

Ficha laboratorial - Clustering

Ex1. Corre o código abaixo

```
1 import numpy as np
2 from matplotlib import pyplot as plt
3 from sklearn import datasets
4
5 from sklearn.cluster import KMeans
6 from sklearn.cluster import AgglomerativeClustering as aggClus
7 from scipy.cluster.hierarchy import dendrogram
8
9
10 Iris = datasets.load_iris()
11 X=Iris.data.T
12 print(X.shape)
13
14 # Num problema de aprend. não supervisionada não teríamos acesso a esta informação
15 trueClass=Iris.target
```

Na célula seguinte coloca a instrução:

```
kmeans = KMeans(n_clusters=3, init='k-means++', n_init=5, max_iter=500, tol=0.0001, verbose=0, random_state=42)
```

e corre o código seguinte:

```
1 kmeans = kmeans.fit(X.T)
2
3 C= kmeans.cluster_centers_
4 y=kmeans.labels_
```

Imprime o conteúdo das variáveis C e y e analisa o seu conteúdo.

Ex.2 Corre os dois códigos apresentados abaixo. Analisa os gráficos obtidos.

```
1 plt.figure(figsize=(10,5))
2 plt.title('Clusters encontrados')
3 plt.plot(X[0,y==0],X[1,y==0], 'g.')
4 plt.plot(X[0,y==1],X[1,y==1], 'b.')
5 plt.plot(X[0,y==2],X[1,y==2], 'r.')
6 plt.plot(C[:,0],C[:,1], 'k+')
7 plt.axis([2.5, 8, 1.3, 5.2])
8 plt.grid()
9 plt.show()
```

```
1 y1=trueClass
2 plt.figure(figsize=(10,5))
3 plt.title('Classes originais')
4 plt.plot(X[0,y1==0],X[1,y1==0], 'b.')
5 plt.plot(X[0,y1==1],X[1,y1==1], 'g.')
6 plt.plot(X[0,y1==2],X[1,y1==2], 'r.')
7 plt.axis([2.5, 8, 1.3, 5.2])
8 plt.grid()
9 plt.show()
10
```

Ex. 3 Corre o código abaixo. A primeira rotina pode ser obtida em :

https://scikit-learn.org/stable/auto_examples/cluster/plot_agglomerative_dendrogram.html

Para a rotina correr corretamente é necessário importar:

```
from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering as aggClus
from scipy.cluster.hierarchy import dendrogram
```

que já foi importado no exercício 1.

```
1 def plot_dendrogram(model, **kwargs):
2     # Create linkage matrix and then plot the dendrogram
3
4     # create the counts of samples under each node
5     counts = np.zeros(model.children_.shape[0])
6     n_samples = len(model.labels_)
7     for i, merge in enumerate(model.children_):
8         current_count = 0
9         for child_idx in merge:
10             if child_idx < n_samples:
11                 current_count += 1 # leaf node
12             else:
13                 current_count += counts[child_idx - n_samples]
14         counts[i] = current_count
15
16     linkage_matrix = np.column_stack(
17         [model.children_, model.distances_, counts]
18     ).astype(float)
19
20     # Plot the corresponding dendrogram
21     dendrogram(linkage_matrix, **kwargs)
```

```
1 # Colocando distance_threshold=0 garante que obtemos a árvore completa.
2 model = aggClus(distance_threshold = 0, n_clusters = None, linkage = 'ward')
3 model = model.fit(X.T)
4 plt.title("Iris - Hierarchical Clustering Dendrogram")
5 # Visualizar todos os 150 níveis do dendrograma
6 plot_dendrogram(model, labels = None, truncate_mode = "level", p=150)
```

Gráfico que deve ser obtido com o código:



Ex. 4 O código abaixo aplica as experiências anteriores a um novo conjunto de dados.

Executa o código replicando o código já introduzido e adapta observando os resultados.

```

1 import numpy as np
2 from matplotlib import pyplot as plt
3 import pickle
4
5 from sklearn.cluster import KMeans
6 from sklearn.cluster import AgglomerativeClustering as aggClus
7 from scipy.cluster.hierarchy import dendrogram

1 file = open("/mix9gaus2d.p", "rb")

1 dados = pickle.load(file)

1 X=dados

1 kmeans = KMeans(n_clusters=9, init='k-means++', n_init =5, max_iter=500, tol = 0.0001, verbose=0, random_state=42)

1 kmeans = kmeans.fit(X.T)

1 C=kmeans.cluster_centers_

1 C

1 y=kmeans.labels_

1 plt.figure(figsize=(10,5))
2 plt.title('Clusters encontrados')
3 plt.plot(X[0,y==0],X[1,y==0], 'g.')
4 plt.plot(X[0,y==1],X[1,y==1], 'b.')
5 plt.plot(X[0,y==2],X[1,y==2], 'r.')
6 plt.plot(X[0,y==3],X[1,y==3], 'c.')
7 plt.plot(X[0,y==4],X[1,y==4], 'm.')
8 plt.plot(X[0,y==5],X[1,y==5], 'y.')
9 plt.plot(X[0,y==6],X[1,y==6], 'k.')
10 plt.plot(X[0,y==7],X[1,y==7], color = 'tab:orange', marker = '*', linestyle = '')
11 plt.plot(X[0,y==8],X[1,y==8], color = 'tab:gray', marker = '.', linestyle = '')
12
13 plt.plot(C[:,0],C[:,1], 'k*')
14 plt.grid()
15 plt.show()

1 #setting distance_threshold=0 ensures we compute the full tree.
2 model = aggClus(distance_threshold = 0, n_clusters = None, linkage = 'ward')
3 model = model.fit(X.T)
4 plt.title("Hierarchical Clustering Dendrogram")
5 # plot the top 10 levels of the dendrogram
6 plot_dendrogram(model, labels = None, truncate_mode = "level", p=10)

```