

Exercicio4

October 25, 2016

```
In [1]: from sklearn import cluster, metrics
import numpy as np
import operator
x = np.loadtxt \
    ('/home/abissoto/Documents/Exercicio4_ML/cluster-data.csv', \
     delimiter=',', skiprows=1)
labels_true = np.loadtxt \
    ('/home/abissoto/Documents/Exercicio4_ML/cluster-data-class.csv', \
     delimiter=',', skiprows=1)
```

```
In [2]: k_inner_metric = {}
k_outter_metric = {}
for k in range(2, 11):
    kmeans = cluster.KMeans(n_clusters=k, n_init=5)
    kmeans.fit(x)
    labels_pred = kmeans.labels_
    k_inner_metric[k] = metrics.silhouette_score \
        (x, labels_pred, metric='euclidean')
    k_outter_metric[k] = metrics.adjusted_mutual_info_score \
        (labels_true, labels_pred)
sorted_k_inner = sorted(k_inner_metric.items(), \
                        key=operator.itemgetter(1))
sorted_k_outter = sorted(k_outter_metric.items(), \
                        key=operator.itemgetter(1))
print(sorted_k_inner[-1])
print(sorted_k_outter[-1])
```

```
(2, 0.61194167066620009)
```

```
(4, 0.48654775014674112)
```

```
In [3]: import plotly.graph_objs as go
import plotly.plotly as py

# Create traces
trace0 = go.Scatter(
    x = list(k_inner_metric.keys()),
    y = list(k_inner_metric.values()),
```

```

        mode = 'lines',
        name = 'Inner Metrics'
    )
    trace1 = go.Scatter(
        x = list(k_outter_metric.keys()),
        y = list(k_outter_metric.values()),
        mode = 'lines',
        name = 'Outter Metrics'
    )
    data = [trace0, trace1]
    layout = go.Layout(
        title='Metrics scores x K values',
        xaxis=dict(
            title='K value',
        ),
        yaxis=dict(
            title='Score',
        )
    )
    figure = go.Figure(data=data, layout=layout)
    #iplot(data, filename='kmeans-lines')
    py.image.show(figure)

```

