06	8.440981554e-07
$P(c_2 = 1 \mid \mathbf{x}_i)$	2.541668597e-09	0.9999999999999992	0.017285595123	0.1430181688275

M-Step:

i	$\boldsymbol{\mu}_i$	$\boldsymbol{\Sigma}_i$	π_i
1	$\begin{bmatrix}1.56538325\\2.10072779\end{bmatrix}$	$\begin{bmatrix}4.13282298 & -1.16336779\\-1.16336779 & 2.60560106\end{bmatrix}$	0.7099240583770576
2	$\begin{bmatrix}-0.38370376\\-3.41757815\end{bmatrix}$	$\begin{bmatrix}2.70166014 & 2.1062406\\2.1062406 & 2.16924195\end{bmatrix}$	0.2900759416229424

Obtendo-se o seguinte esboço:



2)

Para realizar o cálculo de uma Silhueta, começamos por identificar cada ponto ao *cluster* correspondente, obtido no exercício anterior. Para isso, repetimos o *E-Step*, agora para a nova iteração, de modo a conhecer a probabilidade de cada ponto pertencer a um determinado *cluster*.

E-Step:

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$
$P(\mathbf{x}_i \mid c_1 = 1)$	0.02067395404	8.54846338e-07	0.020164024108	0.016312342784
$P(\mathbf{x}_i, c_1 = 1)$	0.01467693735	6.06875982e-07	0.014314925828	0.011580524591
$P(c_1 = 1 \mid \mathbf{x}_i)$	0.9999999999998	1.69969795e-05	0.99999999999999	0.912965367988

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework IV – Group 024

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$
$P(\mathbf{x}_i \mid c_2 = 1)$	8.687026493e-15	0.12308612669	5.835206781e-16	0.00380587321
$P(\mathbf{x}_i, c_2 = 1)$	2.519897390e-15	0.03570432410	1.692653102e-16	0.00110399226
$P(c_2 = 1 \mid \mathbf{x}_i)$	1.716909549e-13	0.99998300302	1.182439310e-14	0.08703463201

Classificamos assim os pontos $\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4\}$ como *cluster* c_1, e $\{\mathbf{x}_2\}$ como c_2. Procedemos agora ao cálculo da Silhueta:

$$a(\mathbf{x}_i) = \begin{cases} \frac{1}{|c_k| - 1} \sum_{j \in c_k}^{|c_k|-1} \left\| (\mathbf{x}_i, \mathbf{x}_j) \right\|_2, |c_k| > 1 \\ 0, \quad |c_k| = 1 \end{cases}$$

Sendo que $\mathbf{x}_i \in c_k$.

$$b(\mathbf{x}_i) = \min_{j \neq k} \left\{ \frac{1}{|c_k|} \sum_{l \in c_j}^{|c_j|-1} \|(\mathbf{x}_i, \mathbf{x}_l)\|_2 \right\}$$

$$S(\mathbf{x}_i) = \frac{b(\mathbf{x}_i) - a(\mathbf{x}_i)}{max\{a(\mathbf{x}_i), b(\mathbf{x}_i)\}}$$

$$S(c_k) = \frac{\sum_{\mathbf{x}_i \in c_k}^{|c_k|} S(\mathbf{x}_i)}{|c_k|}$$

$$S(C) = \frac{\sum_{c_k \in C}^{|C|} S(c_k)}{|C|} \quad (Silhueta)$$

Onde C é o conjunto de *clusters*.

Concretizando:

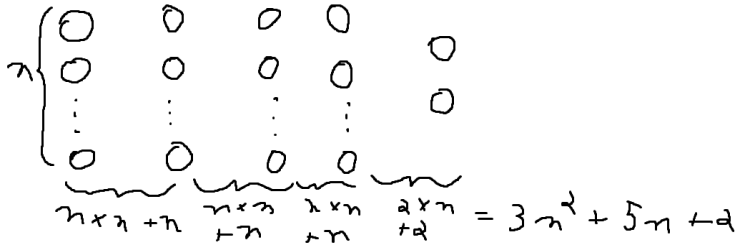
	Cluster	$a(\mathbf{x}_i)$	$b(\mathbf{x}_i)$	$S(\mathbf{x}_i)$	$S(c_i)$	$S(C)$
$\mathbf{x}_1$	c_1	4.038843615	8.5440037453	0.527289109928	0.3361122996	0.6680561498
$\mathbf{x}_3$	c_1	4.495358041	6	0.250773659783		
$\mathbf{x}_4$	c_1	4.928650381	6.4031242374	0.230274128955		
$\mathbf{x}_2$	c_2	0	6.9823759943	1	1	

O valor obtido é mais próximo de 1 do que de 0, o que indica que existe uma boa distinção entre clusters (de modo genérico).

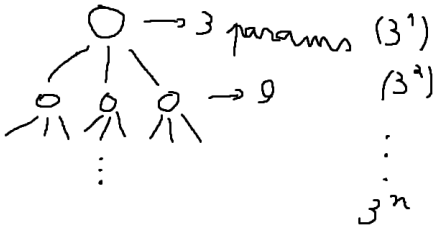
Aprendizagem 2021/22

Homework IV – Group 024

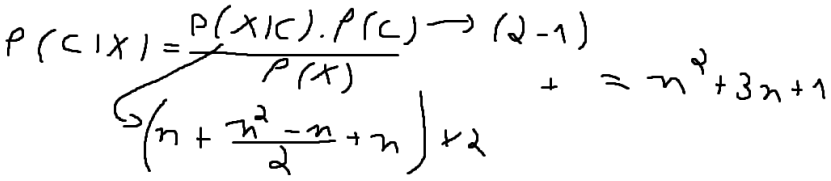
3) i)



ii)



iii)

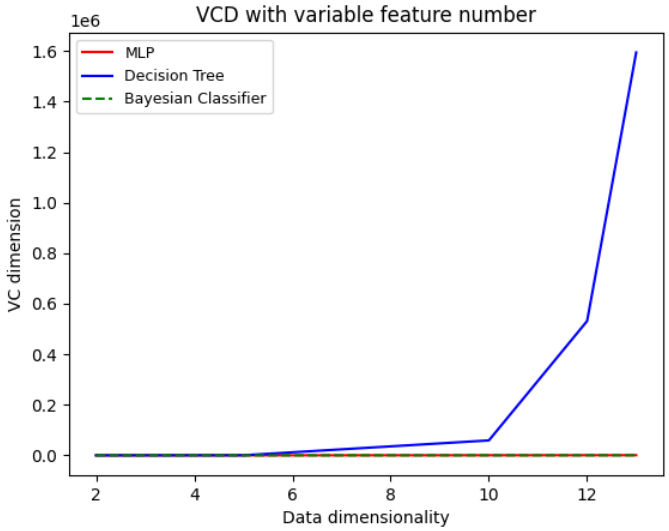


a) i) Para n=5, $VCD = 3 \cdot 5^2 + 5 \cdot 5 + 2 = 102$.

ii) $VCD = 3^5 = 243$.

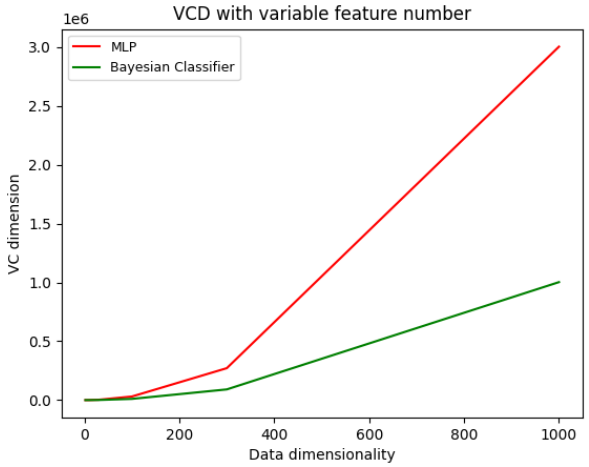
iii) $VCD = 5^2 + 3 \cdot 5 + 1 = 41$.

b)



Conclui-se que a dimensão VC da árvore de decisão cresce muito mais do que as restantes.

c)



Em conclusão, a dimensão VC do MLP é superior à do classificador bayesiano.

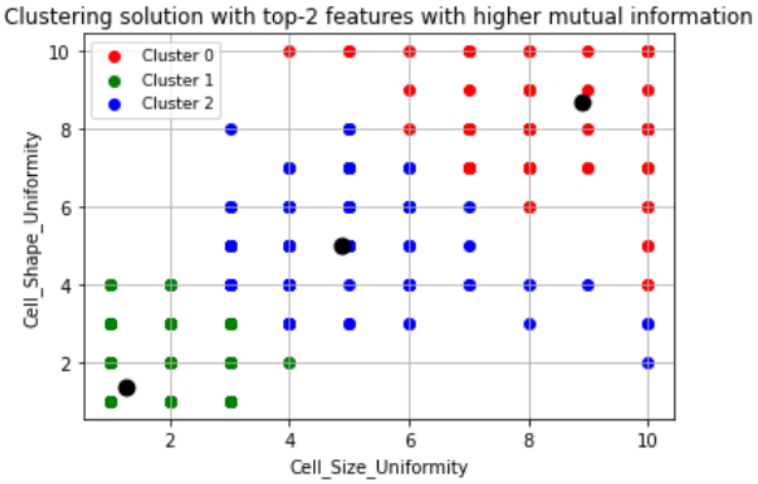
II. Programming and critical analysis

4)

	ECR	Silhueta
$k = 2$	13.5	0.5967981179111456
$k = 3$	6.66666666667	0.5244403755902178

a) Quanto maior for o ECR, mais classificações incorretas existem. Assim k=3 tem um valor melhor.
b) Quanto maior for a Silhueta, maior é a consistência dentro de cada *cluster* (de modo genérico). Deste modo, k=2 tem um valor melhor.

5)



6) A partir do gráfico anterior, foi calculado o ECR e a Silhueta, respetivamente, 11.67 e 0.4927. Comparando estes resultados com os obtidos no exercício 4, para k=3 com todas as variáveis, podemos observar que o ECR aumentou significativamente, mas a Silhueta diminuiu ligeiramente.

A diferença pode ocorrer devido à sobreposição de observações, uma vez que a dimensionalidade das variáveis foi reduzida para 2, sendo que os pontos sobrepostos podem ter classificações diferentes (aumentando o ECR). Como foram escolhidas as duas variáveis com maior *mutual information*, isto é, as variáveis cujos *clusters* classificam melhor as observações, as posições relativas dos centróides estão próximas das originais, pelo que se obtém um valor semelhante para a silhueta.

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework IV – Group 024

III. APPENDIX

```
from scipy.io import arff
import pandas as pd
from sklearn.cluster import KMeans
from sklearn import metrics

cancer = arff.loadarff(r'breast.w.arff')
df = pd.DataFrame(cancer[0])
df.dropna(inplace=True)
df = df.replace(df['Class'][0], 0)
x = 0
while df['Class'][x] == 0:
    x += 1
df = df.replace(df['Class'][x],1)

x=df[['Clump_Thickness','Cell_Size_Uniformity','Cell_Shape_Uniformity','Marginal_Adhesion','Single_E
pi_Cell_Size','Bare_Nuclei','Bland_Chromatin','Normal_Nucleoli','Mitoses']]
y = df['Class']
kmeans2 = KMeans(n_clusters = 2, init = 'random', random_state = 0).fit(x)
kmeans3 = KMeans(n_clusters = 3, init = 'random', random_state = 0).fit(x)

def phi(i, C):
    L = y.tolist()
    cont0 = 0
    cont1 = 0
    for j in range(len(C)):
        if C[j] == i:
            if L[j] == 1:
                cont1 +=1
            else:
                cont0 +=1
    return max(cont0, cont1)
def ecr(K, C):
    res = 0
    for i in range(K):
        z = 0
        for j in C:
            if i == j:
                z+=1
        res += z - phi(i, C)
    return (1/K) * (res)

print(ecr(2, kmeans2.labels_))
print(ecr(3, kmeans3.labels_))
print(metrics.silhouette_score(x, kmeans2.labels_, metric='euclidean'))
print(metrics.silhouette_score(x, kmeans3.labels_, metric='euclidean'))
```

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework IV – Group 024

```
from scipy.io import arff
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans
from sklearn import metrics
from sklearn.feature_selection import mutual_info_classif as MIC

cancer = arff.loadarff(r'breast.w.arff')
df = pd.DataFrame(cancer[0])
df.dropna(inplace=True)
df = df.replace(df['Class'][0], 0)
x = 0
while df['Class'][x] == 0:
    x += 1
df = df.replace(df['Class'][x],1)

x=df[['Clump_Thickness','Cell_Size_Uniformity','Cell_Shape_Uniformity','Marginal_Adhesion','Single_E
pi_Cell_Size','Bare_Nuclei','Bland_Chromatin','Normal_Nucleoli','Mitoses']]
y = df['Class']

def topF():
    mi_score = MIC(x, y)
    j = mi_score.tolist()
    j.sort()
    return [np.where(mi_score == j[-1])[0][0], np.where(mi_score == j[-2])[0][0]]
def newX():
    ind = topF()
    nx = x.columns.values.tolist()
    return df[[nx[ind[0]], nx[ind[1]]]]

X = newX()
F1 = X.columns.values.tolist()[0]
F2 = X.columns.values.tolist()[1]

kmeans3 = KMeans(n_clusters = 3, init = 'random').fit(X)
centroids = kmeans3.cluster_centers_
c0 = X[kmeans3.predict(X) == 0]
c1 = X[kmeans3.predict(X) == 1]
c2 = X[kmeans3.predict(X) == 2]

fig, ax = plt.subplots()
plt.scatter(c0.loc[:,F1].tolist() , c0.loc[:,F2].tolist(), color = 'red')
plt.scatter(c1.loc[:,F1].tolist() , c1.loc[:,F2].tolist(), color = 'green')
plt.scatter(c2.loc[:,F1].tolist() , c2.loc[:,F2].tolist(), color = 'blue')
plt.scatter(centroids[:,0] , centroids[:,1] , s = 80, color = '0')
```

```
ax.legend(labels=['Cluster 0', 'Cluster 1', 'Cluster 2'], loc=2, fontsize = 9)
ax.set_xlabel(F1)
ax.set_ylabel(F2)
ax.set_title('Clustering solution with top-2 features with higher mutual information')
ax.grid(True)
plt.show()
```

END