Aprendizagem 2023/24

Homework IV – Group 004

(ist102695, ist102703)

I. Pen-and-paper

1)

$$\left\{ \begin{pmatrix} 1 \\ 0,6 \\ 0,1 \end{pmatrix}, \begin{pmatrix} 0 \\ -0,4 \\ 0,3 \end{pmatrix}, \begin{pmatrix} 0 \\ 0,2 \\ 0,5 \end{pmatrix}, \begin{pmatrix} 1 \\ 0,4 \\ -0,1 \end{pmatrix} \right\} \quad \pi_1 = 0,5 \quad \pi_2 = 0,5$$

$$\mathcal{N}_1\left(\mu_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 2 & 0,5 \\ 0,5 & 2 \end{pmatrix}\right) \quad \mathcal{N}_2\left(\mu_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 1,5 & 1 \\ 1 & 1,5 \end{pmatrix}\right)$$

1)

$$x_1 = \begin{pmatrix} 1 \\ 0,6 \\ 0,1 \end{pmatrix} \quad x_2 = \begin{pmatrix} 0 \\ -0,4 \\ 0,3 \end{pmatrix} \quad x_3 = \begin{pmatrix} 0 \\ 0,2 \\ 0,5 \end{pmatrix} \quad x_4 = \begin{pmatrix} 1 \\ 0,4 \\ -0,1 \end{pmatrix}$$

Cluster 1 $\rightarrow \mu = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \Sigma = \begin{pmatrix} 2 & 0,5 \\ 0,5 & 2 \end{pmatrix}, \pi = 0,5$

Cluster 2 $\rightarrow \mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma = \begin{pmatrix} 1,5 & 1 \\ 1 & 1,5 \end{pmatrix}, \pi = 0,5$

E-step

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

Aprendizagem 2023/24

Homework IV – Group 004

(ist102695, ist102703)

x_1

c_1 $p(c_1|x_1) = P(y_1=1|c_1) \times p(y_2, y_3|c_1) \times \pi_1 =$

$= p_1 \times N\left(\begin{bmatrix}0,6\\0,1\end{bmatrix} \middle| \mu = \begin{bmatrix}1\\1\end{bmatrix}, \Sigma = \begin{bmatrix}2 & 0,5\\0,5 & 2\end{bmatrix}\right) \times \pi_1 =$

$= 0,3 \times 0,066575 \times 0,5 = 0,009986$

c_2 $p(c_2|x_1) = P(y_1=1|c_2) \times p(y_2, y_3|c_2) \times \pi_2 =$

$= p_2 \times N\left(\begin{bmatrix}0,6\\0,1\end{bmatrix} \middle| \mu = \begin{bmatrix}0\\0\end{bmatrix}, \Sigma = \begin{bmatrix}1,5 & 1\\1 & 1,5\end{bmatrix}\right) \times \pi_2 =$

$= 0,7 \times 0,11962 \times 0,5 = 0,041867$

$$\gamma(c_{11}) = \frac{p(c_1|x_1)}{\underbrace{p(c_1|x_1) + p(c_2|x_1)}_{p(x_1)}} = 0,19258 \qquad \gamma(c_{12}) = \frac{p(c_2|x_1)}{p(c_1|x_1) + p(c_2|x_1)} = 0,80742$$

x_2

c_1 $p(c_1|x_2) = P(y_1=0|c_1) \times p(y_2, y_3|c_1) \times \pi_1 =$

$= (1-p_1) \times N\left(\begin{bmatrix}-0,4\\0,8\end{bmatrix} \middle| \mu = \begin{bmatrix}1\\1\end{bmatrix}, \Sigma = \begin{bmatrix}2 & 0,5\\0,5 & 2\end{bmatrix}\right) \times \pi_1 =$

$= 0,7 \times 0,050049 \times 0,5 = 0,017517$

c_2 $p(c_2|x_2) = P(y_1=0|c_2) \times p(y_2, y_3|c_2) \times \pi_2 =$

$= (1-p_2) \times N\left(\begin{bmatrix}-0,4\\0,8\end{bmatrix} \middle| \mu = \begin{bmatrix}0\\0\end{bmatrix}, \Sigma = \begin{bmatrix}1,5 & 1\\1 & 1,5\end{bmatrix}\right) \times \pi_2 =$

$= 0,3 \times 0,068191 \times 0,5 = 0,010229$

$$\gamma(c_{21}) = \frac{p(c_1|x_2)}{p(c_1|x_2) + p(c_2|x_2)} = 0,6313 \qquad \gamma(c_{22}) = \frac{p(c_2|x_2)}{p(c_1|x_2) + p(c_2|x_2)} = 0,3687$$

TÉCNICO LISBOA

Aprendizagem 2023/24

Homework IV – Group 004

(ist102695, ist102703)

x_3

c_1 $p(c_1|x_3) = P(y_1=0|c_1) \times P(y_2,y_3|c_1) \times \pi_1 =$

$= (1-p_1) \times N\left(\begin{bmatrix}0,2\\0,5\end{bmatrix} \middle| \mu = \begin{bmatrix}1\\1\end{bmatrix}, \Sigma = \begin{bmatrix}2 & 0,5\\0,5 & 2\end{bmatrix}\right) \times \pi_1 =$

$= 0,7 \times 0,068377 \times 0,5 = 0,023930$

c_2 $p(c_2|x_3) = P(y_1=0|c_2) \times P(y_2,y_3|c_2) \times \pi_2 =$

$= (1-p_2) \times N\left(\begin{bmatrix}0,2\\0,5\end{bmatrix} \middle| \mu = \begin{bmatrix}0\\0\end{bmatrix}, \Sigma = \begin{bmatrix}1,5 & 1\\1 & 1,5\end{bmatrix}\right) \times \pi_2 =$

$= 0,3 \times 0,112958 \times 0,5 = 0,019437$

$\gamma(c_{31}) = \frac{p(c_1|x_3)}{p(c_1|x_3)+p(c_2|x_3)} = 0,5518$ $\gamma(c_{32}) = \frac{p(c_2|x_3)}{p(c_1|x_3)+p(c_2|x_3)} = 0,4482$

x_4

c_1 $p(c_1|x_4) = P(y_1=1|c_1) \times P(y_2,y_3|c_1) \times \pi_1 =$

$= p_1 \times N\left(\begin{bmatrix}0,4\\-0,1\end{bmatrix} \middle| \mu = \begin{bmatrix}1\\1\end{bmatrix}, \Sigma = \begin{bmatrix}2 & 0,5\\0,5 & 2\end{bmatrix}\right) \times \pi_1 =$

$= 0,3 \times 0,059047 \times 0,5 = 0,008857$

c_2 $p(c_2|x_4) = P(y_1=1|c_2) \times P(y_2,y_3|c_2) \times \pi_2 =$

$= p_2 \times N\left(\begin{bmatrix}0,4\\-0,1\end{bmatrix} \middle| \mu = \begin{bmatrix}0\\0\end{bmatrix}, \Sigma = \begin{bmatrix}1,5 & 1\\1 & 1,5\end{bmatrix}\right) \times \pi_2 =$

$= 0,7 \times 0,1245 \times 0,5 = 0,043575$

$\gamma(c_{41}) = \frac{p(c_1|x_4)}{p(c_1|x_4)+p(c_2|x_4)} = 0,1689$ $\gamma(c_{42}) = \frac{p(c_2|x_4)}{p(c_1|x_4)+p(c_2|x_4)} = 0,8311$

TÉCNICO LISBOA

Aprendizagem 2023/24

Homework IV – Group 004

(ist102695, ist102703)

M-Step

$$N_1 = \sum_{n=1}^{4} \gamma(c_{n1}) = 0,19258 + 0,6313 + 0,55518 + 0,1689 = 1,54458$$

$$N_2 = \sum_{n=1}^{4} \gamma(c_{n2}) = 0,80742 + 0,3687 + 0,44482 + 0,8311 = 2,45542$$

$$\mu_1 = \frac{1}{N_1} \times \sum_{n=1}^{4} \left(\gamma(c_{n1}) \times x_n \right) = \frac{1}{1,54458} \times \sum_{n=1}^{4} \left(\gamma(c_{n1}) \times x_n \right) = \begin{pmatrix} 0,2340 \\ 0,02651 \\ 0,5071 \end{pmatrix}$$

$$\mu_2 = \frac{1}{N_2} \times \sum_{n=1}^{4} \left(\gamma(c_{n2}) \times x_n \right) = \frac{1}{2,45542} \times \sum_{n=1}^{4} \left(\gamma(c_{n2}) \times x_n \right) = \begin{pmatrix} 0,6673 \\ 0,3091 \\ 0,2104 \end{pmatrix}$$

$$\Sigma_1 = \frac{1}{N_1} \left(\sum_{n=1}^{4} \left(\gamma(c_{n1}) \times (x_n - \mu_1) \times (x_n - \mu_1)^T \right) \right) =$$

$$= \frac{1}{1,54458} \left(\begin{pmatrix} 0,06334 & -0,04496 \\ -0,04496 & 0,03192 \end{pmatrix} + \begin{pmatrix} 0,11484 & -0,07885 \\ -0,07885 & 0,05414 \end{pmatrix} + \begin{pmatrix} 0,01060 & -0,00068 \\ -0,00068 & 0,00002 \end{pmatrix} + \begin{pmatrix} 0,02356 & -0,03829 \\ -0,03829 & 0,06225 \end{pmatrix} \right) =$$

$$= \begin{pmatrix} 0,14136 & -0,10539 \\ -0,10539 & 0,09601 \end{pmatrix}$$

$$\Sigma_2 = \frac{1}{N_2} \left(\sum_{n=1}^{4} \left(\gamma(c_{n2}) \cdot (x_n - \mu_2) \cdot (x_n - \mu_2)^T \right) \right) =$$

$$= \frac{1}{2,45542} \left(\begin{pmatrix} 0,06831 & -0,02593 \\ -0,02593 & 0,00483 \end{pmatrix} + \begin{pmatrix} 0,1854 & -0,1541 \\ -0,1541 & 0,12815 \end{pmatrix} + \begin{pmatrix} 0,00533 & -0,01416 \\ -0,01416 & 0,03758 \end{pmatrix} + \begin{pmatrix} 0,00656 & -0,02344 \\ -0,02344 & 0,08069 \end{pmatrix} \right) =$$

$$= \begin{pmatrix} 0,10829 & -0,08865 \\ -0,08865 & 0,10412 \end{pmatrix}$$

$$\pi_1 = \frac{N_1}{N_1 + N_2} = \frac{1,54458}{1,54458 + 2,45542} = 0,386145$$

$$\pi_2 = \frac{N_2}{N_1 + N_2} = \frac{2,45542}{1,54458 + 2,45542} = 0,613855$$

Homework IV – Group 004

(ist102695, ist102703)

2)

2-

$$x_{new} = \begin{pmatrix} 1 \\ 0,3 \\ 0,7 \end{pmatrix}$$

(c1)

$$P(c_1 | x_{new}) = P(y_1 = 1 | c_1) \cdot P(y_2, y_3 | c_1)\, \pi_2 =$$

$$= 0,2340 \times N\left(\begin{bmatrix} 0,3 \\ 0,7 \end{bmatrix} \middle| M_1 = \begin{pmatrix} 0,02651 \\ 0,5071 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 0,14136 & -0,10539 \\ -0,10539 & 0,09601 \end{pmatrix} \right) \times 0,376145 =$$

$$= 0,002434$$

(c2)

$$P(c_2 | x_{new}) = P(y_1 = 1 | c_2) \cdot P(y_2, y_3 | c_2)\, \pi_2 =$$

$$= 0,6673 \times N\left(\begin{bmatrix} 0,3 \\ 0,7 \end{bmatrix} \middle| M_2 = \begin{pmatrix} 0,3091 \\ 0,2104 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 0,10829 & -0,08865 \\ -0,08865 & 0,10412 \end{pmatrix} \right) \times 0,613855 =$$

$$= 0,028045$$

$$\gamma(C_{1new}) = \frac{P(C_1 | x_{new})}{P(C_1 | x_{new}) + P(C_2 | x_{new})} = 0,079859$$

$$\gamma(C_{2new}) = \frac{P(C_2 | x_{new})}{P(C_1 | x_{new}) + P(C_2 | x_{new})} = 0,92014$$

Homework IV – Group 004

(ist102695, ist102703)

3)

$x_2 \rightarrow$ cluster 1

$x_3 \rightarrow$ cluster 1

$x_1 \rightarrow$ cluster 2

$x_4 \rightarrow$ cluster 2

$$d(x_1, x_2) = |x_{11} - x_{21}| + |x_{12} - x_{22}| + |x_{13} - x_{23}|$$

$$= 2,7$$

$d(x_1, x_3) = 1,8$ $\quad$ $d(x_2, x_3) = 0,9$

$d(x_1, x_4) = 0,4$ $\quad$ $d(x_2, x_4) = 2,7$

$d(x_3, x_4) = 1,8$

x_1:

$$a(x_1) = d(x_1, x_4) = 0.4 \rightarrow \text{menor valor}$$

$$b(x_1) = \frac{d(x_1, x_2) + d(x_1, x_3)}{2}$$

$$= 2,25$$

$$\therefore S(x_1) = 1 - \frac{0.4}{2.25} = 0.822$$

x_2:

$$a(x_2) = 0.9 \quad b(x_2) = 2.7 \quad S(x_2) = 1 - \frac{0.9}{2.7} = 0.667$$

x_3:

$a(x_3) = 0.9$ $\quad b(x_3) = 1.8$ $\quad S(x_3) = 1 - \frac{0.9}{1.8} = 0.5$

x_4:

$a(x_4) = 0.4$ $\quad b(x_4) = 2.25$ $\quad S(x_4) = 1 - \frac{0.4}{2.25} = 0.822$

4)

$$\text{purity} = \frac{1}{N} \sum_{k}^{\hat{}} \text{argmax} \, |C_k \cap L_j|$$

$$0.75 = \frac{1}{4} \sum_{k}^{2} \text{argmax} \, |C_k \cap L_j|$$

$$\sum_{k}^{2} \text{argmax} \, |C_k \cap L_j| = 3$$

one class

$$\text{argmax}(2) + \text{argmax}(2) = 4 \neq 3$$

two classes

$$\text{argmax}(0,2) + \text{argmax}(1,1) = 2+1 = 3 = 3 \checkmark$$

Therefore, there are two possible classes.

II. Programming and critical analysis

1)

```
#1
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.metrics.cluster import contingency_matrix
from scipy.stats import mode
from sklearn.datasets import fetch_openml
from scipy.io import arff


data, meta = arff.loadarff('column_diagnosis.arff')

df = pd.DataFrame(data)

features = df.drop('class', axis=1)
labels = df['class']

# Normalize the data using MinMaxScaler
scaler = MinMaxScaler()
normalized_features = scaler.fit_transform(features)

# Define values of k
k_values = [2, 3, 4, 5]

# Perform k-means clustering for different values of k
for k in k_values:
    # Initialize k-means object with explicit n_init parameter
    kmeans = KMeans(n_clusters=k, random_state=0, n_init=10)

    # Fit the k-means model to the normalized data
```

```
cluster_labels = kmeans.fit_predict(normalized_features)

# Calculate silhouette score
silhouette_avg = silhouette_score(normalized_features, cluster_labels)

# Calculate purity
contingency = contingency_matrix(labels, cluster_labels)
purity = np.sum(np.max(contingency, axis=0)) / np.sum(contingency)

# Output results
print(f'Number of clusters: {k}')
print(f'Silhouette Score: {silhouette_avg}')
print(f'Purity: {purity}\n')
```

```
Number of clusters: 2
Silhouette Score: 0.3604412434044111
Purity: 0.632258064516129

Number of clusters: 3
Silhouette Score: 0.2957905573000225
Purity: 0.667741935483871

Number of clusters: 4
Silhouette Score: 0.27442402122340176
Purity: 0.6612903225806451

Number of clusters: 5
Silhouette Score: 0.23823928397844843
Purity: 0.6774193548387096
```

2)

```
#2
import numpy as np
from scipy.io import arff
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
```

```
data, meta = arff.loadarff('column_diagnosis.arff')


features = meta.names()[:-1]


data_array = np.array(data[features].tolist())

# Initialize the MinMaxScaler
scaler = MinMaxScaler()

# Normalize the data
normalized_data = scaler.fit_transform(data_array)

# Initialize PCA with 2 components
pca = PCA(n_components=2)

pca_result = pca.fit_transform(normalized_data)

explained_variance_ratio = pca.explained_variance_ratio_

print("Variance explained by the top two components:
{:.2f}%".format(sum(explained_variance_ratio) * 100))

weights = np.abs(pca.components_.T)

# Sort input variables by relevance for the first principal component
sorted_variables_pc1 = sorted(list(zip(features, weights[:, 0])), key=lambda x: abs(x[1]),
reverse=True)

# Sort input variables by relevance for the second principal component
sorted_variables_pc2 = sorted(list(zip(features, weights[:, 1])), key=lambda x: abs(x[1]),
reverse=True)

# Print the sorted variables for the first principal component
```

```
print("Sorted variables for the first principal component:")
for variable, weight in sorted_variables_pc1:
    print("{}: {:.4f}".format(variable, weight))

# Print the sorted variables for the second component
print("\nSorted variables for the second principal component:")
for variable, weight in sorted_variables_pc2:
    print("{}: {:.4f}".format(variable, weight))
```

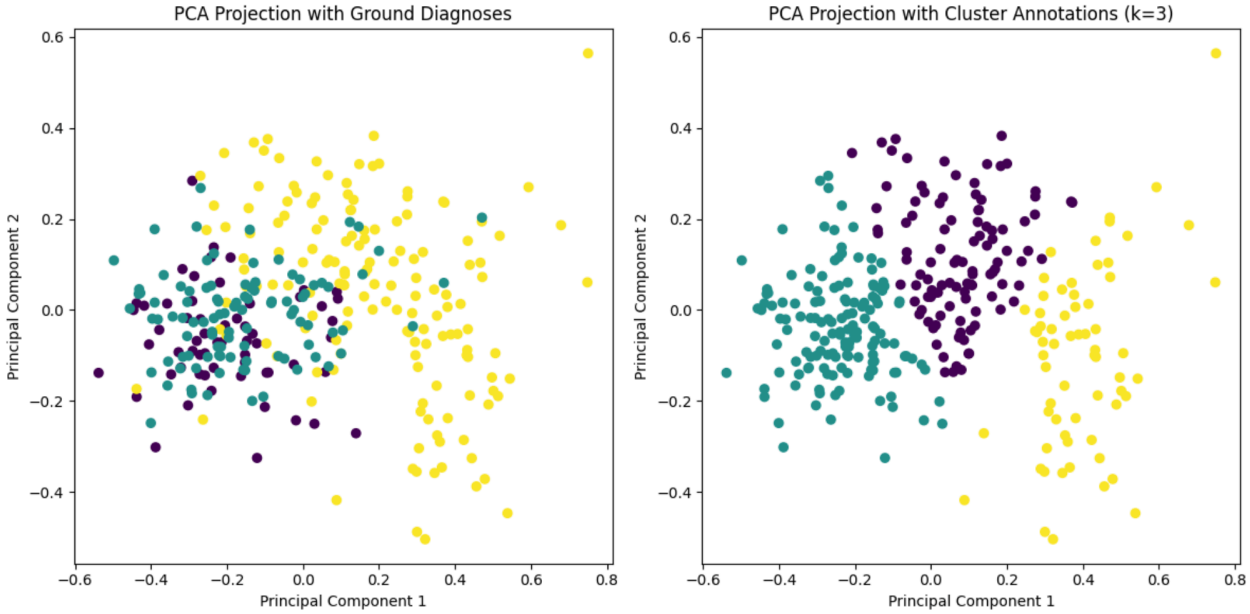
```
Variance explained by the top two components: 77.14%
Sorted variables for the first principal component:
pelvic_incidence: 0.5916
lumbar_lordosis_angle: 0.5151
pelvic_tilt: 0.4670
sacral_slope: 0.3257
degree_spondylolisthesis: 0.2169
pelvic_radius: 0.1158

Sorted variables for the second principal component:
pelvic_tilt: 0.6704
pelvic_radius: 0.5811
sacral_slope: 0.4433
pelvic_incidence: 0.1000
lumbar_lordosis_angle: 0.0800
degree_spondylolisthesis: 0.0046
```

3)

```
#3

import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from scipy.io import arff
from io import StringIO
```

```
data, meta = arff.loadarff('column_diagnosis.arff')
data = pd.DataFrame(data)


features = data.drop(columns=['class'])


scaler = MinMaxScaler()
normalized_data = scaler.fit_transform(features)


pca = PCA(n_components=2)
pca_result = pca.fit_transform(normalized_data)


kmeans = KMeans(n_clusters=3, random_state=42, n_init=10)
cluster_labels = kmeans.fit_predict(normalized_data)

# Plot the PCA projection with ground diagnoses and cluster annotations
plt.figure(figsize=(12, 6))

# Plot for ground diagnoses
plt.subplot(1, 2, 1)
plt.scatter(pca_result[:, 0], pca_result[:, 1], c=data['class'].astype('category').cat.codes,
cmap='viridis')
plt.title('PCA Projection with Ground Diagnoses')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')

# Plot for k=3 clustering solution
plt.subplot(1, 2, 2)
plt.scatter(pca_result[:, 0], pca_result[:, 1], c=cluster_labels, cmap='viridis')
plt.title('PCA Projection with Cluster Annotations (k=3)')
```

```
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')

plt.tight_layout()
plt.show()
```



4)

Algoritmos de agrupamento, conforme mostrado na pergunta 1, podem ser usados para identificar subgrupos distintos dentro da população. Por exemplo agrupar com k=2 produziu o maior Silhouette Score, indicando clusters bem definidos. Esses clusters podem representar diferentes estados de saúde, como “Doente” e “Saudável”. Ao examinar as atribuições e características dos clusters, os profissionais de saúde podem obter insights sobre as características que diferenciam estes subgrupos. Esta informação pode ser valiosa para adaptar tratamentos, intervenções ou estratégias de monitorização específicas para cada subgrupo.

O clustering pode ser utilizado como uma forma de deteção de anomalias. Quando os dados de um novo indivíduo são adicionados ao conjunto de dados, os algoritmos de clustering podem atribuir o indivíduo a um dos clusters existentes. Se o indivíduo for significativamente diferente dos padrões típicos dentro do grupo atribuído, isso pode indicar um potencial problema de saúde. Este método pode ser particularmente útil para o diagnóstico precoce. Ao monitorizar continuamente novos dados e agrupar indivíduos, os prestadores de cuidados de saúde podem identificar desvios da norma, permitindo potencialmente uma intervenção precoce e medidas proactivas de cuidados de saúde tanto para indivíduos doentes como saudáveis.

END