Aprendizagem 2022/23

Homework III– Group 104

I. Pen-and-paper

1) Please see Appendix (1).

I. Pen-and-paper

$$X = \left\{ \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\} \Rightarrow x_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix},\ x_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix},\ x_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

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

1)

1.1) EXPECTATION (E-STEP) → ATRIBUIÇÃO

Recordar:

- $\gamma_{ik} = P(C_k \mid x_i) = \dfrac{P(x_i \mid C_k)P(C_k)}{P(x_i)} = P(x_i \mid C_k)P(C_k) = N(x_i \mid \mu_k, \Sigma_k) \cdot \pi_k$
- $N(x_i \mid \mu_k, \Sigma_k) = \dfrac{1}{(2\pi)^{\frac{m}{2}} |\Sigma|^{1/2}} e^{-\frac{1}{2}\left((x_i-\mu_k)^T \Sigma^{-1}(x_i-\mu_k)\right)}$

Cálculos:

- $\gamma_{11} = P(C_1 \mid x_1) = P(x_1 \mid C_1)P(C_1) = N(x_1 \mid \mu_1, \Sigma_1) \cdot \pi_1$

$$= N\left(\begin{pmatrix} 1 \\ 2 \end{pmatrix} \middle| \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\right) \times 0.5 = \frac{1}{(2\pi)^{\frac{2}{2}} \left|\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\right|^{1/2}} e^{-\frac{1}{2}\left(\begin{pmatrix} 1 \\ 2 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \end{pmatrix}\right)^T \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}^{-1} \left(\begin{pmatrix} 1 \\ 2 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \end{pmatrix}\right)} \times 0.5$$

$= 0.032920$

Pela mesma lógica,

- $\gamma_{21} = P(C_1 \mid x_2) = P(x_2 \mid C_1)P(C_1) = N(x_2 \mid \mu_1, \Sigma_1) \cdot \pi_1 = 0.004455$
- $\gamma_{31} = P(C_1 \mid x_3) = P(x_3 \mid C_1)P(C_1) = N(x_3 \mid \mu_1, \Sigma_1) \cdot \pi_1 = 0.016902$
- $\gamma_{12} = P(C_2 \mid x_1) = P(x_1 \mid C_2)P(C_2) = N(x_1 \mid \mu_2, \Sigma_2) \cdot \pi_2 = 0.011400$
- $\gamma_{22} = P(C_2 \mid x_2) = P(x_2 \mid C_2)P(C_2) = N(x_2 \mid \mu_2, \Sigma_2) \cdot \pi_2 = 0.024133$
- $\gamma_{32} = P(C_2 \mid x_3) = P(x_3 \mid C_2)P(C_2) = N(x_3 \mid \mu_2, \Sigma_2) \cdot \pi_2 = 0.030987$

Aprendizagem 2022/23
Homework III– Group 104

Normalizando, agora, os valores obtidos anteriormente,

Recordar: (normalização)

$$\boxed{\gamma_{ik} = P(C_k|x_i) = \frac{P(C_k|x_i)}{\sum_j P(C_j|x_i)}}$$

- $\gamma_{11} = \dfrac{P(C_1|x_1)}{P(C_1|x_1)+P(C_2|x_1)} = 0.742788$
- $\gamma_{12} = \dfrac{P(C_2|x_1)}{P(C_2|x_1)+P(C_1|x_1)} = 0.257212$
- $\gamma_{21} = \dfrac{P(C_1|x_2)}{P(C_1|x_2)+P(C_2|x_2)} = 0.155843$
- $\gamma_{22} = \dfrac{P(C_2|x_2)}{P(C_2|x_2)+P(C_1|x_2)} = 0.844157$
- $\gamma_{31} = \dfrac{P(C_1|x_3)}{P(C_1|x_3)+P(C_2|x_3)} = 0.352936$
- $\gamma_{32} = \dfrac{P(C_2|x_3)}{P(C_2|x_3)+P(C_1|x_3)} = 0.647064$

Aprendizagem 2022/23

Homework III– Group 104

1.2) MAXIMIZATION (M-STEP) → ATUALIZAÇÃO

Recordar:

- $N_K = \sum_{i=1}^{m} \gamma_{ik}$
- $\mu_K = \frac{1}{N_K} \sum_{i=1}^{m} \gamma_{ik} x_i$
- $\Sigma_K = \frac{1}{N_K} \sum_{i=1}^{m} \gamma_{ik} (x_i - \mu_K) \cdot (x_i - \mu_K)^T$
- $\pi_K = P(C_K) = \frac{N_K}{N}$

- $N_1 = \sum_{i=1}^{3} \gamma_{i1} = \gamma_{11} + \gamma_{21} + \gamma_{31} = 0.742788 + 0.155843 + 0.352936 = 1.251566$
- $N_2 = \sum_{i=1}^{3} \gamma_{i2} = \gamma_{12} + \gamma_{22} + \gamma_{32} = 0.257212 + 0.844157 + 0.647064 = 1.748434$
- $\mu_1 = \frac{1}{N_1} \sum_{i=1}^{3} \gamma_{i1} x_i = \frac{1}{N_1} \left(\gamma_{11} x_1 + \gamma_{21} x_2 + \gamma_{31} x_3 \right) = \begin{pmatrix} 0.750964 \\ 1.311491 \end{pmatrix}$
- $\mu_2 = \frac{1}{N_2} \sum_{i=1}^{3} \gamma_{i2} x_i = \frac{1}{N_2} \left(\gamma_{12} x_1 + \gamma_{22} x_2 + \gamma_{32} x_3 \right) = \begin{pmatrix} 0.034385 \\ 0.777028 \end{pmatrix}$
- $\Sigma_1 = \frac{1}{N_1} \sum_{i=1}^{3} \gamma_{i1} (x_i - \mu_1) \cdot (x_i - \mu_1)^T = \begin{pmatrix} 0.436053 & 0.077573 \\ 0.077573 & 0.778455 \end{pmatrix}$
- $\Sigma_2 = \frac{1}{N_2} \sum_{i=1}^{3} \gamma_{i2} (x_i - \mu_2) \cdot (x_i - \mu_2)^T = \begin{pmatrix} 0.998818 & -0.215305 \\ -0.215305 & 0.467476 \end{pmatrix}$
- $\pi_1 = \frac{N_1}{N} = \frac{N_1}{3} = \frac{1.251566}{3} = 0.417189$
- $\pi_2 = \frac{N_2}{N} = \frac{N_2}{3} = \frac{1.748434}{3} = 0.582811$

Aprendizagem 2022/23

Homework III– Group 104

2) Please see Appendix (1).

2.

a.

E - STEP

Realizar expectation step com os novos parâmetros:

- $\gamma_{11} = P(C_1 \mid x_1) = P(x_1 \mid C_1)\,P(C_1) = N(x_1 \mid \mu_1, \Sigma_1).\pi_1 = 0.081642$
- $\gamma_{21} = P(C_1 \mid x_2) = P(x_2 \mid C_1)\,P(C_1) = N(x_2 \mid \mu_1, \Sigma_1).\pi_1 = 0.003419$
- $\gamma_{31} = P(C_1 \mid x_3) = P(x_3 \mid C_1)\,P(C_1) = N(x_3 \mid \mu_1, \Sigma_1).\pi_1 = 0.032193$
- $\gamma_{12} = P(C_2 \mid x_1) = P(x_1 \mid C_2)\,P(C_2) = N(x_1 \mid \mu_2, \Sigma_2).\pi_2 = 0.007879$
- $\gamma_{22} = P(C_2 \mid x_2) = P(x_2 \mid C_2)\,P(C_2) = N(x_2 \mid \mu_2, \Sigma_2).\pi_2 = 0.083718$
- $\gamma_{32} = P(C_2 \mid x_3) = P(x_3 \mid C_2)\,P(C_2) = N(x_3 \mid \mu_2, \Sigma_2).\pi_2 = 0.061069$

Normalização:

- $\gamma_{11} = \dfrac{P(C_1 \mid x_1)}{P(C_1 \mid x_1) + P(C_2 \mid x_1)} = 0.911983$
- $\gamma_{12} = \dfrac{P(C_2 \mid x_1)}{P(C_2 \mid x_1) + P(C_1 \mid x_1)} = 0.088017$
- $\gamma_{21} = \dfrac{P(C_1 \mid x_2)}{P(C_1 \mid x_2) + P(C_2 \mid x_2)} = 0.039237$
- $\gamma_{22} = \dfrac{P(C_2 \mid x_2)}{P(C_2 \mid x_2) + P(C_1 \mid x_2)} = 0.960763$
- $\gamma_{31} = \dfrac{P(C_1 \mid x_3)}{P(C_1 \mid x_3) + P(C_2 \mid x_3)} = 0.345186$
- $\gamma_{32} = \dfrac{P(C_2 \mid x_3)}{P(C_2 \mid x_3) + P(C_1 \mid x_3)} = 0.654814$

Hard assignment:

$\gamma_{11} > \gamma_{12} \Rightarrow x_1 \in C_1$

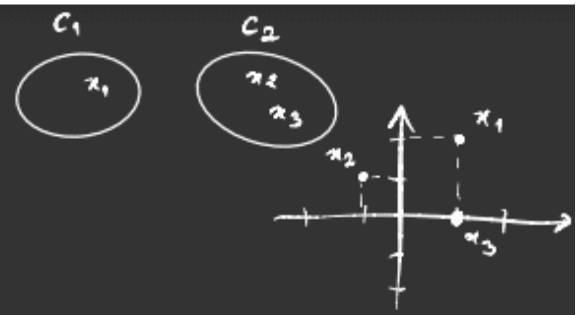
$\gamma_{21} < \gamma_{22} \Rightarrow x_2 \in C_2$

$\gamma_{31} < \gamma_{32} \Rightarrow x_3 \in C_2$

R.: É atribuído ao cluster 1 a observação x_1 e ao cluster 2 as observações x_2 e x_3.

Homework III– Group 104

b. C_2 é o maior cluster.



$$x_1 = \begin{pmatrix}1\\2\end{pmatrix},\ x_2 = \begin{pmatrix}-1\\1\end{pmatrix},\ x_3 = \begin{pmatrix}1\\0\end{pmatrix}$$

$$s(x_2) = 1 - \frac{a(x_2)}{b(x_2)} = 1 - \frac{d(x_2, x_3)}{d(x_2, x_1)} = 1 - \frac{\sqrt{(-1-1)^2+(1-0)^2}}{\sqrt{(-1-1)^2+(1-2)^2}} = 1 - \frac{\sqrt{5}}{\sqrt{5}} = 0$$

$$s(x_3) = \frac{b(x_3)}{a(x_3)} - 1 = \frac{d(x_3, x_1)}{d(x_3, x_2)} - 1 = \frac{\sqrt{(1-1)^2+(0-2)^2}}{\sqrt{(1-(-1))^2+(0-1)^2}} - 1 = \frac{\sqrt{4}}{\sqrt{5}} - 1 = -0.105573$$

$$s(C_2) = \frac{1}{m}\sum_{i=1}^{m} s(x_i) \mid x_i \in C_2$$, m é o nº observações do cluster 2

$$= \frac{1}{2}\left(s(x_2) + s(x_3)\right) = \frac{1}{2}\left(0 + (-0.105573)\right) = -0.0527864$$

R.: O valor da silhueta do cluster maior (C_2) é de -0.0527864.

Recordar:

$$\boxed{s(x_i) = \begin{cases} 1 - \dfrac{a(x_i)}{b(x_i)}, & a(x_i) \leq b(x_i) \\[2ex] \dfrac{b(x_i)}{a(x_i)} - 1, & a(x_i) > b(x_i) \end{cases}}$$

Aprendizagem 2022/23

Homework III– Group 104

II. Programming and critical analysis

1) The code used to answer this question is presented in Appendix (2).

Silhouette (seed=0): 0.11362027575179424

Purity (seed=0): 0.7671957671957672
Silhouette (seed=1): 0.11403554201377067
Purity (seed=1): 0.7632275132275133
Silhouette (seed=2): 0.11362027575179424
Purity (seed=2): 0.7671957671957672

2) The initialization of the centroids is done randomly (hence, the non-determinism). Therefore, different initializations produce different centroids which, consequently, may result in:

1) different clusters obtained at the end of the algorithm;

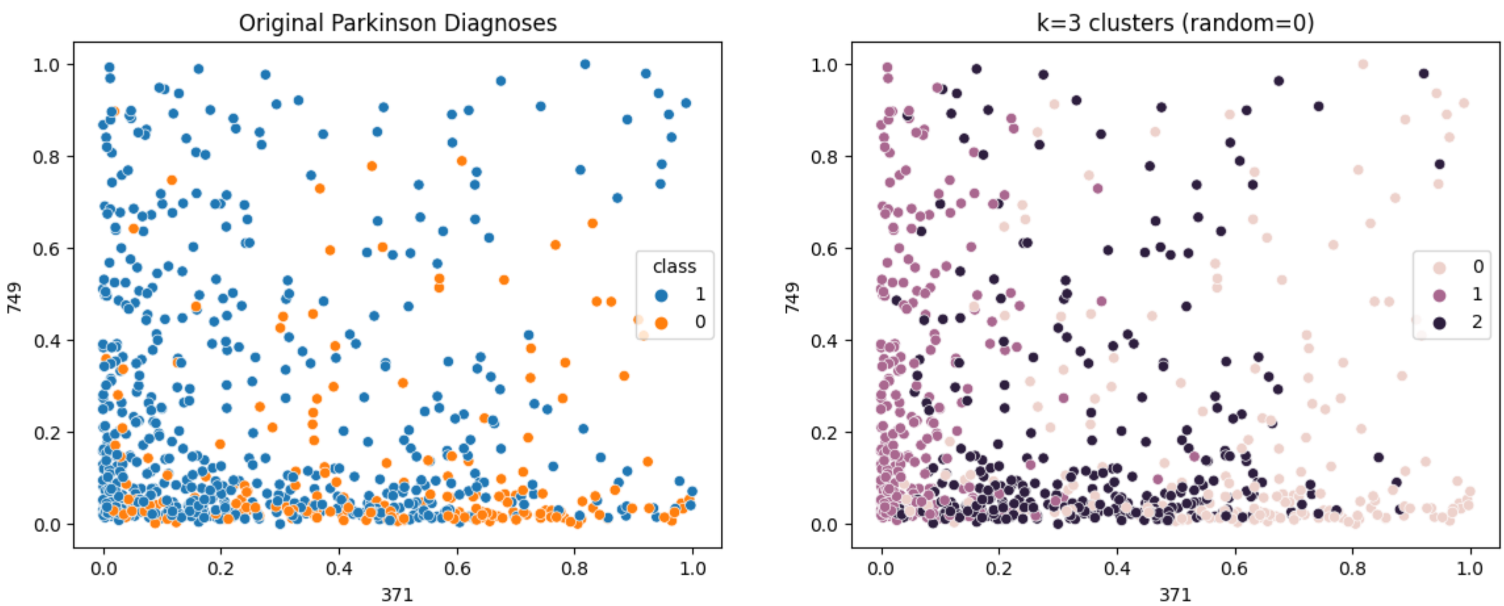
2) differences in clusters' convergence time.

Let's say we ran two different k-means (seed = 0 and seed = 1).

When assigning points to clusters, because centroids are different between k-means (seed = 0) and k-means (seed = 1), observation X may be assigned to cluster A in the former but to cluster C in the latter. Consequently, when updating centroids, because cluster A of k-means (seed = 0) have different points than cluster A of k-means (seed = 1), their underlying centroids, after update, will be different.

Note that non-determinism may also impact performance metrics like silhouette.

3)



4) The code used for this question is presented in Appendix (2).

Number of components: 31
Explained variance (%): 0.8006327717388223

Aprendizagem 2022/23
Homework III– Group 104

III. APPENDIX

(1)

```
import math
import numpy as np


# Auxiliary function
def get_probs(observations, u_lst, sigma_lst, pi_lst):

    probs_lst, probs_lst_norm = [], []

    for i in range(len(observations)):
        x_i = observations[i]
        obs_lst, obs_lst_norm = [], []

        for k in range(len(pi_lst)):
            u, sigma, pi = u_lst[k], sigma_lst[k], pi_lst[k]

            diff_x_u = x_i - u
            mult_1 = np.matmul(np.transpose(diff_x_u), np.linalg.inv(sigma))
            mult_2 = np.matmul(mult_1, diff_x_u)

            n = (1 / (pow(2 * math.pi, 2/2) * math.sqrt(np.linalg.det(sigma)))) *
pow(math.e, -1/2 * mult_2.item())
            obs_lst.append(n * pi)
            print(f'P(C_{k+1} | x_{i+1}) = {n * pi}')

        sum_obs_lst = sum(obs_lst)
        for j in range(len(obs_lst)):
            prob_norm = obs_lst[j] / sum_obs_lst
            obs_lst_norm.append(prob_norm)
            print(f'P(C_{j+1} | x_{i+1}) = {prob_norm} (normalizada)')

        probs_lst += obs_lst
        probs_lst_norm += obs_lst_norm

    return probs_lst_norm

obs = [np.matrix([[1], [2]]), np.matrix([[-1], [1]]), np.matrix([[1], [0]])]
u_lst = [np.matrix([[2], [2]]), np.matrix([[0], [0]])]
sigma_lst = [np.matrix([[2, 1], [1, 2]]), np.matrix([[2, 0], [0, 2]])]
pi_lst = [0.5, 0.5]
```

```python
probs = get_probs(obs, u_lst, sigma_lst, pi_lst)
print(probs)

n_lst, u_lst_updated, sigma_lst_updated, pi_lst_updated = [], [0, 0], [0, 0], [0, 0]

probs_c1 = [probs[i] for i in range(len(probs)) if i % 2 == 0]
probs_c2 = [probs[i] for i in range(len(probs)) if i % 2 != 0]
probs_lst = [probs_c1, probs_c2]

n1, n2 = sum(probs_c1), sum(probs_c2)
n_lst = [n1,  n2]
print(f'N_1 = {n1}\nN_2 = {n2}')

for i in range(len(u_lst_updated)):
    for j in range(len(obs)):
        u_lst_updated[i] += probs_lst[i][j] * obs[j]
    u_lst_updated[i] *= (1 / n_lst[i])
    print(f'u_{i+1} = {u_lst_updated[i] * (1 / n_lst[i])}')


for i in range (len(sigma_lst_updated)):
    for j in range(len(obs)):
        diff_x_u = obs[j] - u_lst_updated[i]

        calc1 = probs_lst[i][j] * (diff_x_u)
        calc2 = np.matmul(calc1, np.transpose(diff_x_u))

        sigma_lst_updated[i] += calc2
    sigma_lst_updated[i] *= (1 / n_lst[i])
    print(f'sigma_{i+1} = {sigma_lst_updated[i] * (1 / n_lst[i])}')


for i in range(len(pi_lst_updated)):
    n = len(obs)
    pi_lst_updated[i] = n_lst[i] / n
    print(f'pi_{i+1} = {n_lst[i] / n}')
probs_2 = get_probs(obs, u_lst_updated, sigma_lst_updated, pi_lst_updated)
print(probs_2)
```

Homework III– Group 104

(2)

```
from scipy.io.arff import loadarff
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn import datasets, metrics, cluster, mixture


# Read pd_speech.arff dataset
raw_data = loadarff('pd_speech.arff.txt')
df = pd.DataFrame(raw_data[0])
y_true = df['class'].str.decode('utf-8')
X = df.drop('class', axis=1)


# Normalize data
X_norm = MinMaxScaler().fit_transform(X)
df_norm = pd.DataFrame(X_norm)
df_norm.head()


def purity_score(y_true, y_pred):
    # compute contingency/confusion matrix
    confusion_matrix = metrics.cluster.contingency_matrix(y_true, y_pred)
    return np.sum(np.amax(confusion_matrix, axis=0)) / np.sum(confusion_matrix)


no_seeds = 3
y_pred_0 = []

for i in range(no_seeds):
    kmeans_algo = cluster.KMeans(n_clusters=3, random_state=i)
    kmeans_model = kmeans_algo.fit(df_norm)

    y_pred = kmeans_model.labels_

    if i == 0:
        y_pred_0 = y_pred
```

```
    # assess silhouette and purity
    print(f"Silhouette (seed={i}):",metrics.silhouette_score(df_norm, y_pred,
metric='euclidean'))
    print(f"Purity (seed={i}):",purity_score(y_true, y_pred))

# Feature Selection
variances = df_norm.var()
two_largest = variances.nlargest(2)
two_largest_idxs = [variances[variances == el].index[0] for el in two_largest]

x_f, y_f = df_norm[two_largest_idxs]

# Scatter Plots
# i) original Parkinson diagnoses
plt.figure(figsize=(14, 5))
plot1 = plt.subplot(121)
plot1.title.set_text('Original Parkinson Diagnoses')
sns.scatterplot(data=df_norm[two_largest_idxs], x=x_f, y=y_f, hue=y_true)

# ii) previously learned k=3 clusters (random = 0)
plot2 = plt.subplot(122)
plot2.title.set_text('k=3 clusters (random=0)')
sns.scatterplot(data=df_norm[two_largest_idxs], x=x_f, y=y_f, hue=y_pred_0)


# Code for question 4 (PCA)
n_comp = 1
sum_exp_var_ratios = 0

while sum_exp_var_ratios <= 0.8:
    pca = PCA(n_components=n_comp)
    pca.fit(df_norm)
    print(f"Number of components: {n_comp}")
    print("Components (eigenvectors):\n",pca.components_)
    print("Explained variance (eigenvalues) =",pca.explained_variance_)
    print("Explained variance (ratio) =",pca.explained_variance_ratio_)
    sum_exp_var_ratios = sum(pca.explained_variance_ratio_)
    n_comp += 1
    print(sum_exp_var_ratios)

print(f'Number of components: {n_comp-1}\nExplained variance:
{sum_exp_var_ratios}')
```

END