

Agrupamento por similaridade - Processos clássicos

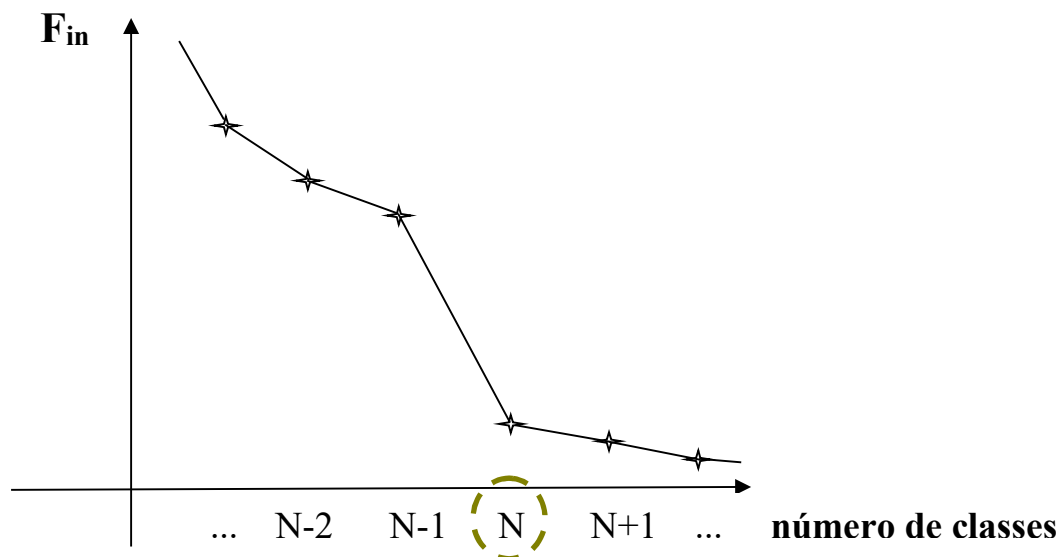
Agrupar, “de forma natural”,
conjuntos de dados com “similaridade interna”

determinar o número de classes à usar e

alocar os elementos às classes para minimizar F_{in}

1 - Determinação do número de classes a utilizar

Heurística: um agrupamento “natural” (um número N “bom” de classes) é alcançado na vizinhança imediata de uma grande variação em F_{in} .



2 - Alocação dos elementos às classes para minimizar a dispersão total intra classes F_{in}

função à minimizar

$$F_{in} = \sum_{\forall C_j} F_j \quad \text{onde} \quad F_j = n_j \sigma_j^2 = \sum_{\forall \vec{x} \in C_j} \left| \vec{x} - \vec{m}_j \right|^2$$

F_j é a dispersão intra classe da classe C_j

2.1 Localização dos padrões das classes

O padrão $\underline{p}_j$ da classe C_j deve minimizar a dispersão intra classe da classe C_j

$$F_j = \sum_{\forall \vec{x} \in C_j} \left| \vec{x} - \vec{p}_j \right|^2 = \sum_{\forall \vec{x} \in C_j} \sum_{\forall k} (x_k - p_{jk})^2 = \sum_{\forall k} \sum_{\forall \vec{x} \in C_j} (x_k - p_{jk})^2$$

Como visto anteriormente

o padrão que minimiza a dispersão intra classe (ou erro de representação) de uma classe é o seu baricentro .

$$p_{jk} = m_{jk} \quad \Rightarrow \quad \vec{p}_j = \vec{m}_j$$

2.2 – Processos de Agrupamento

Como realizar a

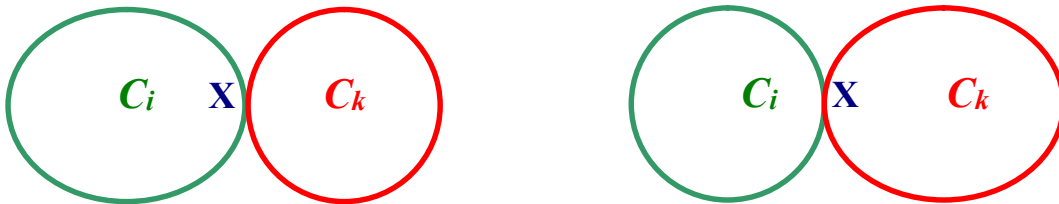
Minimização interativa de F_{in}

(e simultaneamente a maximização de F_{out}) ?

$$F_{in} = \sum_j F_j$$

Rearranjando os elementos entre as classes para reduzir F_{in}

Deslocando um elemento $\hat{x}$ da classe C_i para a classe C_k



Após alguma álgebra:



$$\vec{m}_k = \frac{1}{n_k} \sum_{\forall \vec{x} \in C_k} \vec{x} \quad \Rightarrow \quad \vec{m}'_k = \frac{1}{n'_k} \sum_{\forall \vec{x} \in C'_k} \vec{x} = \frac{1}{n_k + 1} (\hat{x} + n_k \vec{m}_k) = \vec{m}_k + \frac{\hat{x} - \vec{m}_k}{n_k + 1}$$

$$F_k = \sum_{\forall \vec{x} \in C_k} |\vec{x} - \vec{m}_k|^2 \quad \Rightarrow \quad F'_k = \sum_{\forall \vec{x} \in C'_k} |\vec{x} - \vec{m}'_k|^2 = \dots = F_k + \frac{n_k}{n_k + 1} |\hat{x} - \vec{m}_k|^2$$

$$\vec{m}_i = \frac{1}{n_i} \sum_{\forall \vec{x} \in C_i} \vec{x} \quad \Rightarrow \quad \vec{m}'_i = \frac{1}{n'_i} \sum_{\forall \vec{x} \in C'_i} \vec{x} = \frac{1}{n_i - 1} (\hat{x} + n_i \vec{m}_i) = \vec{m}_i - \frac{\hat{x} - \vec{m}_i}{n_i - 1}$$

$$F_i = \sum_{\forall \vec{x} \in C_i} |\vec{x} - \vec{m}_i|^2 \quad \Rightarrow \quad F'_i = \sum_{\forall \vec{x} \in C'_i} |\vec{x} - \vec{m}'_i|^2 = \dots = F_i - \frac{n_i}{n_i - 1} |\hat{x} - \vec{m}_i|^2$$

$$\Delta F_{in} = (F'_k - F_k) + (F'_i - F_i) = \frac{n_k}{n_k + 1} |\hat{x} - \vec{m}_k|^2 - \frac{n_i}{n_i - 1} |\hat{x} - \vec{m}_i|^2$$

F_{in} é reduzido quando deslocamos um elemento $\hat{x}$ da classe C_i para a classe C_k

se $\Delta F_{in} < 0$, i.e. se

$$\frac{n_i}{n_i - 1} \left| \hat{x} - \vec{m}_i \right| > \frac{n_k}{n_k + 1} \left| \hat{x} - \vec{m}_k \right|^2$$

como $\frac{n_i}{n_i - 1} > 1 > \frac{n_k}{n_k + 1}$ podemos simplificar a condição, basta que

$$\left| \hat{x} - \vec{m}_i \right| > \left| \hat{x} - \vec{m}_k \right|^2$$

i.e. que a distância do elemento ao baricentro da nova classe seja menor que ao da antiga.

2.3 Um processo de clusterização iterativo simples e eficaz:

K-means, C-means, Isodata ou Lloyd

Stuart Lloyd
(engenheiro telecom)

entradas: $\vec{x}_i \quad i = 1, 2, \dots, N$

centros de classe: $\vec{m}_j \quad j = 1, 2, \dots, K$

Inicialização: Arbitre K centros $\vec{m}_j \quad j = 1, 2, \dots, K$

sugestão: use as K primeiras entradas $\vec{x}_i \quad i = 1, 2, \dots, K$
não muito próximas entre si (à discutir mais tarde)

Loop**1 - Reclassificação das entradas**

$$\forall \vec{x}_i \quad i = 1, 2, \dots, N$$

Se $\vec{m}_j$ é a melhor representação de $\vec{x}_i$, i.e. se

$$|\vec{x}_i - \vec{m}_j| < |\vec{x}_i - \vec{m}_k| \quad \forall k \neq j \quad \text{então} \quad \vec{x}_i \in C_j$$

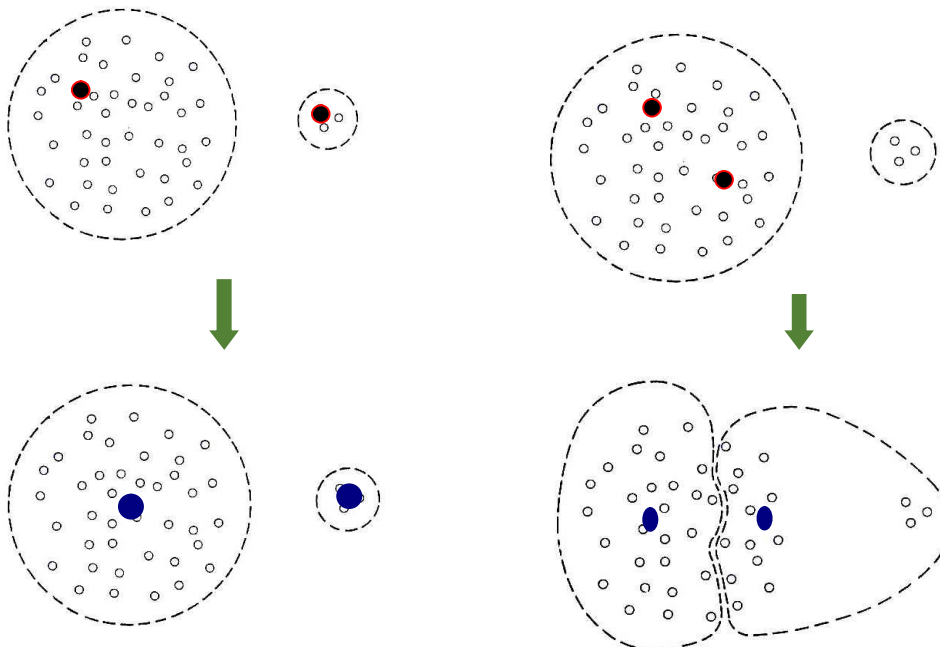
2 - Atualização dos centros de classe

$$\forall j = 1, 2, \dots, K$$

$$\vec{m}_j = \frac{1}{n_j} \sum_{\vec{x} \in C_j} \vec{x}$$

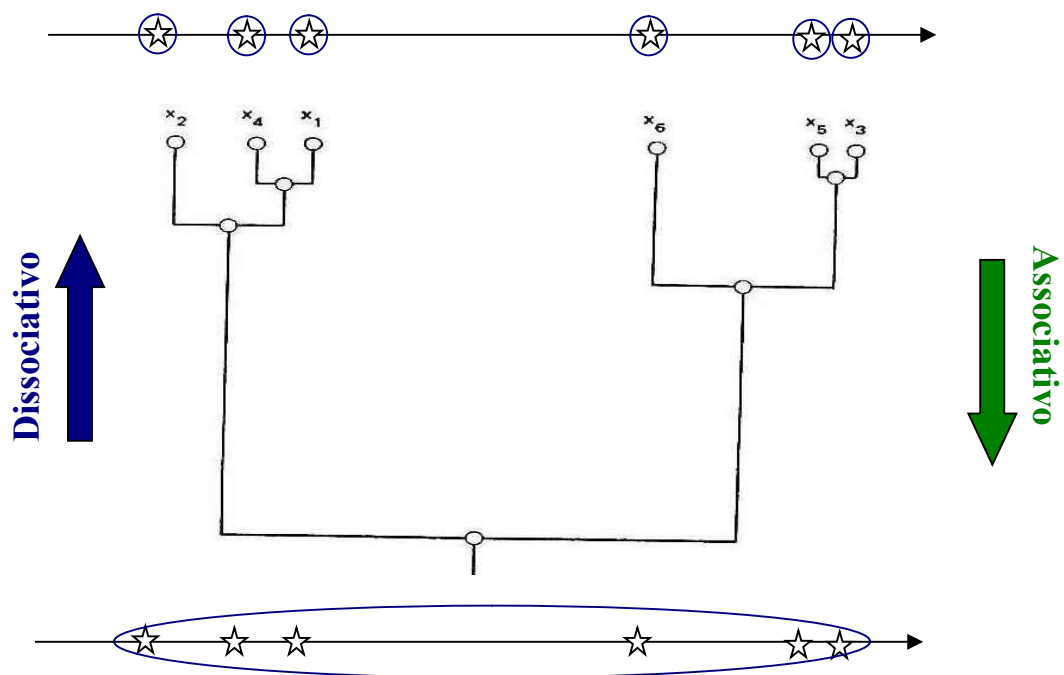
3 - Retorne ao passo 1

Quando nenhum elemento trocar de classe: Fim.



inicializar com entradas não muito próximas

2.4 - Algoritmos Hierárquicos: dendogramas, árvores

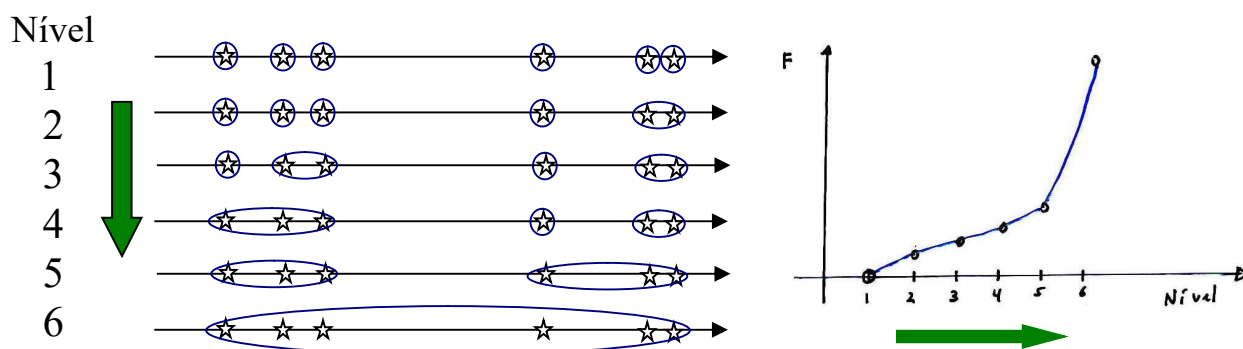


2.4.1 Algoritmo Aglomerativo, Associativo:

n elementos c classes

$$c = \quad n \quad \quad n-1 \quad \quad n-2 \quad \dots \quad 2 \quad \quad 1$$

$$F_{in} = \quad F_n=0 \quad < \quad F_{n-1} \quad < \quad F_{n-2} \quad < \quad F_2 \quad < \quad F_1$$



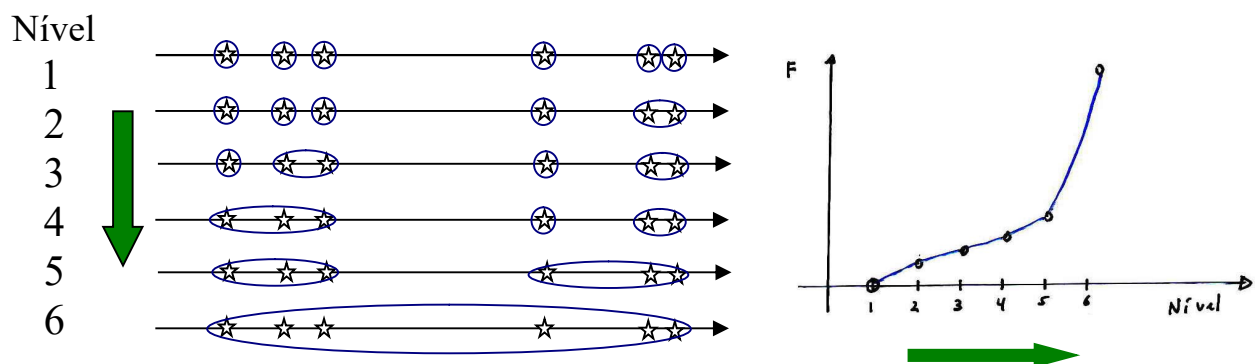
Algoritmo Hierárquico Associativo:

Inicialização: n elementos $\Rightarrow$ n clusters

Loop:

Junte os dois clusters que aumentam F_{in} o mínimo possível *

Repetir até chegar ao número de clusters desejado
ou F_{in} crescer muito



Obs:

se $\underline{x}_i$ e $\underline{x}_j$ estão em um mesmo cluster, permanecerão juntos.

É dito um **algoritmo hierárquico**.

Que clusters juntar ?

$$C_j \quad \vec{m}_j = \frac{1}{n_j} \sum_{\forall \vec{x} \in C_j} \vec{x} \quad F_j = \sum_{\forall \vec{x} \in C_j} |\vec{x}|^2 - n_j |\vec{m}_j|^2$$

$$C_k \quad \vec{m}_k = \frac{1}{n_k} \sum_{\forall \vec{x} \in C_k} \vec{x} \quad F_k = \sum_{\forall \vec{x} \in C_k} |\vec{x}|^2 - n_k |\vec{m}_k|^2$$

$$C_{jk} = C_j \cup C_k \quad \vec{m}_{jk} = \frac{n_j \vec{m}_j + n_k \vec{m}_k}{n_j + n_k} \quad F_{jk} = \sum_{\forall \vec{x} \in C_j, C_k} |\vec{x}|^2 - (n_j + n_k) |\vec{m}_{jk}|^2$$

$$\Delta F = F_{jk} - (F_j + F_k) = \frac{n_j n_k}{n_j + n_k} |\vec{m}_j - \vec{m}_k|^2$$

$$\Delta F = \frac{n_j n_k}{n_j + n_k} |\vec{m}_j - \vec{m}_k|^2 \quad \text{deve ser pequeno}$$

escolher centros próximos:

$$\vec{m}_j \approx \vec{m}_k$$

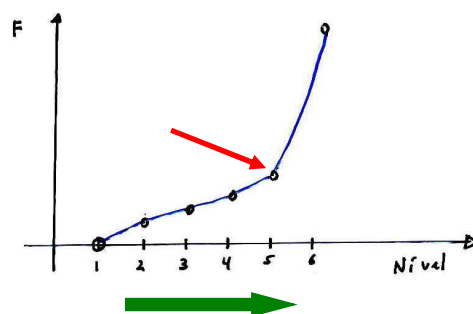
de preferência com populações desiguais:

$$\frac{n_j n_k}{n_j + n_k} \approx \begin{cases} n_j & \text{se } n_j \ll n_k \\ \frac{n_j}{2} & \text{se } n_j \approx n_k \end{cases}$$

Quando parar o processo ?

Heurística: um agrupamento “natural” (um número “bom” de classes) é alcançado no algoritmo associativo imediatamente antes de F_{in} apresentar um grande acréscimo em um único passo.

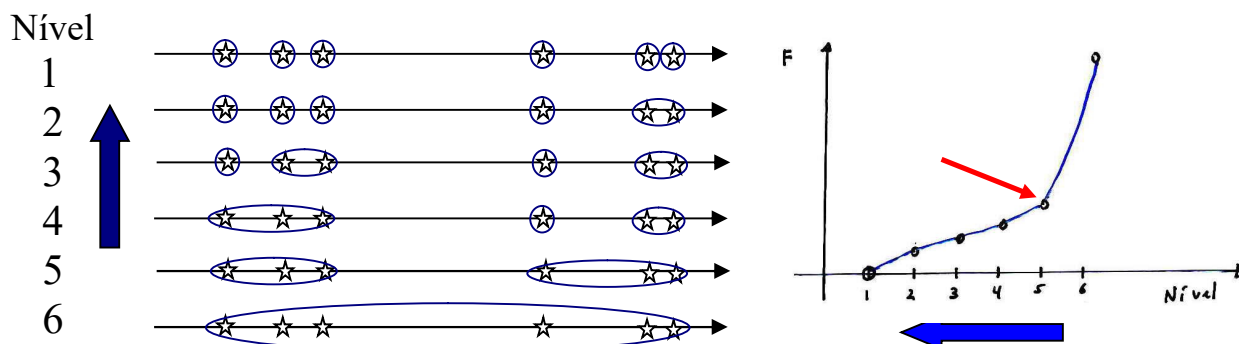
Por este critério o ponto de parada está indicado pela seta vermelha no exemplo anterior.



2.4.2 Algoritmo Divisivo, Dissociativo:

n elementos c classes

$c =$ 1 2 3 ... n-1 n
 $F_{in} =$ $F_1 < F_2 < F_3 < F_{n-1} < F_n = 0$



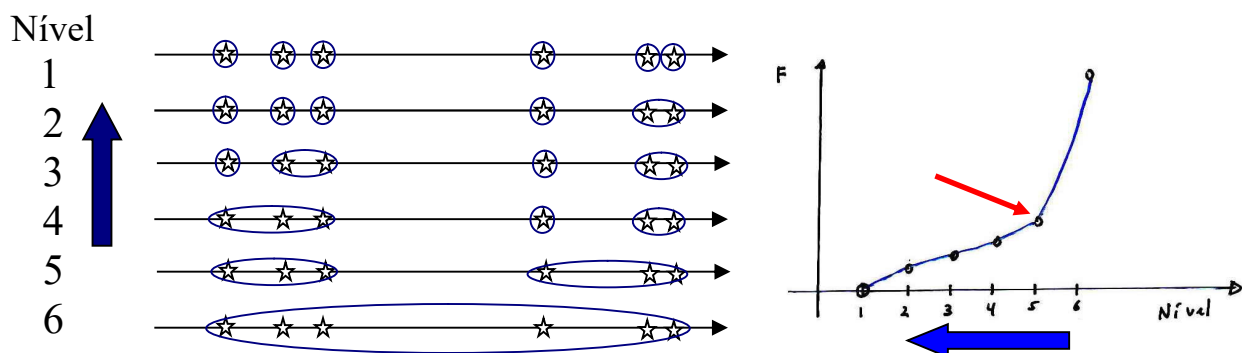
Algoritmo Hierárquico Dissociativo:

Inicialização: n elementos $\Rightarrow$ 1 cluster

Loop:

Divida o cluster que reduz F_{in} o máximo possível *

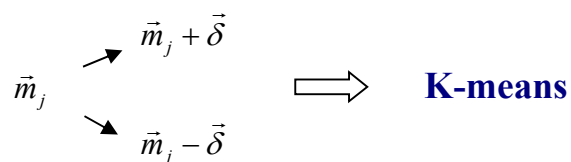
Repetir até chegar ao número de clusters desejado
ou F_{in} quase não decair



Que cluster dividir ?

Critério: maior F_j , por exemplo

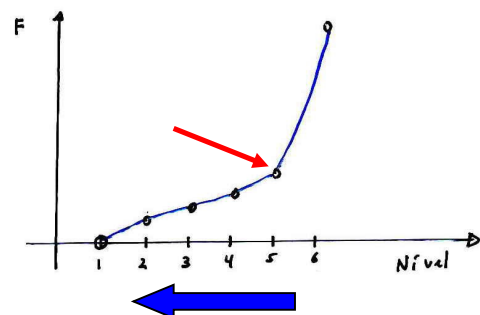
Método:



Quando parar o processo ?

Heurística: um agrupamento “natural” (um número “bom” de classes) é obtido no algoritmo dissociativo imediatamente após F_{in} apresentar um grande decréscimo em um único passo.

Por este critério o ponto de parada está indicado pela seta vermelha no exemplo anterior.



2.4.3 Que classes agrupar / dissociar

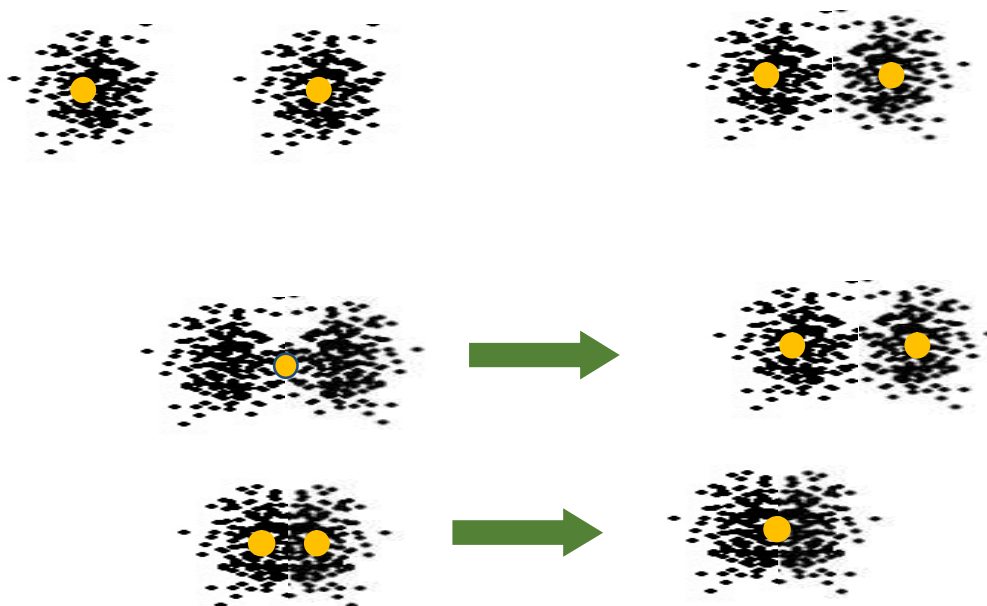
No caso em que as classes são formadas por padrões adicionados de ruído branco, como discutido no capítulo sobre Ruído (cap 07), a decisão sobre unir ou dissociar duas classes C_i e C_j deve ser tomada

1 - classes com desvio padrão muito maior que a média são candidatas a serem dissociadas

2 - duas classes com desvios padrão menores que a média e centros muito próximos são candidatas a serem agrupadas em uma única classe.

Comparar os desvios padrão das classes com a distância entre os centros das classes, por exemplo:

$$s = \frac{|\vec{m}_i - \vec{m}_j|}{(2\sigma_i + 2\sigma_j)} > 1 \quad ?$$



2.5 - Algoritmos Mistos

Iniciar com K-means

Associar / Dissociar verificando a variação de F_{in}