

Exercício 4 - MO444 - Aprendizado de máquina e reconhecimento de padrões

Renato Lopes Moura - 163050

Importando os módulos que serão utilizados no exercício.

```
In [1]: import numpy as np
import pandas as pd
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

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

Preparação dos dados para clusterização.

```
In [2]: #Carregando o conjunto de dados do csv usando o pandas
data = pd.read_csv('cluster-data.csv')

#Conversao para arrays do numpy
array = data.values

#Carregando o conjunto de classes dos dados do csv usando o pandas
data_class = pd.read_csv('cluster-data-class.csv')

#Conversao para arrays do numpy
array_class = data_class.values
```

Loop de clusterização pela métrica interna (Silhouette)

```
In [3]: #Inicializacao das variaveis que armazenarao o melhor k (numero de clusters)
# e o melhor score interno
internal_k = 0
internal_score = 0

#Loop de escolha do melhor k segundo a metrica interna (Silhouette)
for k in range(2,11):

    #Instanciacao do KMeans e avaliacao do score da clusterizacao
    kmeans = KMeans(n_clusters=k, n_init=5, max_iter=1000, init='random').fit(array)
    labels = kmeans.labels_
    current_score = metrics.silhouette_score(array, labels, metric='euclidean')

    #Escolha do melhor k na avaliacao interna
    if current_score > internal_score:
        internal_score = current_score
        internal_k = k

print "k escolhido segundo a metrica interna: "+str(internal_k)
```

```
k escolhido segundo a metrica interna: 2
```

Loop de clusterização pela métrica externa (Adjusted Rand Score)

```
In [4]: #Inicializacao das variaveis que armazenarao o melhor k (numero de clusters)
# e o melhor score externo
external_k = 0
external_score = 0

#Loop de escolha do melhor k segundo a metrica interna (Silhouette)
for k in range(2,11):

    #Instanciacao do KMeans e avaliacao do score da clusterizacao
    kmeans = KMeans(n_clusters=k, n_init=5, max_iter=1000, init='random').fit(X)
    labels = kmeans.labels_
    current_score = metrics.adjusted_rand_score(array_class[:,0], labels)

    #Escolha do melhor k na avaliacao interna
    if current_score > external_score:
        external_score = current_score
        external_k = k

print "k escolhido segundo a metrica externa: "+str(external_k)
```

```
k escolhido segundo a metrica externa: 4
```