Aprendizagem 2022

Homework IV – Group 126

Part I: Pen and paper

1. Given the following observations:

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

i Initialization

$$\mu_1 = \begin{pmatrix}2\\2\end{pmatrix},\ \mu_2 = \begin{pmatrix}0\\0\end{pmatrix},\ \Sigma_1 = \begin{pmatrix}2 & 1\\1 & 2\end{pmatrix},\ \Sigma_2 = \begin{pmatrix}2 & 0\\0 & 2\end{pmatrix},\ \pi_1 = 0.5,\ \pi_2 = 0.5$$

ii Expectation

First we had to calculate the normal destribution for each data point x_i for each cluster c_k using:

$$p(x_i|c_k) = N(x_i|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)} \tag{1}$$

And got:

	c_1	c_2
x_1	0.0658	0.0228
x_2	0.0089	0.0483
x_3	0.0338	0.0620

Then we calculated the weight for each data point x_i for each cluster c_k using:

$$\gamma_{ki} = \frac{N(x_i|\mu_k, \Sigma_k) \cdot \pi_k}{\sum_{j=1}^{k} \pi_j N(x_i|\mu_j, \Sigma_j)} = \frac{N(x_i|\mu_k, \Sigma_k) \cdot \pi_k}{\pi_1 N(x_i|\mu_1, \Sigma_1) + \pi_2 N(x_i|\mu_2, \Sigma_2)} \tag{2}$$

And got:

	c_1	c_2
x_1	0.7428	0.2572
x_2	0.1558	0.8442
x_3	0.3529	0.6471

iii Maximization

Each observation x_i will contribute to update cluster c_k with weight γ_{ki}

$$N_k = \sum_{i=1}^{n} \gamma_{ki} \tag{3}$$

$$N_1 \simeq 1.2516 \qquad N_2 \simeq 1.7484$$

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{n} \gamma_{ki} \cdot x_i \tag{4}$$

$$\mu_1 \simeq \begin{pmatrix} 0.7510 \\ 0.9388 \end{pmatrix} \qquad \mu_2 \simeq \begin{pmatrix} 0.0480 \\ 0.7770 \end{pmatrix}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{n} \gamma_{ki} \cdot (x_i - \mu_k) \cdot (x_i - \mu_k)^T \tag{5}$$

$$\Sigma_1 \simeq \begin{pmatrix} 0.4361 & 0.0776 \\ 0.0776 & 0.9174 \end{pmatrix} \qquad \Sigma_2 \simeq \begin{pmatrix} 0.9990 & -0.2153 \\ -0.2153 & 0.4675 \end{pmatrix}$$

$$\pi_k = p(c_k) = \frac{N_k}{N} \tag{6}$$

$$\pi_1 \simeq 0.4172 \qquad \pi_2 \simeq 0.5828$$

2. (a) To perform an hard assignment we do the expansion step again, using the updated parameters:

$$\mu_1 = \begin{pmatrix} 0.7510 \\ 0.9388 \end{pmatrix}, \qquad \Sigma_1 = \begin{pmatrix} 0.4361 & 0.0776 \\ 0.0776 & 0.9174 \end{pmatrix}, \qquad \pi_1 \simeq 0.4172$$

$$\mu_2 = \begin{pmatrix} 0.0480 \\ 0.7770 \end{pmatrix}, \qquad \Sigma_2 = \begin{pmatrix} 0.9990 & -0.2153 \\ -0.2153 & 0.4675 \end{pmatrix}, \qquad \pi_2 \simeq 0.5828$$

And obtain:

	c_1	c_2
x_1	0.8733	0.1267
x_2	0.0341	0.9659
x_3	0.4836	0.5164

From this results we can assign x_1 to c_1 and x_2, x_3 to c_2.

(b) To compute the silhouette of the lerger cluster we have to compute the average of the silhouettes of the larger cluster's observations.

$$s_{cluster} = \frac{s_2 + s_3}{2} \tag{7}$$

$$s_{observation} = 1 - \frac{a}{b}, \text{ if } a < b \quad \text{or} \quad s_{observation} = \frac{b}{a} - 1, \text{ if } a \geqslant b \tag{8}$$

Since there are only two points in our cluster, a is the same for both:

$$a = \sqrt{(y_{1_2} - y_{1_3})^2 + (y_{2_2} - y_{2_3})^2} = \sqrt{(-2)^2 + (1)^2} = \sqrt{5}$$

As there is only one point in the other cluster:

- For x_2:

$$b = \sqrt{(y_{1_2} - y_{1_1})^2 + (y_{2_2} - y_{2_1})^2} = \sqrt{(-2)^2 + (-1)^2} = \sqrt{5}$$

- For x_3:

$$b = \sqrt{(y_{1_3} - y_{1_1})^2 + (y_{2_3} - y_{2_1})^2} = \sqrt{(0)^2 + (-2)^2} = 2$$

Using the silhouette formula(8) for $a \geqslant b$:

$$s_2 = \frac{\sqrt{5}}{\sqrt{5}} - 1 = 0 \qquad s_3 = \frac{2}{\sqrt{5}} - 1 \simeq -0.1056$$

We can now compute the cluster silhouette(7):

$$s_{cluster} = \frac{0 - 0.1056}{2} \simeq -0.0528$$

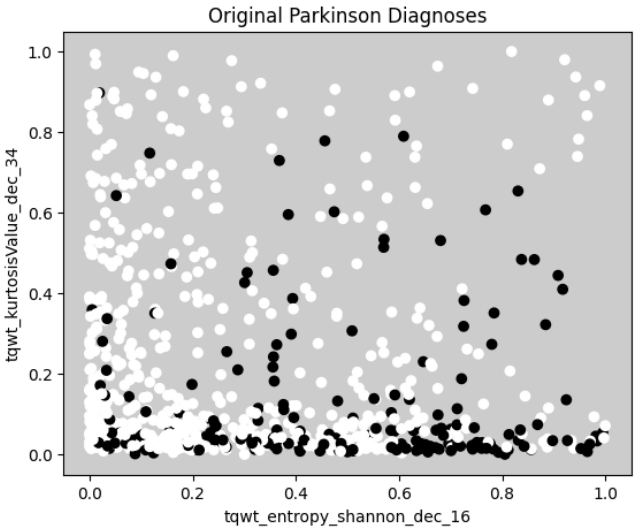
Part II: Programming

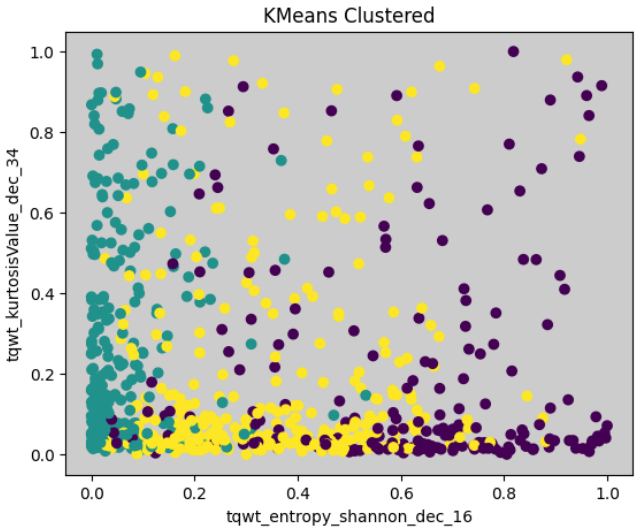
1.

Seed	Silhouette	Purity
0	0.1136	0.7672
1	0.1140	0.7632
2	0.1136	0.7672

2. As we use different seeds, different initializations cause different results.

3.





4. 31 principal components are necessary to explain more than 80% of variability.

APPENDIX

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.io.arff import loadarff
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from sklearn.metrics import cluster, silhouette_score
from sklearn.feature_selection import VarianceThreshold
from sklearn.decomposition import PCA

# Load Data
data = loadarff('../data/pd_speech.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')
X = df.iloc[:, :-1]
y = df.iloc[:, -1]
```

```
# Normalization
scaler = MinMaxScaler()
X = pd.DataFrame(scaler.fit_transform(X), columns=X.columns)

# Purity score
def purity_score(y_true, y_pred):
    confusion_matrix = cluster.contingency_matrix(y_true, y_pred)
    return np.sum(np.amax(confusion_matrix, axis=0)) / np.sum(confusion_matrix)

# K-means
y_pred = []
for seed in [2, 1, 0]:
    kmeans = KMeans(n_clusters=3, random_state=seed).fit(X)
    y_pred = kmeans.labels_
    print(seed, 'Silhouette:', silhouette_score(X, y_pred))
    print(seed, 'Purity:', purity_score(y, y_pred))

# Feature Selection
variances = VarianceThreshold().fit(X).variances_
variances.sort()
selector = VarianceThreshold(threshold=variances[-3]).fit(X)
X_new = pd.DataFrame(selector.transform(X),
                     columns=X.columns[selector.get_support()])

# Plot
fig, axes = plt.subplot_mosaic("AB")
fig.set_size_inches(14, 5)
axes['A'].scatter(X_new.iloc[:, 0], X_new.iloc[:, 1], c=y)
axes['A'].set_title('Original Parkinson Diagnoses')
axes['A'].set_xlabel(X_new.columns[0])
axes['A'].set_ylabel(X_new.columns[1])
axes['A'].set_facecolor('#CCCCCC')
axes['B'].scatter(X_new.iloc[:, 0], X_new.iloc[:, 1], c=y_pred)
axes['B'].set_title('KMeans Clustered')
axes['B'].set_xlabel(X_new.columns[0])
axes['B'].set_ylabel(X_new.columns[1])
axes['B'].set_facecolor('#CCCCCC')
plt.show()

# PCA
components = 0
total_variance = 0
while (total_variance <= 0.8):
    components += 1
    pca = PCA(n_components=components).fit(X)
    total_variance = np.sum(pca.explained_variance_ratio_)
print('# Components =', components)
```