Aprendizado de Máquina - MO444

Exercício 4

Aluno: Paulo Ricardo Finardi. RA: 144809

0 Preliminares

Foi utilizado `Python 3` na resolução dos exercícios.

0.1 Métrica Interna

Para a métrica interna utilizamos a `Silhouete` da biblioteca `sklearn`.
Sobre a `Silhouete`: A `Silhouete` fornece uma representação gráfica de quão bem cada objeto encontra-se dentro de seu cluster. Os valores são uma medida de quão semelhante um objeto é para seu próprio cluster (coesão) em comparação a outros clusters (separação). A `Silhouete` varia de -1 a 1, onde um alto valor indica que o objeto é bem adaptado ao seu próprio cluster.

0.2 Métrica Externa

Para a métrica externa utilizamos a função `Adjusted Mutual Information` também da biblioteca `sklearn`.
Sobre a `Adjusted Mutual Information (AMI)`: A `Mutual Information` é uma medida de similaridade entre duas *labels* do mesmo dado. Como o exercício forneceu as corretas *labels*, as comparamos com as *labels* preditas pelo algoritmo `k-means`. A interpretação do valor da AMI é dado da seguinte forma: Se não existir nenhuma diferença entre os dois conjuntos de `labels`, o AMI retornará pontuação igual a 1.

1 Código

1.1 Leitura dos dados

Enunciado: *Use os dados do arquivo cluster-data.csv (os dados sao uma media de 30 medidas por vez da pessoa 1 do dataset Activity Recognition from Single Chest-Mounted Accelerometer Data Set.*

Resposta:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples,
silhouette_score, adjusted_mutual_info_score

# data armazena o arquivo cluster-data.csv
data = pd.read_csv('cluster-data.csv', header=0)

# X é o data em formato numpy-array
X = data.as_matrix(columns=None)

# y são as labels fornecidas pelo arquivo cluster-data-class.csv
labels = pd.read_csv('cluster-data-class.csv', header=0)
y = labels.as_matrix(columns=None)
y = np.ravel(y)
```

1.2 K-means

Enunciado: *Rode o kmeans nos dados, com numero de restarts = 5. Use alguma metrica interna (algum Dunn, Silhouette, Calinski-Harabaz index) - apenas uma -para escolher o k entre 2 e 10.*

Resposta:

```
# variável mutual será utilizada no plot do AMI
mutual = []

# variando k entre 2 e 10
for k in range(2,11):
    # km é o k-means
    km = KMeans(n_clusters=k, n_init=5)
    km_labels = km.fit_predict(X)

```

```
    # métrica interna Silhouette
    silhouette_avg = silhouette_score(X, km_labels)
    print('Para #%d clusters:\nMétrica Interna: Silhouette = %.6f' % \
          (k, silhouette_avg))

    # métrica externa Mutual Information
    metric = adjusted_mutual_info_score(y, km_labels)
    mutual.append(metric)
    print('Métrica Externa: Mutal information = %.6f' % metric)

    # plot para a métrica interna Silhouette
    fig, ax1 = plt.subplots(1)
    fig.set_size_inches(8,7)
    ax1.set_xlim([-0.1, 1])
    ax1.set_ylim([0, len(X)])

    # cômputo do Silhouette para cada exemplo
    sample_silhouette_values = silhouette_samples(X, km_labels)
    y_lower = 10
    for i in range(k):
        ith_cluster_silhouette_values = \
            sample_silhouette_values[km_labels == i]
        ith_cluster_silhouette_values.sort()

        size_cluster_i = ith_cluster_silhouette_values.shape[0]
        y_upper = y_lower + size_cluster_i

        color = cm.spectral(float(i) / k)
        ax1.fill_betweenx(np.arange(y_lower, y_upper),
                          0, ith_cluster_silhouette_values,
                          facecolor=color, edgecolor=color, alpha=0.7)

        # label do Silhouette com o número de clusters
        ax1.text(-0.05, y_lower + 0.5 * size_cluster_i, str(i))
        y_lower = y_upper + 10  # 10 for the 0 samples

```

```
    ax1.set_title("Silhouette dos Clusters",  fontsize=14)
    ax1.set_xlabel("Pontuação do Silhouette ",  fontsize=14)
    ax1.set_ylabel("Rótulo do Cluster",  fontsize=14)
    ax1.axvline(x=silhouette_avg, color="red", linestyle="--")
    ax1.set_yticks([])
    ax1.set_xticks([-0.1, 0, 0.2, 0.4, 0.6, 0.8, 1])
```

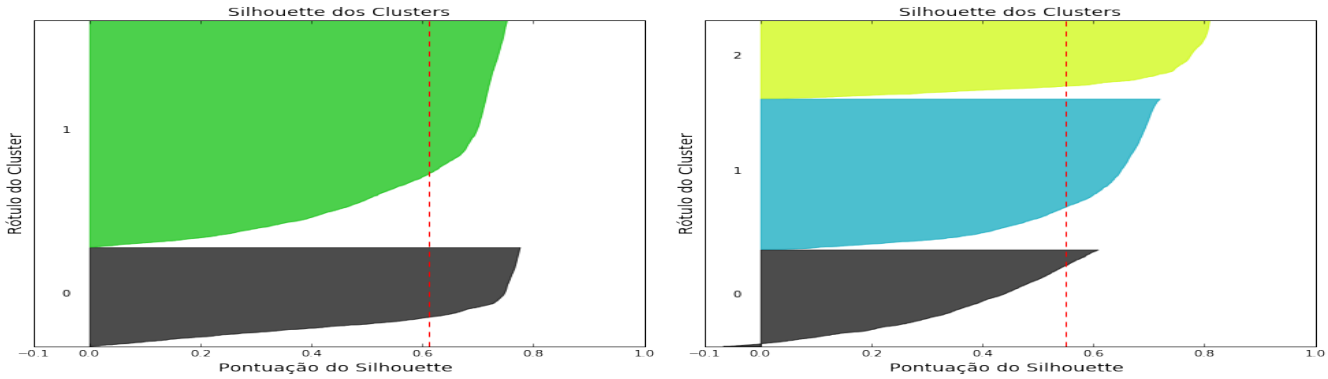
Os *prints* desse bloco são:

```
Para #2 clusters:
Métrica Interna: Silhouette = 0.611942
Métrica Externa: Mutal information = 0.252702
Para #3 clusters:
Métrica Interna: Silhouette = 0.549662
Métrica Externa: Mutal information = 0.436861
Para #4 clusters:
Métrica Interna: Silhouette = 0.497895
Métrica Externa: Mutal information = 0.486548
Para #5 clusters:
Métrica Interna: Silhouette = 0.428067
Métrica Externa: Mutal information = 0.415292
Para #6 clusters:
Métrica Interna: Silhouette = 0.428477
Métrica Externa: Mutal information = 0.417380
Para #7 clusters:
Métrica Interna: Silhouette = 0.430129
Métrica Externa: Mutal information = 0.397152
Para #8 clusters:
Métrica Interna: Silhouette = 0.432447
Métrica Externa: Mutal information = 0.372535
Para #9 clusters:
Métrica Interna: Silhouette = 0.416470
Métrica Externa: Mutal information = 0.351052
Para #10 clusters:
Métrica Interna: Silhouette = 0.358527
Métrica Externa: Mutal information = 0.326851
```

1.3 Plots

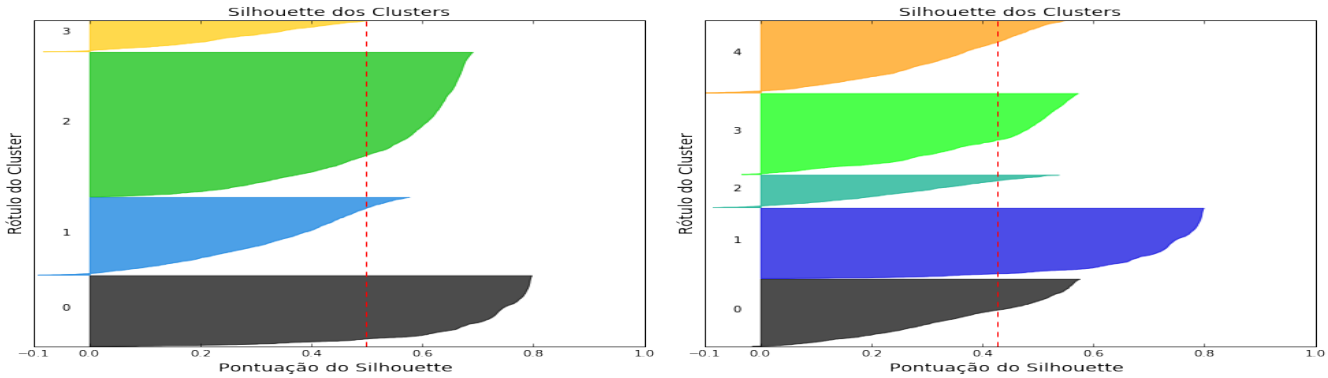
Enunciado: *O arquivo cluster-data-class.csv contem a classe correta de cada ponto. Use alguma medida externa (Normalized/adjusted Rand, Mutual information, variation of information) para decidir no k. Plote os graficos correspondentes das 2 metricas (interna e externa) para os varios valores de k (extra).*

Resposta: Os plots a seguir se referem ao bloco de código anterior.



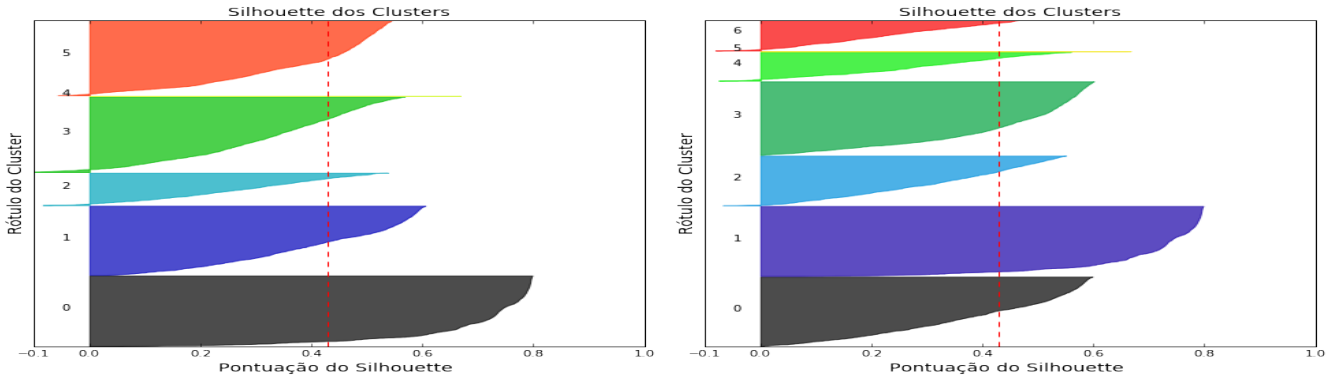
(a) (b)

Figura 1: Figura (a): $k = 2$, Pontuação Sillhouette = 0.611942. Pontuação Mutual Information = 0.252702. (b): $k = 3$, Pontuação Sillhouette = 0.549662. Pontuação Mutual Information = 0.436861.



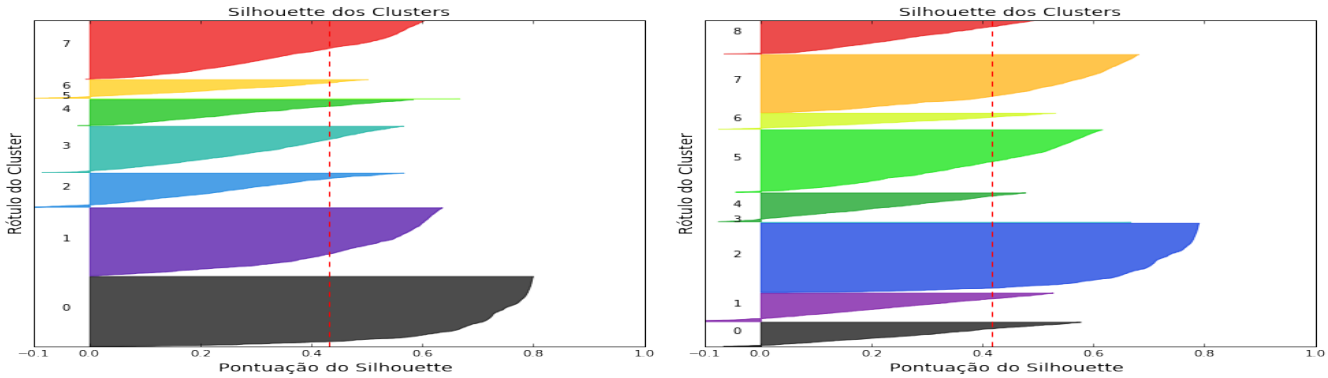
(a) (b)

Figura 2: Figura (a): $k = 4$, Pontuação Sillhouette = 0.497895. Pontuação Mutual Information = 0.486548. (b): $k = 5$, Pontuação Sillhouette = 0.428067. Pontuação Mutual Information = 0.415292.



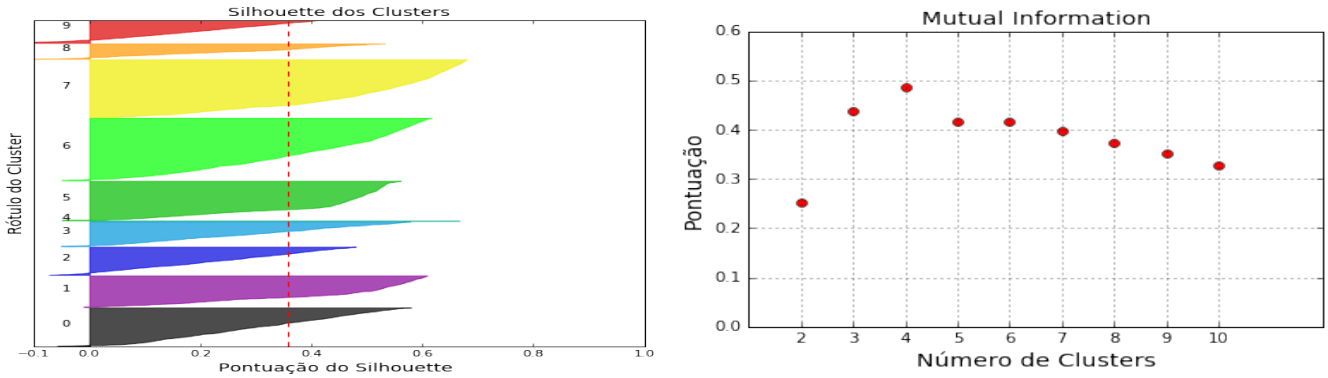
(a) (b)

Figura 3: Figura (a): $k = 6$, Pontuação Sillhouette = 0.428477. Pontuação Mutual Information = 0.417380. (b): $k = 7$, Pontuação Sillhouette = 0.430129. Pontuação Mutual Information = 0.397152.



(a) (b)

Figura 4: Figura (a): $k = 8$, Pontuação Sillhouette = 0.432447. Pontuação Mutual Information = 0.372535. (b): $k = 9$, Pontuação Sillhouette = 0.416470. Pontuação Mutual Information = 0.351052.



(a) (b)

Figura 5: Figura (a): $k = 10$, Pontuação Sillhouette = 0.358527. Pontuação Mutual Information = 0.326851. (b): Gráfico Mutual Information para k entre (2,10).

1.4 Conclusão

Decidimos escolher o valor de k para a métrica que possui maior valor de `Mutual Information`. Dessa forma, escolhemos $k = 4$ e temos: `Mutual Information` = 0.486548 e `Silhouette` = 0.497895, Figura 2−(a). Repetimos o `K-means` com o valor $k = 4$.

```
# k-means com k=4
km_new = KMeans(n_clusters=4, n_init=5)
km_labels_new = km_new.fit_predict(X)

# plot
fig, ax = plt.subplots(1)
fig.set_size_inches(18,7)
colors = cm.spectral(km_labels_new.astype(float)/4)
ax.scatter(X[:,0], X[:,1], marker='.', s=400, lw=0, alpha=0.7, c=colors)
ax.set_title('Visualização dos dados Clusterizados', fontsize=14)
plt.xticks([])
plt.yticks([])
plt.show()
```

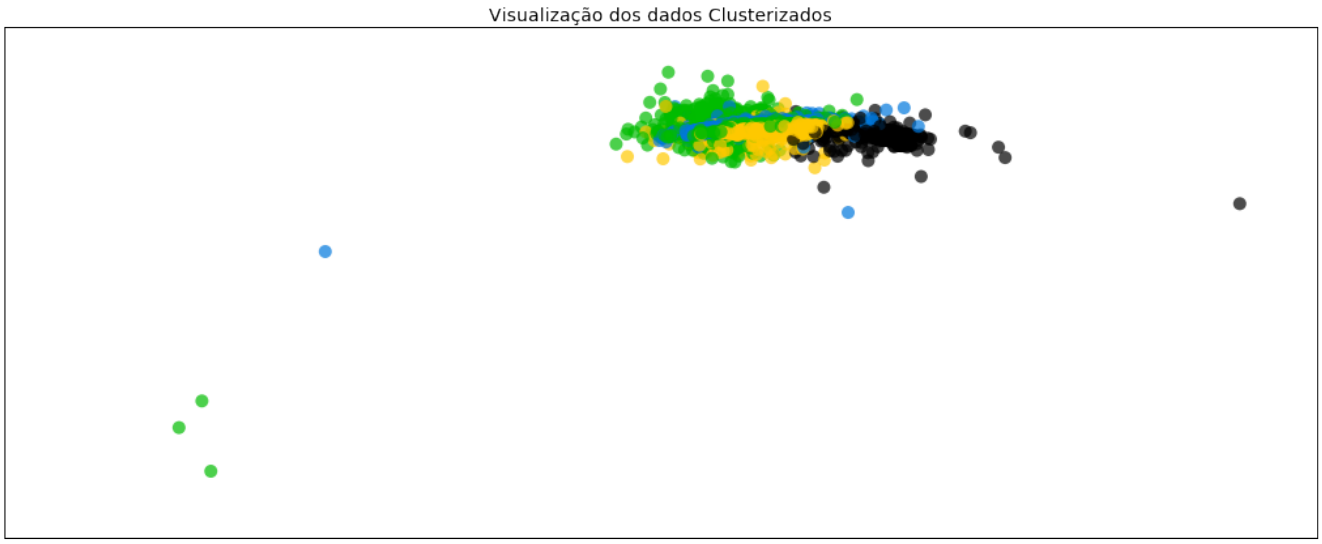


Figura 6: Visualização dos dados com algoritmo K-means, com $k = 4$.