Titulo: Exercício 4 - k-means, métricas internas e métricas externas
Autor: Juan Sebastián Beleño Díaz
Data: 25 de Outubro de 2016

Introdução

Neste trabalho é implementado um k-means para agrupar um conjunto de dados em *clusters* de diferentes tamanhos. Além disso, são utilizadas métricas internas e externas para evaluar os *clusters*.

Dados

Os arquivos usados neste trabalho são cluster-data.csv e cluster-data-class.csv que pertencem ao conjunto de dados Activity Recognition from Single Chest-Mounted Accelerometer Data Set. O conjunto de dados foi coletado de pessoas relizando diferentes padrões de movimentos. O arquivo *cluster-data.csv* contem as cordenadas (x,y,z) das pessoas realizando os padrões de movimentos. O arquivo *cluster-data-class.csv* tem o tipo de padrão de movimento feito pelas pessoas.

Preparação dos dados

Antes de começar trablahar com os dados é preciso incluir as dependecias do projeto:

```
%matplotlib inline

# Loading the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.cluster import KMeans
from sklearn import metrics
```

Existem muitas maneiras de abrir os arquivos e obter os dados, mas neste caso foi usado *pandas* para obter os *dataframes* diretamente desde a URL.

```
# Defining the URIs with raw data
url_parameters =
'http://www.ic.unicamp.br/~wainer/cursos/2s2016/ml/cluster-data.csv'
url_classes =
'http://www.ic.unicamp.br/~wainer/cursos/2s2016/ml/cluster-data-class.csv'

# Reading the files with the raw data
df_parameters = pd.read_csv(url_parameters, header = 0, delimiter = ",")
df_classes = pd.read_csv(url_classes, header = 0, delimiter = ",")
```

Embaixo é apresentado um conjunto de variáveis necessárias para executar o k-means e aplicar as metricas(internas e externas).

```
# k values to iterate until get a good k
k_parameters = range(2, 11)

# Number of restarts
n_restarts = 5

best_internal_score = 0 # This is for Calinski Harabaz score
best_internal_k = 2

best_external_score = -1 # This is the minimum value for adjusted rand score
best_external_k = 2

# Matrix for external and internal metrics
# First line is for k values
# Second line is for internal metrics
# Third line is for external metrics
matrix_plot = [[0]*9 for i in range(3)]
```

Busca dos Clusters usando k-means

Neste código é usado k-means para obter clusters de diferentes tamanho usando Calinski Harabaz Index como métrica interna e Adjusted Rand Index como métrica externa. As métricas são armazenadas numa matriz que será usada para plotar um gráfico do comportamento das métricas segundo o valor de k.

```
for k in k_parameters:

    # Declaring the model with k clusters
    k_means_model = KMeans(k, n_init = n_restarts)
    k_means_model.fit(df_parameters)


    # Internal metric
    predicted_labels = k_means_model.labels_
    internal_score = metrics.calinski_harabaz_score(df_parameters, predicted_labels)

    #print('Internal score: ', k,' - ',internal_score)
    if internal_score > best_internal_score:
        best_internal_score = internal_score
        best_internal_k = k


    # External metric
    true_labels = np.ravel(df_classes)
    external_score = metrics.adjusted_rand_score(true_labels, predicted_labels)
    #print('External score: ',k,' - ',external_score)
    if external_score > best_external_score:
        best_external_score = external_score
        best_external_k = k


    # Filling the matrix with metrics
    matrix_plot[0][k-2] = k
```

```
    matrix_plot[1][k-2] = internal_score
    matrix_plot[2][k-2] = external_score
```

Resultados

```
# Showing the k values for different metrics
print('K for internal metric (Calinski Harabaz Index): ', best_internal_k)
print('K for external metric (Adjusted Rand Index): ', best_external_k)
```

```
K for internal metric (Calinski Harabaz Index):  3
K for external metric (Adjusted Rand Index):  4
```

Gráficas das métricas internas e externas segundo o K

```
# Plotting charts with internal and external metrics
fig_width = 16
fig_height = 4
fig_dpi = 100
plt.figure(figsize=(fig_width, fig_height), dpi=fig_dpi)
plt.figure(1)
plt.subplot(121) # Grid 1 x 2. Figure #1
plt.plot(matrix_plot[0], matrix_plot[1], '#ff5722', marker='o', linestyle='-')
plt.xlabel('Number of clusters')
plt.ylabel('Calinski Harabaz Index')

plt.subplot(122) # Grid 1 x 2. Figure #2
plt.plot(matrix_plot[0], matrix_plot[2], '#009688', marker='o', linestyle='-')
plt.xlabel('Number of clusters')
plt.ylabel('Adjusted Rand Index')
plt.show()
```