Agrupamento Espectral e Hierárquico

Fabrício Olivetti de França

Universidade Federal do ABC

Tópicos

1. Agrupamento Espectral

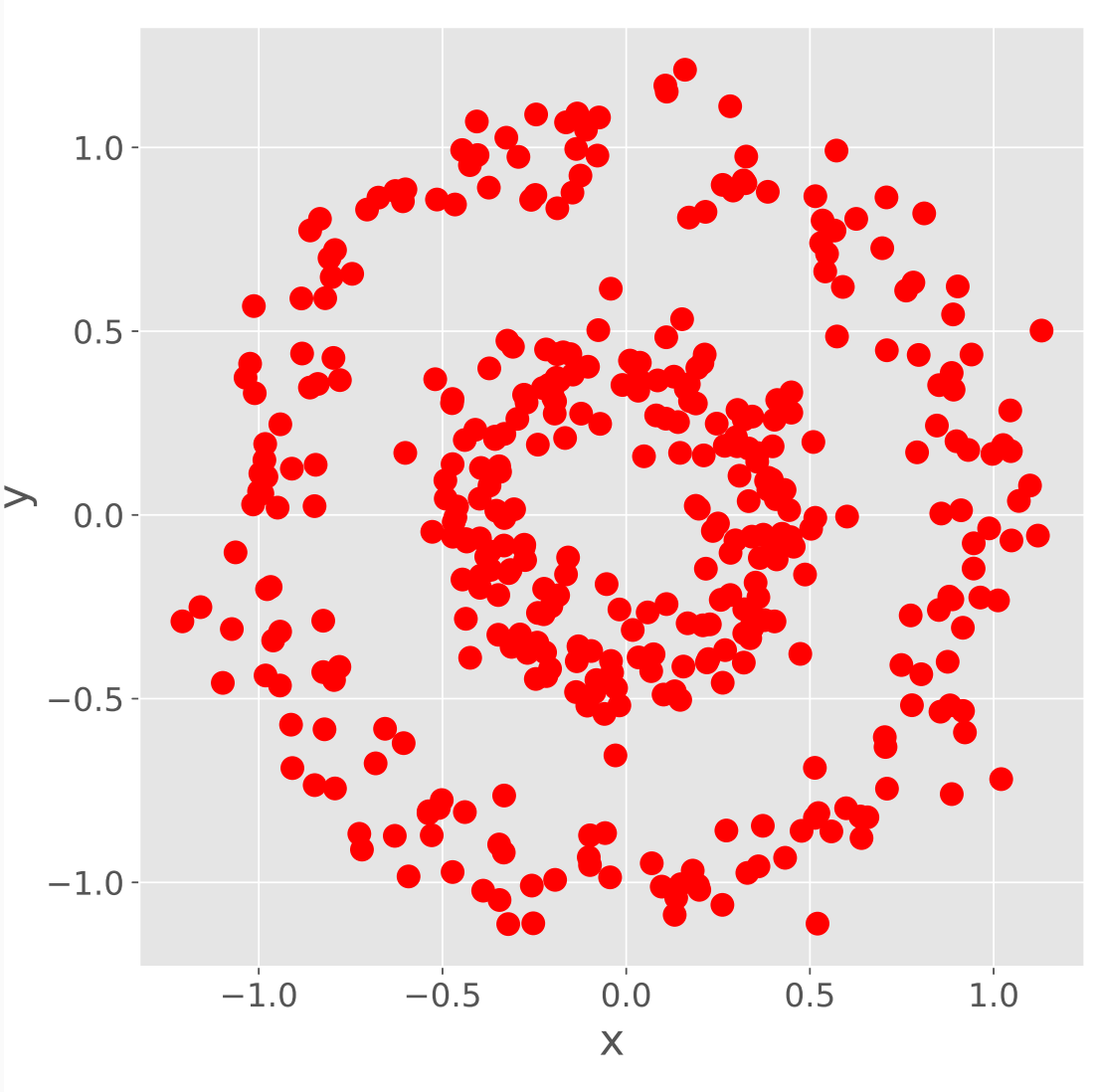
2. Agrupamento Hierárquico

UFABC

Agrupamento Espectral

Agrupamento Espectral

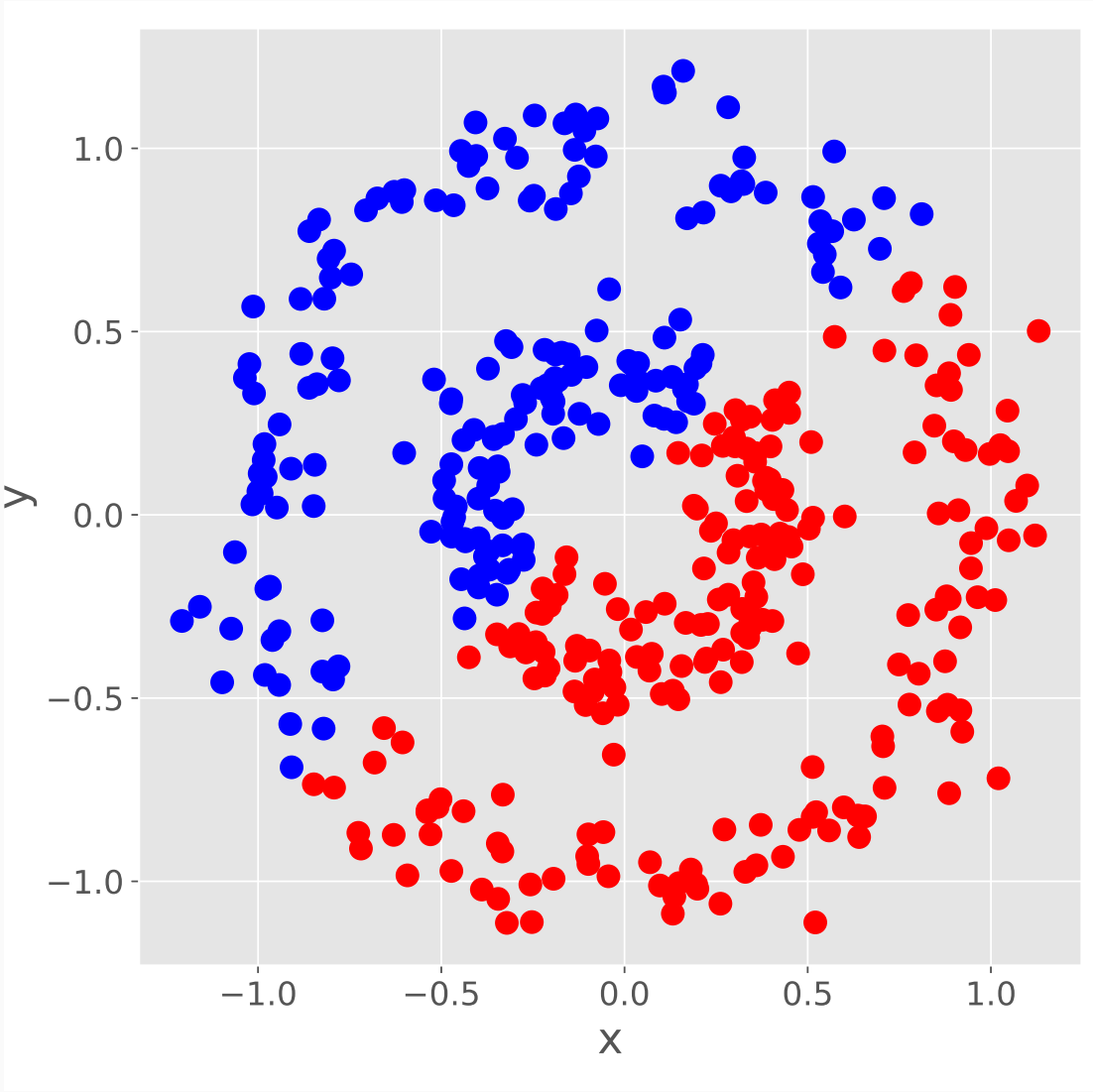
Nem sempre nossos dados apresentam um agrupamento óbvio mensurado por uma medida de similaridade:



UFABC

Agrupamento Espectral

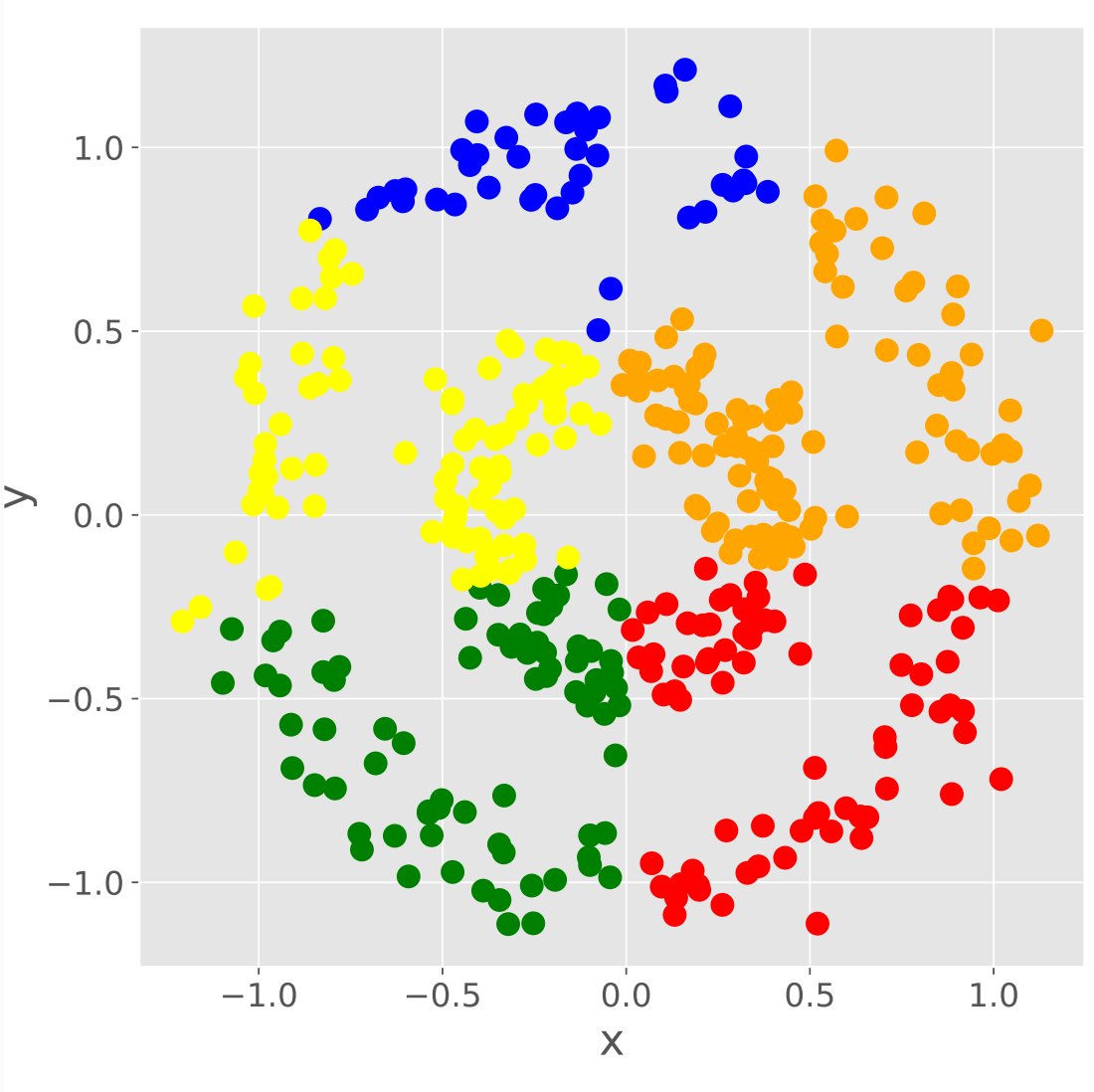
O uso do algoritmo k-Means é insuficiente para encontrar os dois grupos existentes:



UFABC

Agrupamento Espectral

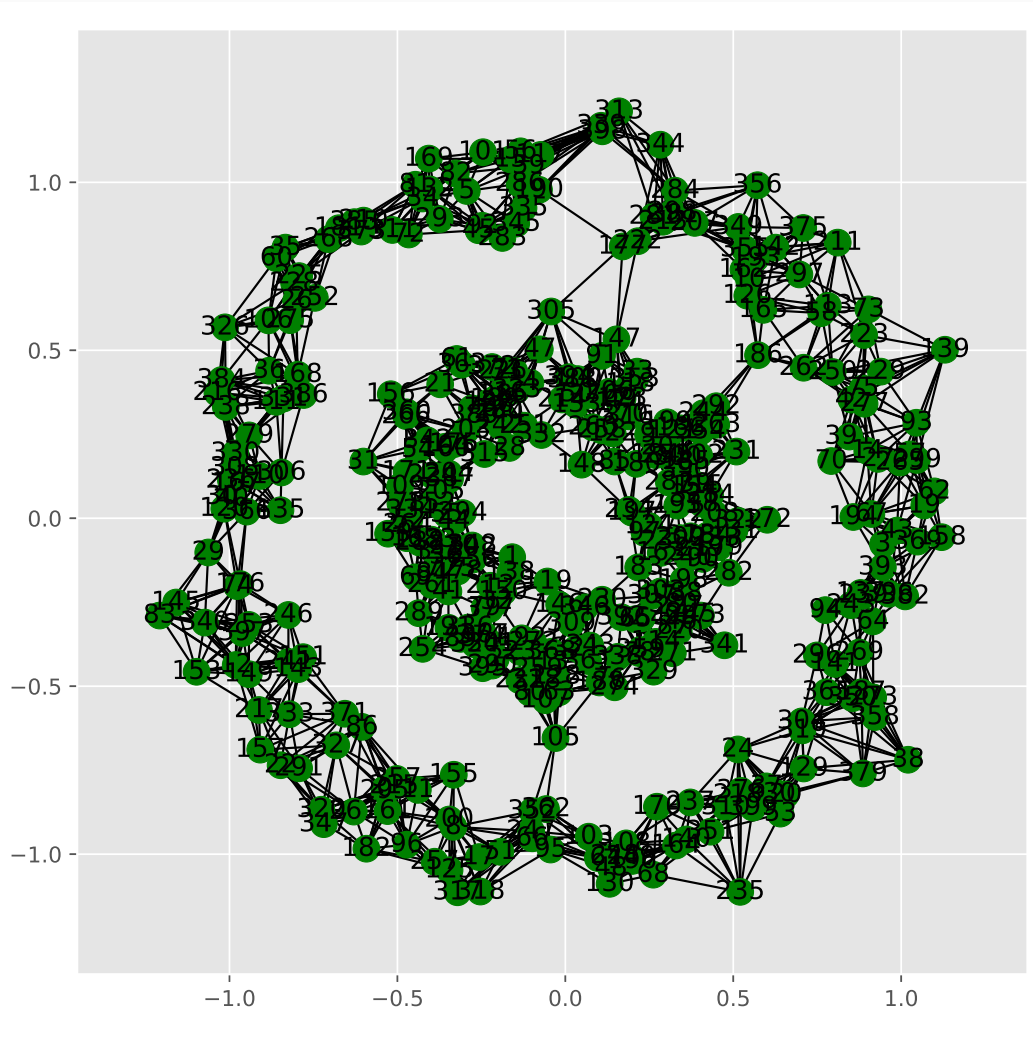
Mesmo utilizando um número maior de clusters o algoritmo apresenta dificuldades para entender a estrutura da base de dados.



UFABC

Agrupamento Espectral

Uma forma alternativa de representar esses objetos é imaginar que eles formam um grafo em que os k pontos mais próximos de um certo ponto, forma uma aresta com este.



UFABC

Agrupamento Espectral

Dessa forma passamos a representar nossos dados através da matriz Laplaciana, que é dada por:

$$L = G - A,$$

com A sendo a matriz de adjacência e G uma matriz diagonal com os elementos da diagonal igual ao grau do nó correspondente.

Agrupamento Espectral

Essa matriz Laplaciana tem algumas propriedades interessantes:

- O número de autovalores iguais a 0 é igual ao número de componentes conexos.
- Os autovetores correspondentes aos autovalores iguais a 0 representam um grupo, sendo os nós pertencentes a esse grupo com valores positivos e todo o restante igual a 0.
- Os autovetores consequentes representam diversas formações de possíveis agrupamentos.

Agrupamento Espectral

Com isso podemos calcular a Laplaciana de uma base de dados e utilizar os k primeiros autovetores dessa matriz (com autovalores diferentes de 0) e gerar (similar ao PCA) uma matriz $n \times k$ contendo a informação dos grupos.

UFABC

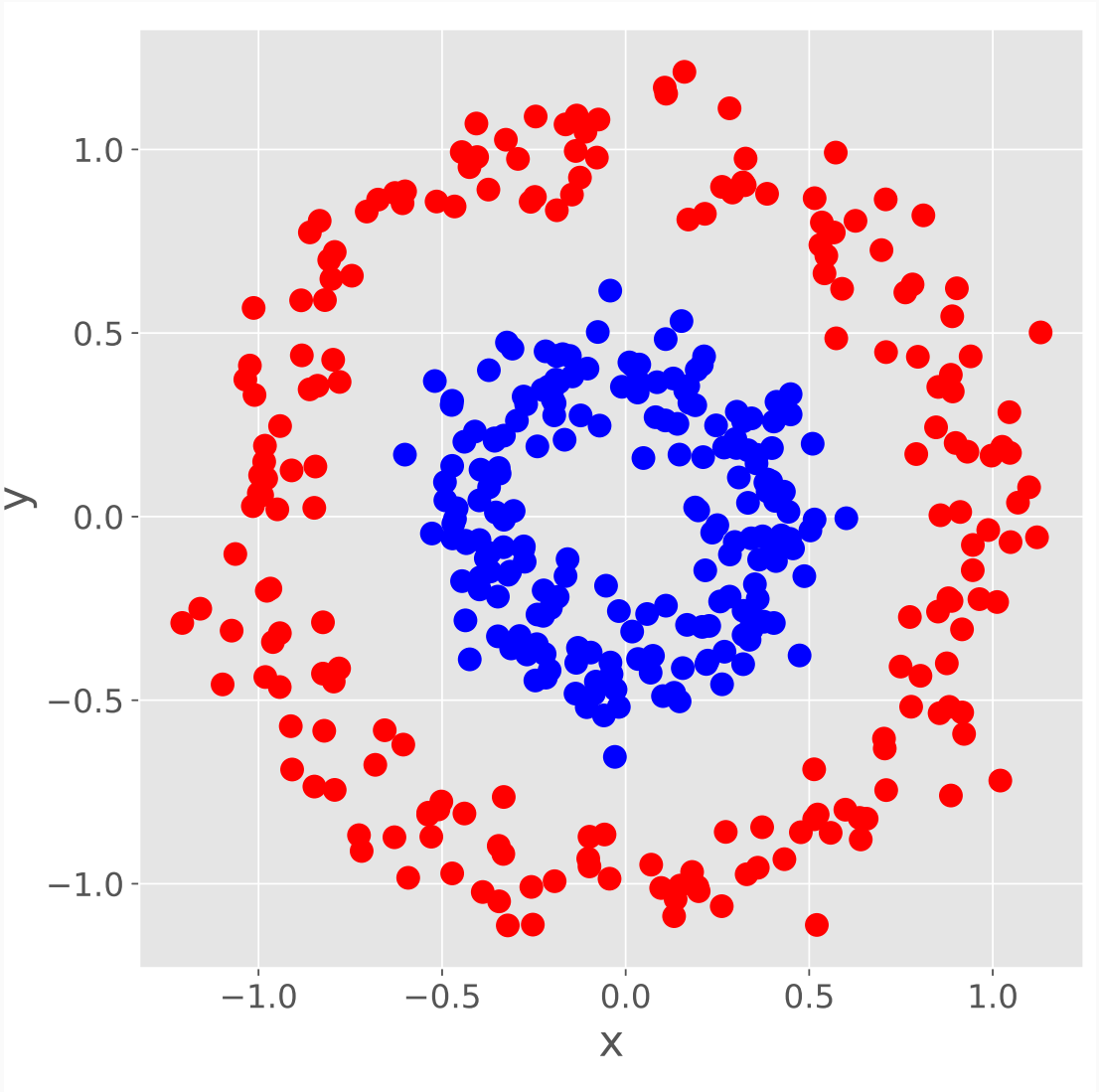
Agrupamento Espectral

Nesse ponto, temos a projeção dos nossos dados em um espaço de dimensão reduzida.

Podemos agora aplicar a técnica k-Means utilizando essa representação para obtermos os grupos.

Agrupamento Espectral

Com isso é possível encontrar os grupos corretos de nosso exemplo:



UFABC

Agrupamento Espectral - Algoritmo

```
import numpy as np

A = adjacencia(X, k)
G = grau(A)
L = G - A
w, v = np.linalg.eig(L)
idx = np.argsort(w)
X_espec = L[:, idx]
clusters = kMeans(X_espec, n_clusters)
```

UFABC

Agrupamento Espectral - Scikit-Learn

```
from sklearn.cluster import SpectralClustering

model = SpectralClustering(n_clusters=n,
                           affinity='euclidean',
                           n_neighbors=k
                          )
model.fit(X)
clusters = model.labels_
```

UFABC

Agrupamento Hierárquico

Agrupamento Aglomerativo

Dada uma base de dados X a ser agrupada, vamos definir o seguinte procedimento:

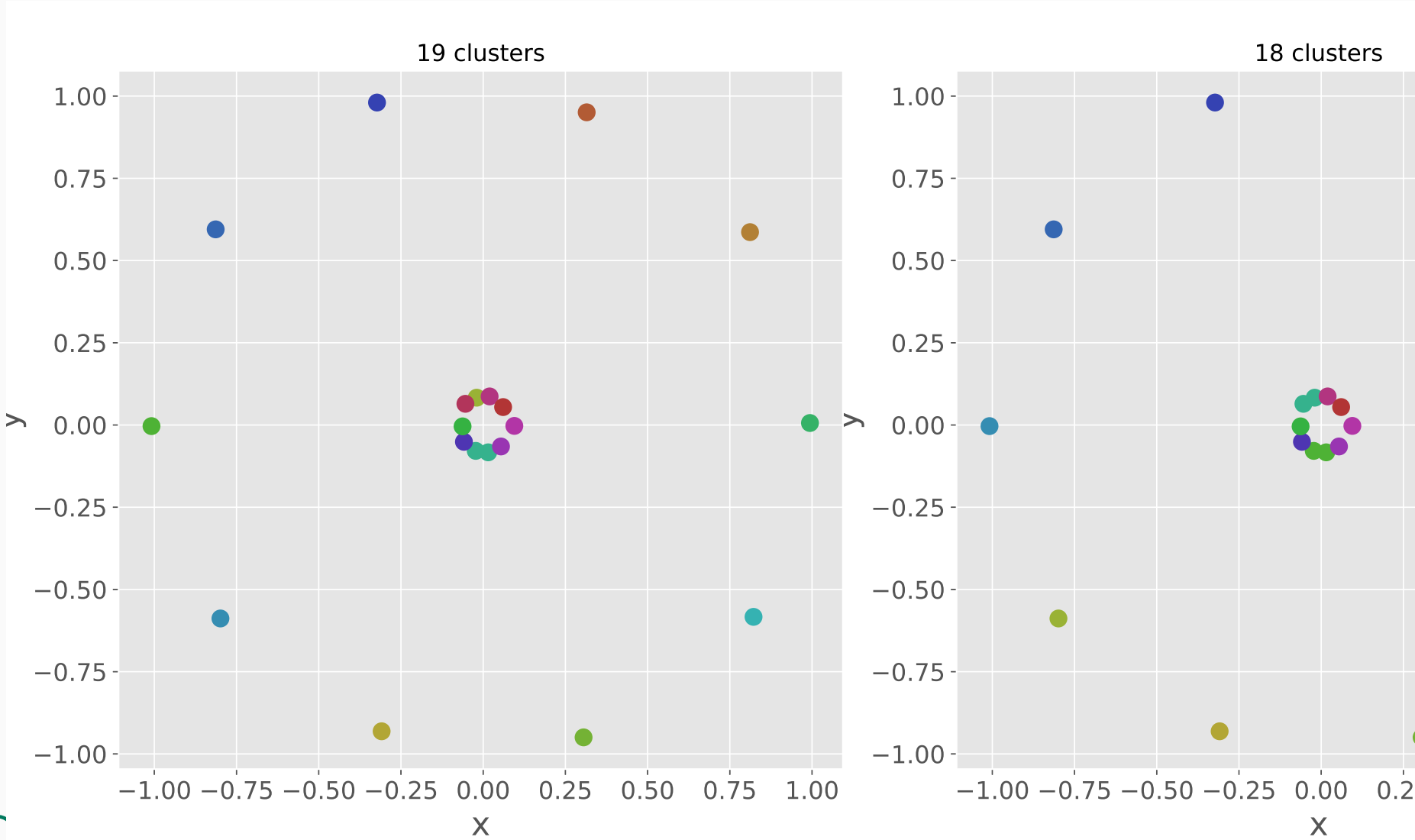
1. Defina cada ponto como um *cluster*.
2. Encontre os dois *clusters* com a menor distância entre si.
3. Crie um novo *cluster* com a união desses dois.
4. Repita o passo 2 até encontrar um número n de *clusters*.

Agrupamento Aglomerativo

Para determinar a distância entre dois *clusters* podemos calcular:

- A média das distâncias entre os pares de pontos (i, j), sendo i pertencente ao primeiro *cluster*, e j ao segundo (Average).
- A maior das distâncias entre os pares de pontos (i, j), sendo i pertencente ao primeiro *cluster*, e j ao segundo (Complete).
- A menor variância das distâncias entre os pares de pontos (i, j), sendo i pertencente ao primeiro *cluster*, e j ao segundo (Ward).

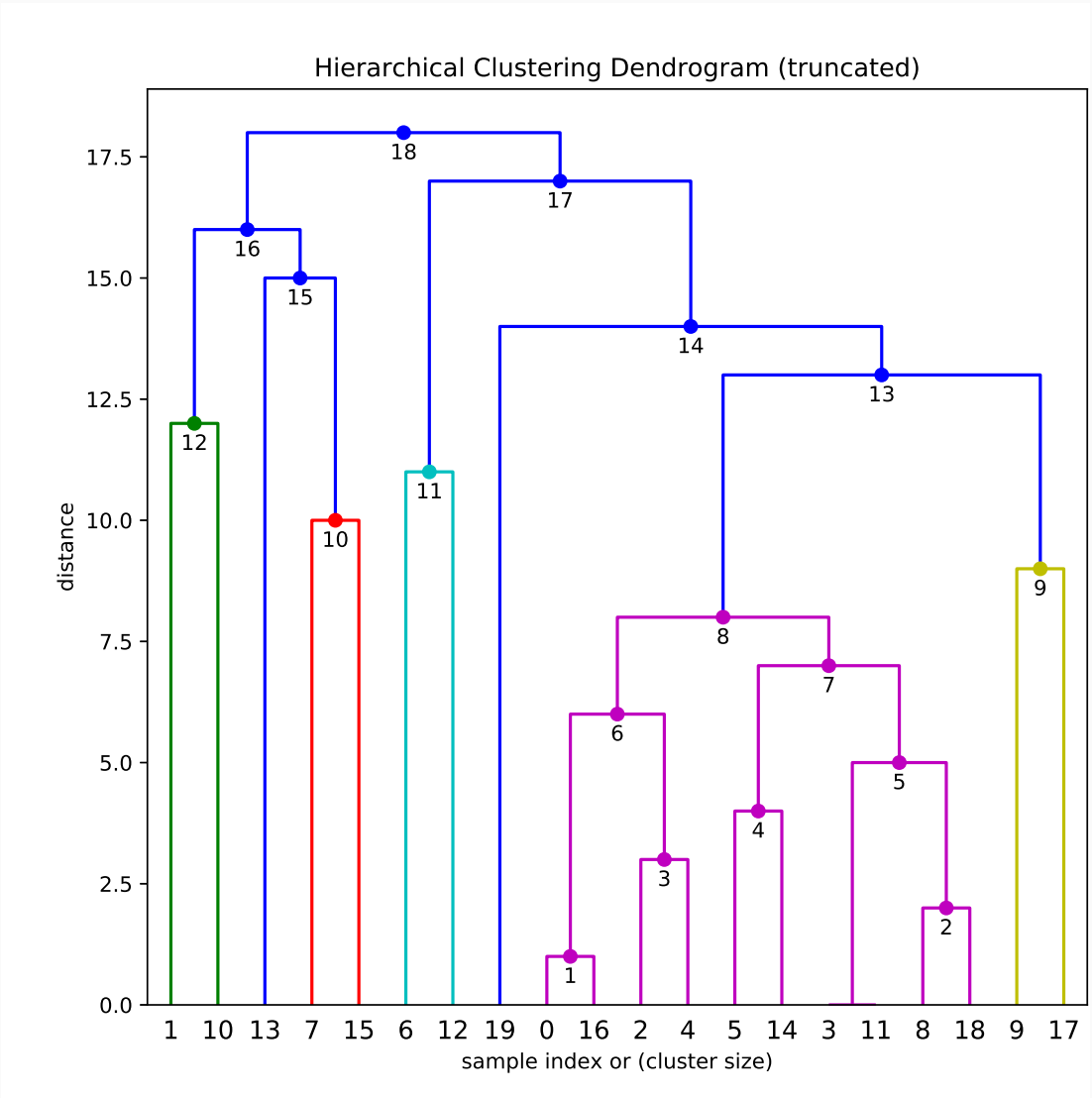
Exemplo



UFABC

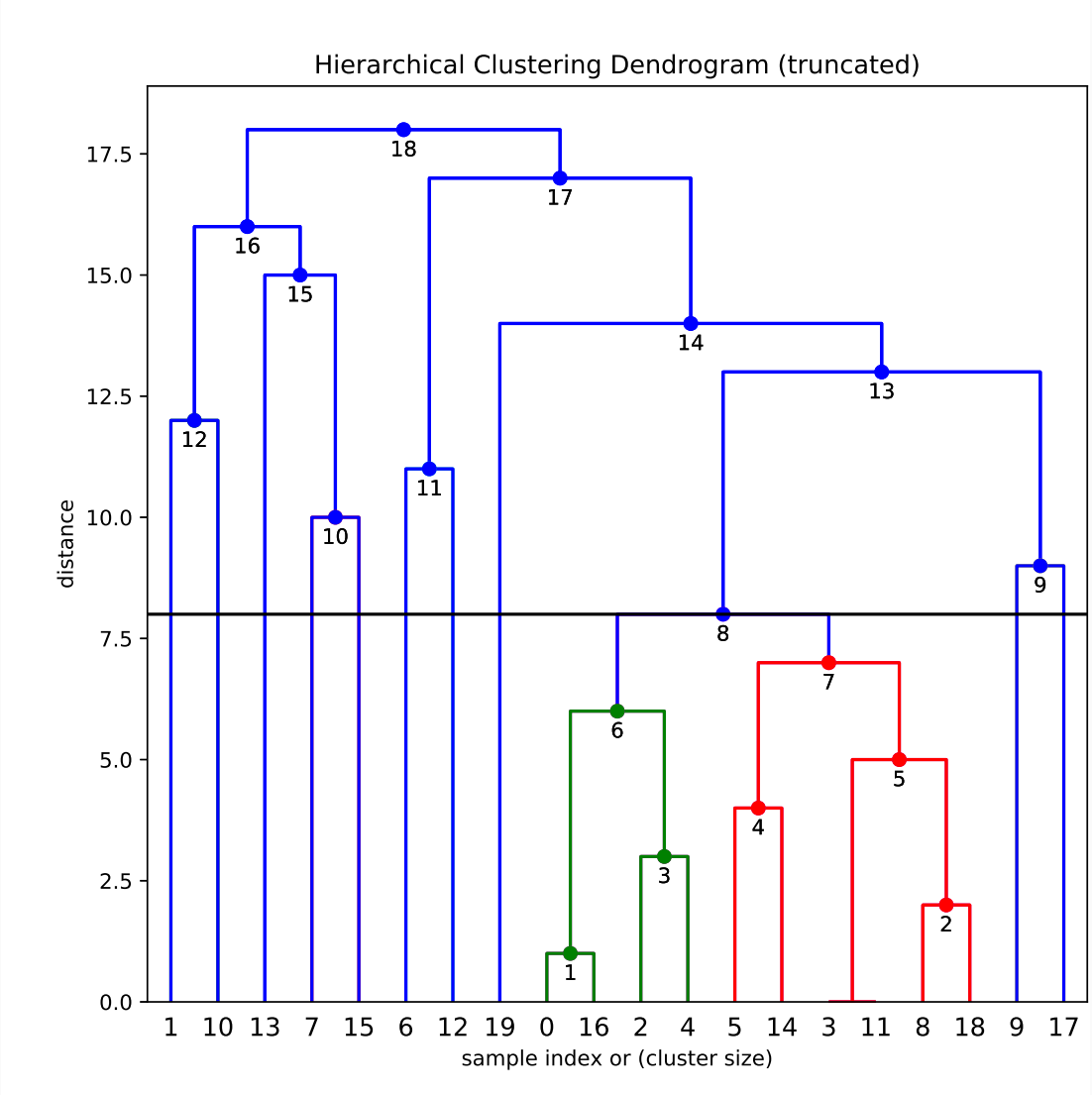
Agrupamento Hierárquico

Esse tipo de agrupamento é conhecido como agrupamento Hierárquico, pois ele induz uma hierarquia dos *clusters*.



Agrupamento Hierárquico

Isso torna possível tomarmos uma decisão mais acertada do verdadeiro número de *clusters* que são de interesse no projeto:



UFABC

Agrupamento Hierárquico

Critérios de parada:

- Número de *clusters*.
- Distância máxima.

UFABC

Agrupamento Hierárquico - Algoritmo

```
n = X.shape[0]
clusters = [[i] for i in range(n)]
n_clusters = len(clusters)

while n_clusters < k:
    clusters   = merge(clusters)
    n_clusters = len(clusters)
```

UFABC

Agrupamento Hierárquico - Algoritmo

```
def merge(clusters):
    minlink = linkage(clusters[0], clusters[1])
    merge_tuple = (0,1)
    n_clusters = len(clusters)
    for i in range(n_clusters-1):
        for j in range(i+1, n_clusters):
            if linkage(clusters[i], clusters[j]) < minlink:
                merge_tuple = (i,j)
    clusters[i] += clusters[j]
    del clusters[j]
    return clusters
```

UFABC

Agrupamento Hierárquico - Scikit-Learn

```
from sklearn.cluster import AgglomerativeClustering

model = AgglomerativeClustering(n_clusters=n,
                                affinity='euclidean',
                                linkage='ward'
                               )
model.fit(X)
clusters = model.labels_
```

UFABC

Próxima Aula

Na próxima aula aprenderemos sobre redução de dimensionalidade e extração de atributos, mais especificamente os algoritmos:

- PCA
- DCDistance
- CARFRE

UFABC

Atividade 07

Complete os Laboratórios:

Clustering_Methods_Exercises.ipynb

UFABC