Aprendizagem 2021/22

Homework IV – Group 9

I. Pen-and-paper

1) Cálculo de probabilidades e probabilidades conjuntas

$\pi_1 = p(c_1 = 1) = 0.7 \qquad \pi_2 = p(c_2 = 1) = 0.3$

$$\mu_1 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad |\Sigma_1| = 1 \quad \Sigma_1^{-1} = \frac{1}{1}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$p(x_i|\, c_1 = 1) = N(x_i|\, \mu_1, \Sigma_1) = \frac{1}{(2\pi)^{\frac{2}{2}} \cdot 1^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2} \cdot \begin{bmatrix} a_1^i - 2 \\ a_2^i - 4 \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_1^i - 2 \\ a_2^i - 4 \end{bmatrix}\right)$$

$$\mu_2 = \begin{bmatrix} -1 \\ -4 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad |\Sigma_2| = 4 \quad \Sigma_2^{-1} = \frac{1}{4}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

$$p(x_i|\, c_2 = 1) = N(x_i|\, \mu_2, \Sigma_2) = \frac{1}{(2\pi)^{\frac{2}{2}} \cdot 4^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2} \cdot \begin{bmatrix} a_1^i + 1 \\ a_2^i + 4 \end{bmatrix}^T \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} a_1^i + 1 \\ a_2^i + 4 \end{bmatrix}\right)$$

	$p(x_n\|c_1 = 1)$	$p(x_n\|c_2 = 1)$	$p(x_n\|c_1 = 1)p(c_1 = 1)$	$p(c_2 = 1\ \|x_n)p(c_2 = 1)$
x_1	$1.5915 * 10^{-1}$	$9.4388 * 10^{-10}$	$1.1114 * 10^{-1}$	$2.8316 * 10^{-10}$
x_2	$2.2391 * 10^{-17}$	$7.9577 * 10^{-2}$	$1.5674 * 10^{-17}$	$2.3873 * 10^{-2}$
x_3	$2.3928 * 10^{-4}$	$9.8206 * 10^{-6}$	$1.6750 * 10^{-4}$	$2.9462 * 10^{-6}$
x_4	$7.2256 * 10^{-6}$	$2.8137 * 10^{-6}$	$5.0579 * 10^{-6}$	$8.4410 * 10^{-7}$

Exemplificação de cálculos para a primeira linha

$p(x_1|c_1 = 1) = N(x_1|\, \mu_1, \Sigma_1) = 1.5915 * 10^{-1}$
$p(x_1|c_1 = 1)p(c_1 = 1) = 1.5915 * 10^{-1} * 0.7 = 1.1114 * 10^{-1}$
$p(x_1|c_2 = 1) = N(x_1|\, \mu_2, \Sigma_2) = 9.4388 * 10^{-10}$
$p(x_1|c_2 = 1)p(c_2 = 1) = 9.4388 * 10^{-10} * 0.3 = 2.8316 * 10^{-10}$

Normalização de *posterioris*

	$p(c_1 = 1\|x_n)$	$p(c_2 = 1\|x_n)$
x_1	0.99999	$2.5417 * 10^{-9}$
x_2	$6.5654 * 10^{-16}$	0.99999
x_3	0.98271	0.01729
x_4	0.85698	0.14302

Exemplificação de cálculos para a primeira linha

$$p(c_1 = 1|x_1) = \frac{1.1114 * 10^{-1}}{1.1114 * 10^{-1} + 2.8316 * 10^{-10}} \approx 0.99999$$

$$p(c_2 = 1|x_1) = \frac{2.8316 * 10^{-10}}{1.1114 * 10^{-1} + 2.8316 * 10^{-10}} \approx 2.5417 * 10^{-9}$$

Cálculo dos parâmetros para os novos clusters

$$\mu_1 = \frac{\left(0.99999 \begin{bmatrix} 2 \\ 4 \end{bmatrix} + 6.5654 * 10^{-16} \begin{bmatrix} -1 \\ -4 \end{bmatrix} + 0.98271 \begin{bmatrix} -1 \\ 2 \end{bmatrix} + 0.85698 \begin{bmatrix} 4 \\ 0 \end{bmatrix}\right)}{0.99999 + 6.5654 * 10^{-16} + 0.98271 + 0.85698} = \begin{bmatrix} 1.5654 \\ 2.1007 \end{bmatrix}$$

$$\mu_2 = \frac{\left(2.5417 * 10^{-9}\begin{bmatrix}2\\4\end{bmatrix} + 0.99999\begin{bmatrix}-1\\-4\end{bmatrix} + 0.01729\begin{bmatrix}-1\\2\end{bmatrix} + 0.14302\begin{bmatrix}4\\0\end{bmatrix}\right)}{2.5417 * 10^{-9} + 0.99999 + 0.01729 + 0.14302} = \begin{bmatrix}-0.3837\\-3.4176\end{bmatrix}$$

$$\Sigma_1 = \frac{\begin{pmatrix} 0.99999\cdot\left(\begin{bmatrix}2\\4\end{bmatrix}-\begin{bmatrix}1.5654\\2.1007\end{bmatrix}\right)\cdot\left(\begin{bmatrix}2\\4\end{bmatrix}-\begin{bmatrix}1.5654\\2.1007\end{bmatrix}\right)^T + 6.5654*10^{-16}\cdot\left(\begin{bmatrix}-1\\-4\end{bmatrix}-\begin{bmatrix}1.5654\\2.1007\end{bmatrix}\right)\cdot\left(\begin{bmatrix}-1\\-4\end{bmatrix}-\begin{bmatrix}1.5654\\2.1007\end{bmatrix}\right)^T \\ +0.98271\cdot\left(\begin{bmatrix}-1\\2\end{bmatrix}-\begin{bmatrix}1.5654\\2.1007\end{bmatrix}\right)\cdot\left(\begin{bmatrix}-1\\2\end{bmatrix}-\begin{bmatrix}1.5654\\2.1007\end{bmatrix}\right)^T + 0.85698\cdot\left(\begin{bmatrix}4\\0\end{bmatrix}-\begin{bmatrix}1.5654\\2.1007\end{bmatrix}\right)\cdot\left(\begin{bmatrix}4\\0\end{bmatrix}-\begin{bmatrix}1.5654\\2.1007\end{bmatrix}\right)^T \end{pmatrix}}{0.99999+6.5654*10^{-16}+0.98271+0.85698} = \begin{bmatrix}4.1328 & -1.1634\\-1.1634 & 2.6056\end{bmatrix}$$

$$\Sigma_2 = \frac{\begin{pmatrix} 2.5417*10^{-9}\cdot\left(\begin{bmatrix}2\\4\end{bmatrix}-\begin{bmatrix}-0.3837\\-3.4176\end{bmatrix}\right)\cdot\left(\begin{bmatrix}2\\4\end{bmatrix}-\begin{bmatrix}-0.3837\\-3.4176\end{bmatrix}\right)^T + 0.99999\cdot\left(\begin{bmatrix}-1\\-4\end{bmatrix}-\begin{bmatrix}-0.3837\\-3.4176\end{bmatrix}\right)\cdot\left(\begin{bmatrix}-1\\-4\end{bmatrix}-\begin{bmatrix}-0.3837\\-3.4176\end{bmatrix}\right)^T \\ +0.01729\cdot\left(\begin{bmatrix}-1\\2\end{bmatrix}-\begin{bmatrix}-0.3837\\-3.4176\end{bmatrix}\right)\cdot\left(\begin{bmatrix}-1\\2\end{bmatrix}-\begin{bmatrix}-0.3837\\-3.4176\end{bmatrix}\right)^T + 0.14302\cdot\left(\begin{bmatrix}4\\0\end{bmatrix}-\begin{bmatrix}-0.3837\\-3.4176\end{bmatrix}\right)\cdot\left(\begin{bmatrix}4\\0\end{bmatrix}-\begin{bmatrix}-0.3837\\-3.4176\end{bmatrix}\right)^T \end{pmatrix}}{2.5417*10^{-9}+0.99999+0.01729+0.14302} = \begin{bmatrix}2.7017 & 2.1062\\2.1062 & 2.1692\end{bmatrix}$$

$$\pi_1 = \frac{0.99999+6.5654*10^{-16}+0.98271+0.85698}{4} = 0.7099$$

$$\pi_2 = \frac{2.5417*10^{-9}+0.99999+0.01729+0.14302}{4} = 0.2901$$

2) De acordo com os valores obtidos na normalização de *posterioris* podemos concluir que os pontos x_1, x_3 e x_4 pertencem aos *cluster* 1 enquanto que o ponto x_2 pertence ao *cluster* 2 devido à maior probabilidade observada para estes *clusters*.

$$s(x_1) = 1 - \frac{1}{2} * \frac{\|x_1 - x_3\| + \|x_1 - x_4\|}{\|x_1 - x_2\|} = 0.5273$$

$$s(x_2) = 0$$

$$s(x_3) = 1 - \frac{1}{2} * \frac{\|x_3 - x_1\| + \|x_3 - x_4\|}{\|x_3 - x_2\|} = 0.2508$$

$$s(x_4) = 1 - \frac{1}{2} * \frac{\|x_4 - x_1\| + \|x_4 - x_3\|}{\|x_4 - x_2\|} = 0.2303$$

$$s(c_1) = \frac{s(x_1) + s(x_3) + s(x_4)}{3} = 0.2594$$

$$s(c_2) = 0$$

$$s(C) = \frac{s(c_1) + s(c_2)}{2} = 0.1297$$

Dado que a o valor da silhueta pode variar entre -1 e 1 e tendo em conta o valor obtido, podemos afirmar que o nível de silhueta não é muito elevado o que significa que os clusters obtidos não se encontram perfeitamente separados e coesos.

3) a)

i) De forma a identificar a dimensão-VC do classificador apresentado iremos identificar o número de parâmetros necessários uma vez que é um bom estimador da dimensão-VC do classificador em questão.

No caso do MLP com 3 camadas internas com tantos nós quantas variáveis de input (5 variáveis neste caso), obtemos que serão necessárias 4 matrizes de pesos e 4 *bias*.

Devido à arquitetura do MLP, quer a matriz de pesos entre a camada de input e a primeira camada interna, quer as matrizes de pesos entre camadas internas serão compostas por 5*5 parâmetros. Já os *bias* para as 3 camadas internas são compostos por 5 elementos.

A matriz de pesos entre a última camada interna e a camada final é composta por 2*5 elementos e o *bias* da camada de output é composto por 2 elementos.

Assim, o número de parâmetros do classificador é dado por $3 * (5 * 5 + 5) + (2 * 5 + 2) = 102$.

Logo a dimensão-VC estimada é de 102.

ii) Para uma árvore de decisão com 5 variáveis discretizadas em 3 *bins*, podemos facilmente notar que no máximo a árvore pode conter 3^5 nós, conseguindo separar, no máximo, 3^5 observações distintas. Logo, a dimensão-VC é de , $3^5 = 243$.

iii) Para o classificador Bayesiano iremos novamente utilizar número de parâmetros para estimar a dimensão-VC.

Para o treino do classificador em questão necessitamos de um dos *priors* (uma vez que o outro pode ser calculado a partir do obtido), de um vetor de 5 elementos com a média para cada uma das 2 classes de output e de uma matriz de covariâncias para cada classe com 5*5 elementos cada. No entanto, devido à simetria da matriz de covariâncias são necessários apenas $\frac{5(5+1)}{2} = 15$ parâmetros para a sua construção completa.

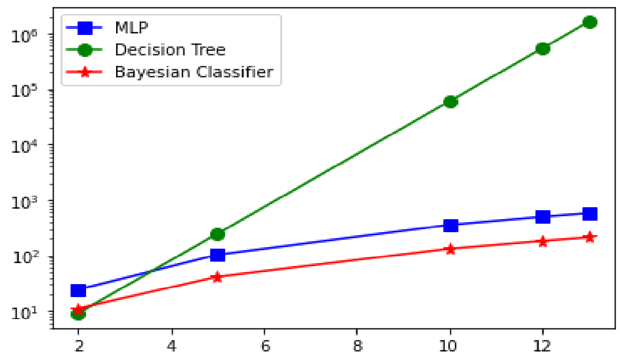
No total, são necessários $2 * \left(5 + \frac{5(5+1)}{2}\right) + 1 = 41$ parâmteros, logo a dimensão-VC aproxima-se de 41

b)

Dimensionalidade	MLP	Decision Tree	Bayesian Classifier
2	24	9	11
5	102	243	41
10	352	59049	131
12	494	531441	181
13	574	159423	209

Fórmulas utilizadas

$$d_{VC_{MLP}} = 3 * (d * d + d) + (2 * d + 2)$$
$$= 3d^2 + 5d + 2$$
$$d_{VC_{Decision\ Tree}} = 3^d$$
$$d_{VC_{Bayesian\ Classifier}} = 1 + 2 * \left(d + \frac{d(d+1)}{2}\right)$$
$$= d^2 + 3d + 1$$



Ao observar o gráfico ao lado podemos observar que a dimensão-VC da árvore de decisão cresce muito mais rapidamente com o aumento da dimensionalidade dos dados que os outros dois classificadores (tal como as fórmulas acima sugeriam).

c)

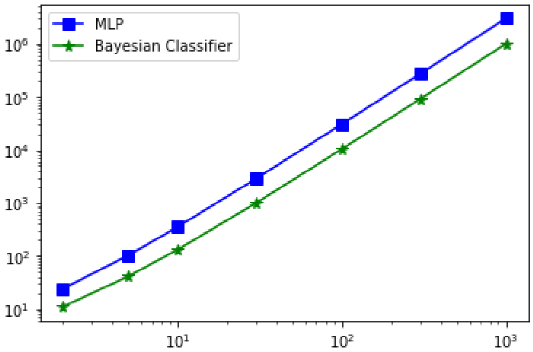
Dimensionalidade	MLP	Bayesian Classifier
2	24	11
5	102	41
10	352	131
30	2852	991
100	30502	10301
300	271502	90901
1000	3005002	1003001

Fórmulas utilizadas

$$d_{VC_{MLP}} = 3 * (d * d + d) + (2 * d + 2) = 3d^2 + 5d + 2$$
$$d_{VC_{Bayesian\ Classifier}} = 1 + 2 * \left(d + \frac{d(d+1)}{2}\right) = d^2 + 3d + 1$$

Aprendizagem 2021/22

Homework IV – Group 9

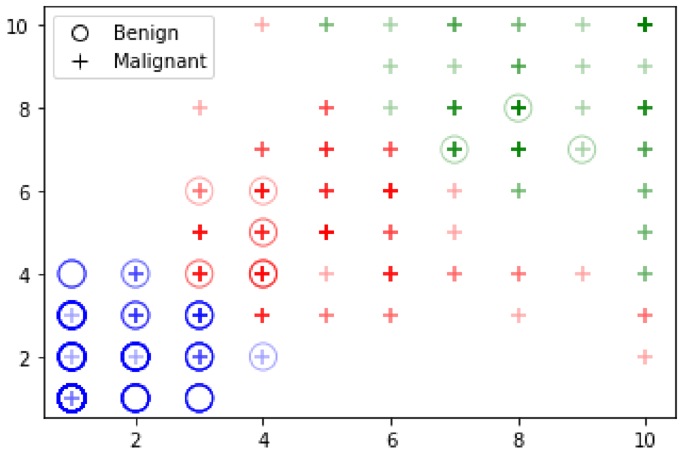


Ao observar o gráfico ao lado podemos observar que a dimensão-VC do MLP é superior à dimensão-VC do classificador Bayesiano apesar de ambos os modelos terem um crescimento semelhante com o aumento da dimensionalidade.

4) a) Ao comparar as soluções produzidas para 2 e 3 *clusters* tendo em conta a medida ECR, verificamos que a solução com 3 *clusters* tem um valor menor (6,666(6)) do que a solução com apenas 2 *clusters* (13.5). Pelo ECR, dado que se trata de uma medida externa podemos concluir que a solução com 3 *clusters* separa melhor as classes das observações.

b) Ao comparar as soluções produzidas para 2 e 3 *clusters* tendo em conta a sua silhueta, verificamos que a solução com 2 *clusters* tem um valor mais próximo de 1 (0.596798) do que a solução com 3 clusters (0.52495). Pela silhueta, dado que se trata de uma medida interna que varia entre -1 e 1 podemos concluir que a solução com 2 clusters forma *clusters* mais bem separados e mais coesos.

5) Representação gráfica da distribuição dos 3 clusters (cada cor corresponde a um cluster) e das classes das observações:



6) Ao observar o gráfico produzido no exercício anterior, podemos observar que os *clusters* formados são bastante coesos uma vez que as observações que lhes pertencem se encontram próximas entre si. No entanto, os mesmos *clusters* não se encontram bem separados uma vez que existem várias observações perto das fronteiras entre eles.

Analisando a distribuição de classes pelos 3 *clusters*, podemos concluir ainda que o *cluster* azul contém maioritariamente observações classificadas como benignas ao contrário dos *clusters* vermelho e verde que contém na sua grande maioria observações classificadas com malignas.

Assim, após esta análise, podemos afirmar que, apesar de os *clusters* formados não se encontrarem bem separados, a seleção de variáveis efetuada resultou numa solução de boa qualidade.

Aprendizagem 2021/22

Homework IV – Group 9

III. APPENDIX

```
import pandas as pd
from scipy.io import arff
from sklearn import cluster
from sklearn.metrics import silhouette_score
import numpy as np

data = arff.loadarff('breast.w.arff')
df = pd.DataFrame(data[0])
df.dropna(inplace=True)
df.replace(b'benign', 0, inplace=True)
df.replace(b'malignant', 1, inplace=True)
data = df.drop(["Class"],axis=1).values
target = df["Class"].values

def ECR(labels, target, k):
    c_counts = []
    for i in range(k):
        ci_filter = []
        for j in range(len(labels)):
            ci_filter.append(labels[j]==i)
        c_counts.append(min(np.bincount(target[ci_filter])))
    error = 0
    for el in c_counts:
        error += el
    return error/k

for k in (2,3):
    cl = cluster.KMeans(n_clusters=k)
    predictions = cl.fit_predict(data)
    print(k, "Clusters")
    print("   ECR          ", ECR(predictions,target,k))
    print("   Silhouette   ", silhouette_score(data,predictions))

from sklearn.feature_selection import SelectKBest, mutual_info_classif
import matplotlib.pyplot as plt
import matplotlib.lines as mlines

data_selected = SelectKBest(mutual_info_classif, k=2).fit_transform(data, target)
cl = cluster.KMeans(n_clusters=3)
pred = cl.fit_predict(data_selected)
clusters = [[[],[]],[[],[]]], [[[],[]],[[],[]]], [[[],[]],[[],[]]]
for i in range(len(data_selected)):
    clusters[pred[i]][target[i]][0].append(data_selected[i].item(0))
    clusters[pred[i]][target[i]][1].append(data_selected[i].item(1))

plt.scatter(clusters[0][0][0],clusters[0][0][1],marker="o",s=200,alpha=0.35,color="none",edgecolors=
"red")
plt.scatter(clusters[0][1][0],clusters[0][1][1],marker="+",s=60 ,alpha=0.35,color="red")
plt.scatter(clusters[1][0][0],clusters[1][0][1],marker="o",s=200,alpha=0.35,color="none",edgecolors=
"blue")
plt.scatter(clusters[1][1][0],clusters[1][1][1],marker="+",s=60 ,alpha=0.35,color="blue")
plt.scatter(clusters[2][0][0],clusters[2][0][1],marker="o",s=200,alpha=0.35,color="none",edgecolors=
"green")
plt.scatter(clusters[2][1][0],clusters[2][1][1],marker="+",s=60 ,alpha=0.35,color="green")
leg1 = mlines.Line2D([], [], marker='o',color="none",markeredgecolor="black",
                          markersize=8, label='Benign')
leg2 = mlines.Line2D([], [], marker='+',color="none",markeredgecolor="black",
                          markersize=8, label='Malignant')
plt.legend(handles=[leg1,leg2])
```

END