Clustering Hierárquico Aglomerativo

- **Matriz de proximidade: NxN**
- **D(i,j): medida de proximidade ou similaridade entre os padrões i e j**

1. **Atribuir um padrão por cluster (N clusters)**
2. **Encontrar o par de clusters Ci e Cj mais semelhantes na matriz de similaridade e juntá-los num único cluster**
3. **Actualizar a matriz de similaridade entre o novo cluster e todos os restantes**
4. **Repetir os passos (2) e (3) até que se tenha um único cluster (i.e. N-1 vezes)**

Medida de semelhança entre clusters

- **A maioria dos métodos de cálculo de similaridade entre clusters pode ser expresso por uma única fórmula que dá a similaridade (ou distância) entre um cluster *k* e o cluster *i+j* formado pela junção dos clusters *i* e *j*:**

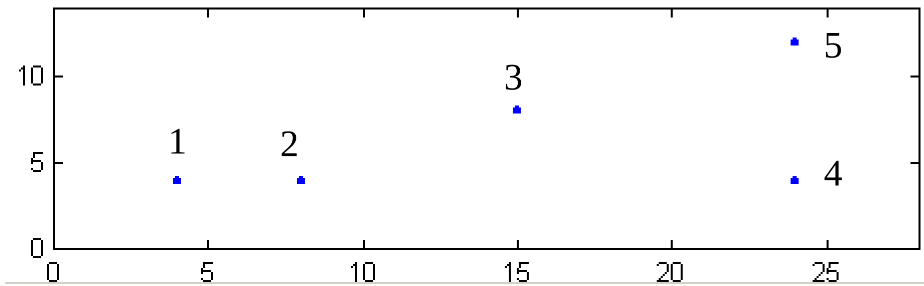
$$d(i+j,k) = a_i d(i,k) + a_j d(j,k) + b d(i,j) + c\left|d(i,k) - d(j,k)\right|$$

Single-link (ligação simples)	$a_i = a_j = 0.5$; $b = 0$; $\mathrm{c} = -0.5$ $d(i+j,k) = \min\{d(i,k), d(j,k)\}$
Complete-link (ligação completa)	$a_i = a_j = 0.5$; $b = 0$; $\mathrm{c} = 0.5$ $d(i+j,k) = \max\{d(i,k), d(j,k)\}$
Centroide	$a_i = \frac{n_i}{n_i + n_j}$ $a_j = \frac{n_j}{n_i + n_j}$ $b = -\frac{n_i n_j}{(n_i + n_j)^2}$ $c = 0$ $d(i+j,k) = d(\mu_{i+j}, \mu_k)$
Mediana	$a_i = a_j = 0.5$; $b = -0.25$; $\mathrm{c} = 0$
Média do Grupo (Average link)	$a_i = \frac{n_i}{n_i + n_j}$ $a_j = \frac{n_j}{n_i + n_j}$ $b = c = 0$ $d(C_i, C_j) = \frac{1}{n_i n_j} \sum_{a \in C_i, b \in C_j} d(a,b)$
Método de Ward (variância mínima)	$a_i = \frac{n_k + n_i}{n_k + n_i + n_j}$ $a_j = \frac{n_k + n_j}{n_k + n_i + n_j}$ $b = -\frac{n_k}{n_k + n_i + n_j}$ $c = 0$

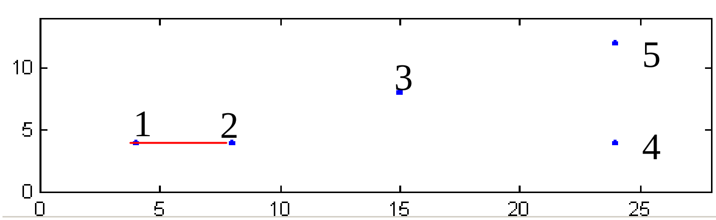
Ex: Single-Link

$$d(C_i, C_j) = \min_{a \in C_i, b \in C_j} \{d(a,b)\}$$

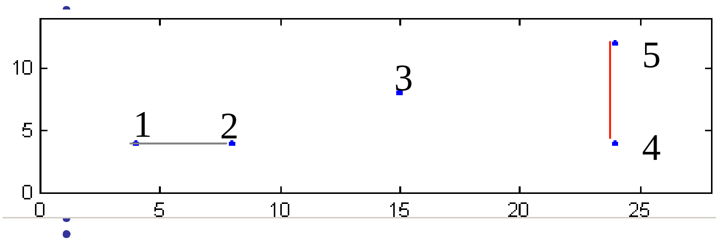
	x	y
1	4	4
2	8	4
3	15	8
4	24	4
5	24	12



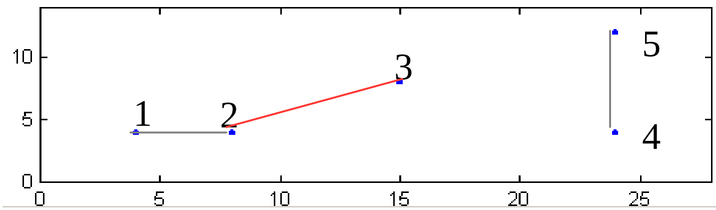
	1	2	3	4	5
1	-	4	11.7	20	21.5
2	4	-	8.1	16	17.9
3	11.7	8.1	-	9.8	9.8
4	20	16	9.8	-	8
5	21.5	17.9	9.8	8	-



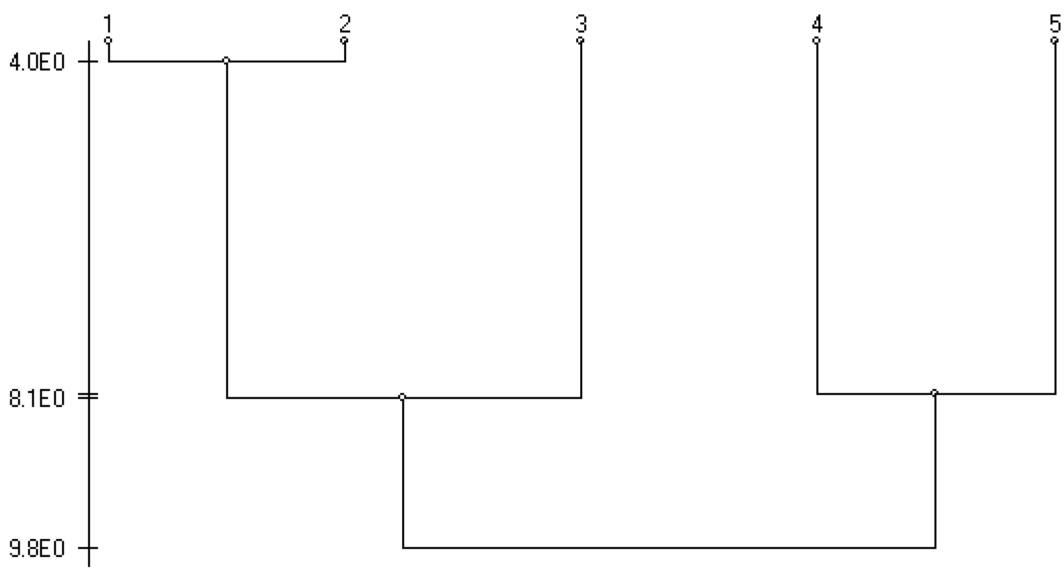
	1,2	3	4	5
1,2	-	8.1	16	17.9
3	8.1	-	9.8	9.8
4	16	9.8	-	8
5	17.9	9.8	8	-



	1,2	3	4,5
1,2	-	8.1	16
3	8.1	-	9.8
4,5	16	9.8	-

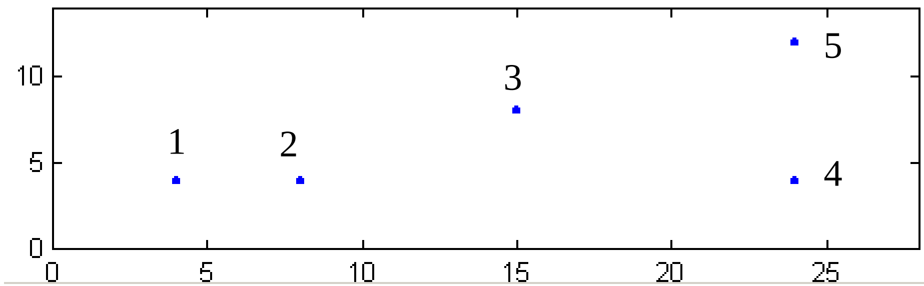


	1,2,3	4,5
1,2,3	-	9.8
4,5	9.8	-

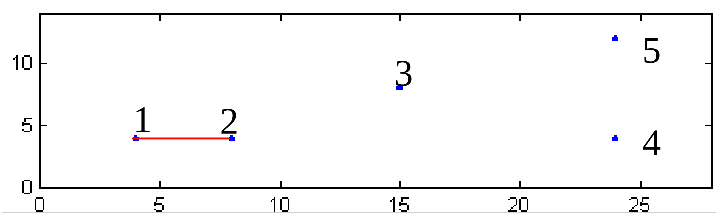


Complete-Link

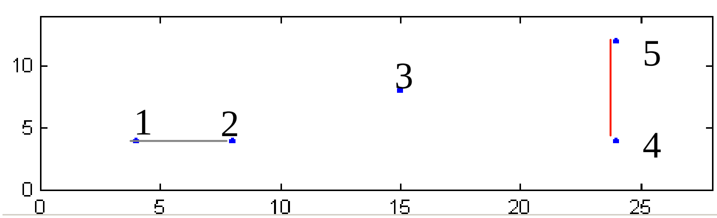
$$d(C_i, C_j) = \max_{a \in C_i, b \in C_j} \{d(a,b)\}$$



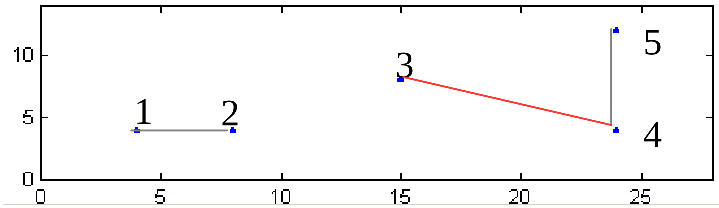
	1	2	3	4	5
1	-	4	11.7	20	21.5
2	4	-	8.1	16	17.9
3	11.7	8.1	-	9.8	9.8
4	20	16	9.8	-	8
5	21.5	17.9	9.8	8	-



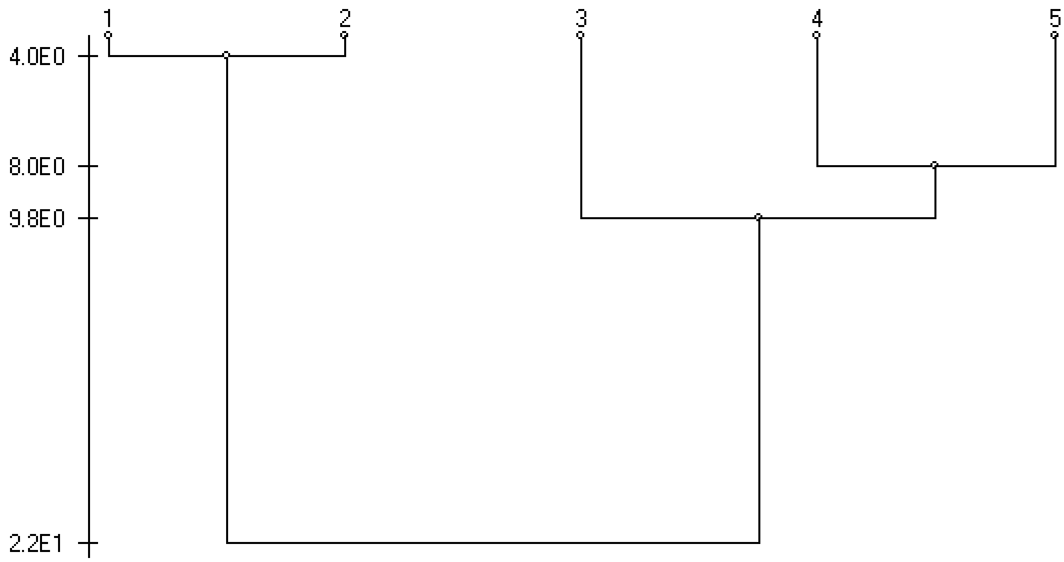
	1,2	3	4	5
1,2	-	11.7	20	21.5
3	11.7	-	9.8	9.8
4	20	9.8	-	8
5	21.5	9.8	8	-

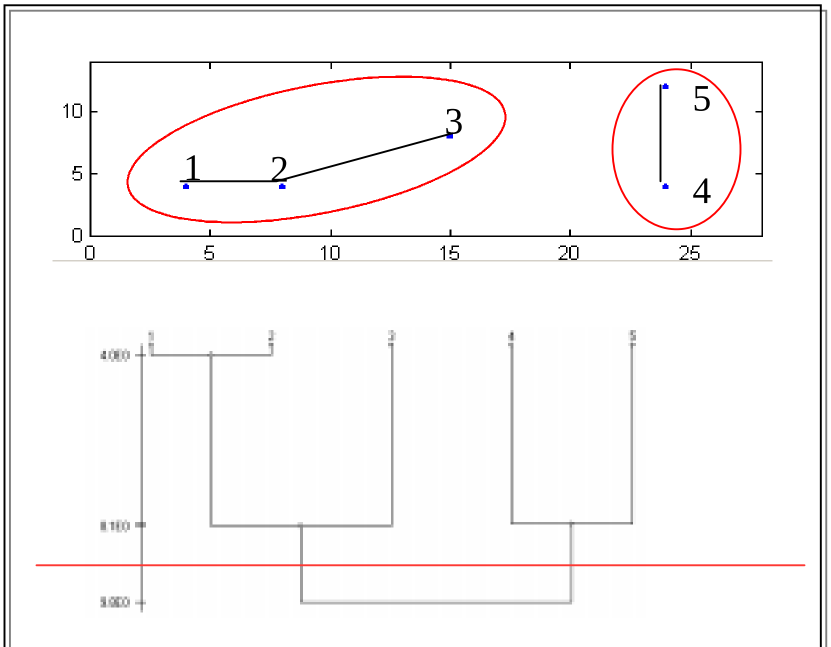


	1,2	3	4,5
1,2	-	11.7	21.5
3	11.7	-	9.8
4,5	21.5	9.8	-

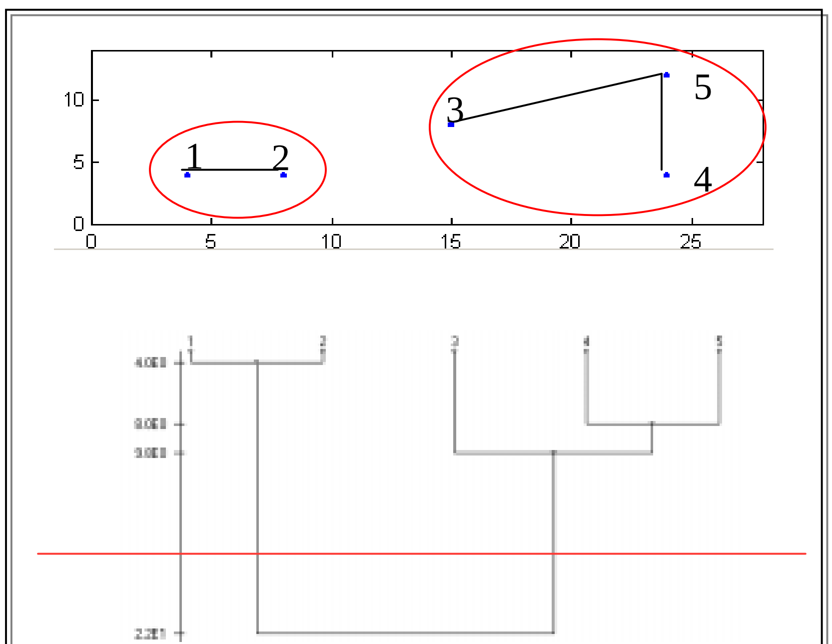


	1,2,3	4,5
1,2,3	-	21.5
4,5	21.5	-





Single-link



Complete-link

Técnicas Particionais de Clustering

- ## Algoritmo de Lloyd-Max Generalizado
  - $x \in \Re^d$, com f.d.p. $p(x)$ conhecida
  - **Pretende-se calcular uma partição $\{R_1, \dots R_C\}$ de $\Re^d$ sendo cada classe representada por um vector $\hat{x}_i \in \Re^d$ designado por *centroide***
  - **A escolha dos centroides e da partição é feita por forma a minimizar a funcional de custo**

$$D = E\{d(x, q(x))\}$$

com

    - » ***q(x)*** **uma função que transforma cada vector *x* no centroide da sua classe,** $\hat{x}_i \in \Re^d$
    - » ***d(.,.)*** **– medida de distância em** $\Re^d$

  - **Seja *Pi* a função característica do conjunto i**

$$P_i(x) = \begin{cases} 1 & \text{se } x \in R_i \\ 0 & \text{caso contrário} \end{cases}$$

$$\therefore \quad D = E\left\{ \sum_{i=1}^{C} P_i(x) d(x, \hat{x}_i) \right\}$$

  - **O classificador obtem-se calculando uma partição e um conjunto de representantes que minimizem *D***

- **<u>Decomposição do problema em dois problemas mais simples</u>:**
  1. **Admite-se conhecida a partição do espaço e calcula-se o conjunto óptimo de centroides**
  2. **Admitem-se conhecidos os centroides e calcula-se a partição óptima do espaço**

1: Seja $\{R_1, \ldots R_C\}$ uma partição do espaço. Pretende-se calcular $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_C$ por forma a minimizar *D*. Admitindo que *D* é diferenciável, uma condição necessária para o mínimo é:

$$\frac{\partial D}{\partial \hat{x}_p} = 0, \quad p = 1, \ldots, C$$

Como as funções de pertença só dependem da partição, vem:

$$\frac{\partial D}{\partial \hat{x}_p} = \frac{\partial}{\partial \hat{x}_p} E\left\{\sum_{i=1}^{C} P_i(x) d(x, \hat{x}_i)\right\} = E\left\{P_p(x) \frac{\partial}{\partial \hat{x}_p} d(x, \hat{x}_p)\right\} = 0$$

ou seja

$$\boxed{\int_{R_p} p(x) \frac{\partial}{\partial \hat{x}_p} d(x, \hat{x}_p) dx = 0 \qquad p = 1, \ldots, C}$$

- **No caso em que *d(.,.)* é o quadrado da norma Euclideana**

$$d(x, \hat{x}) = (x - \hat{x})^T (x - \hat{x})$$

tem-se

$$\int_{R_p} p(x) \frac{\partial}{\partial \hat{x}_p} (x - \hat{x}_p)^T (x - \hat{x}_p) dx = \int_{R_p} -2\, p(x)(x - \hat{x}_p) dx = 0$$

$$\int_{R_p} p(x) \hat{x}_p dx = \int_{R_p} x p(x) dx \qquad \boxed{\hat{x}_p = \frac{\int_{R_p} x p(x) dx}{\int_{R_p} p(x) dx}}$$

2: Suponhamos agora que os centroides são conhecidos. Qual a partição que minimiza *D* ?

$$D = E\left\{\sum_{i=1}^{C} P_i(x) d(x, \hat{x}_i)\right\}$$

A minimização de *D* pode ser feita minimizando a função integranda em cada ponto *x*

$$D_x = \sum_{i=1}^{C} P_i(x) d(x, \hat{x}_i)$$

Esta função é minimizada se *x* for classificada na classe do centroide mais próximo:

$$\boxed{\omega_k : k = \arg \min_{\mathrm{i}} d(x, \hat{x}_i)}$$

- **A resolução simultânea das equações anteriores não é fácil. A solução adoptada consiste em estimar alternadament os centroides e a partição do espaço.**

Algoritmo de Lloyd-Max Generalizado I

- **Hipótese: *p(x)* conhecido**

1. **Inicialização**

   Escolha o nº *C* de classes e inicialize os centroides de cada classe de acordo com algum critério

2. **Cálculo da partição**

   Calcule a partição de $\Re^d$ associando cada ponto à região com centroide mais próximo

$$x \in R_k : k = \arg \min_{i=1,\ldots,C} d(x, \hat{x}_i)$$

3. **Actualização dos centroides**

   Calcule novos centroides para cada uma das classes através da equação

$$\int_{R_p} p(x) \frac{\partial}{\partial \hat{x}_p} d(x, \hat{x}_p) dx = 0 \qquad p = 1, \ldots, C$$

   - ***Se d(.,.)* é o quadrado da norma Euclideana**

$$\boxed{\hat{x}_p = \frac{\int_{R_p} x p(x) dx}{\int_{R_p} p(x) dx} \quad \mathrm{p} = 1, \ldots, \mathrm{C}}$$

4. **Voltar ao ponto 2 até se verificar um critério de paragem**

- **Na prática, raramente se conhece a f.d.p *p(x)***

- **Um procedimento muito utilizado consiste em substituir a esperança matemática pela média sobre todos os padrões de treino, ou seja**

$$D = \sum_{x \in \mathrm{X}} d(x, q(x))$$

- **Os algoritmos de Forgy e k-médias têm por objectivo minimizar o quadrado do erro para um número fixo de clusters**

$$\boxed{D = \sum_{i=1}^{C} \sum_{j=1}^{N_i} (x_j - \mu_i)^T (x_j - \mu_i)}$$

Algoritmo de Lloyd-Max Generalizado II

- **Hipótese: *X* conhecido**

1. <u>Inicialização</u>

Escolha o nº *C* de classes e inicialize os centroides de cada classe de acordo com algum critério

2. <u>Cálculo da partição</u>

Calcule a partição $\{X_1, X_2, ..., X_C\}$ de $\Re^d$ **associando cada ponto à região com centroide mais próximo**

$$x \in X_k : k = \arg \min_{\mathrm{i=1,...,C}} d(x, \hat{x}_i)$$

3. <u>Actualização dos centroides</u>

Calcule novos centroides para cada uma das classes através da equação

$$\sum_{x \in X_p} \frac{\partial}{\partial \hat{x}_p} d(x, \hat{x}_p) = 0 \qquad p = 1, \ldots, C$$

em que Xp designa o conj. de pontos atraibuídos à p-ésima classe

- ***Se d(.,.)* é o quadrado da norma Euclideana**

$$\boxed{\hat{x}_p = \frac{1}{N_p} \sum_{x \in X_p} x \qquad \mathrm{p} = 1, ..., \mathrm{C}}$$

4. Voltar ao ponto 2 até se verificar um critério de paragem

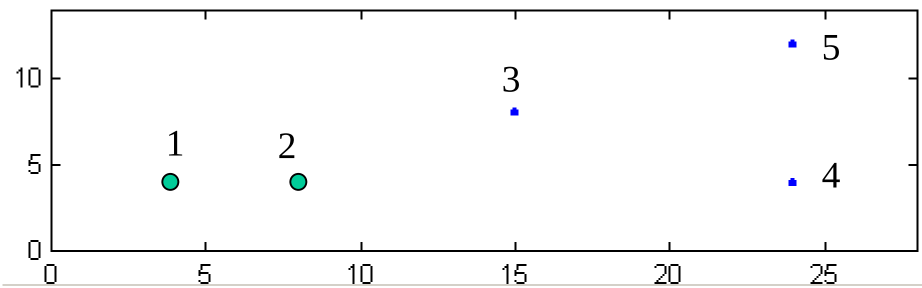
Técnicas Particionais de Clustering

- **Algoritmo de Forgy**
- **Input:**
  - **C: número de clusters**
  - **C padrões seleccionados aleatoriamente, designados pontos semente**

1. **Inicializar os centroides dos clusters nos pontos semente**
2. **Para cada amostra, determinar o centroide mais próximo. Colocar a amostra nesse cluster**
3. **Se nenhuma amostra mudou de cluster no passo 2, parar**
4. **Calcular os centroides dos clusters resultantes e voltar ao passo 2.**

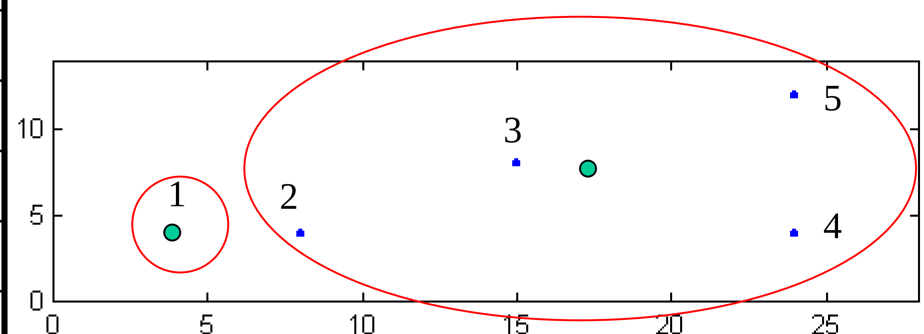
Ex: Algoritmo de Forgy

	x	y
1	4	4
2	8	4
3	15	8
4	24	4
5	24	12



Fixe C=2

	Centroide mais próximo
1	(4,4)
2	(8,4)
3	(8,4)
4	(8,4)
5	(8,4)



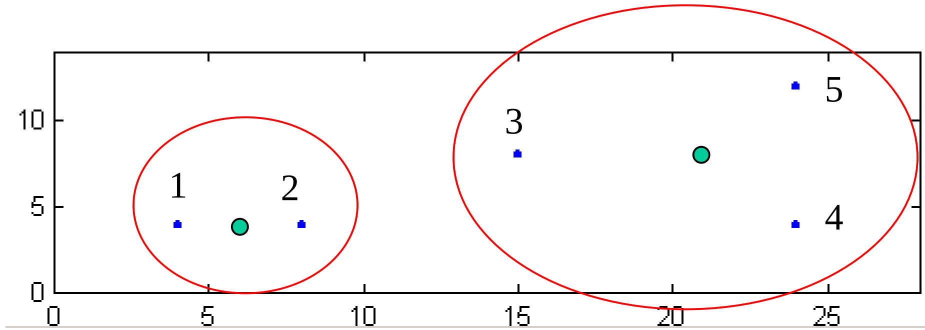
Cálculo do novo centroide para o 2º cluster:

$$\mu_2 = (\bar{x}, \bar{y})$$

$$\bar{x} = \frac{8+15+24+24}{4} = 17.75 \qquad \bar{y} = \frac{4+8+4+12}{4} = 7.0$$

Nova iteração:

	Centroide mais próximo
1	(4,4)
2	(4,4)
3	(17.75,7)
4	(17.75,7)
5	(17.75,7)



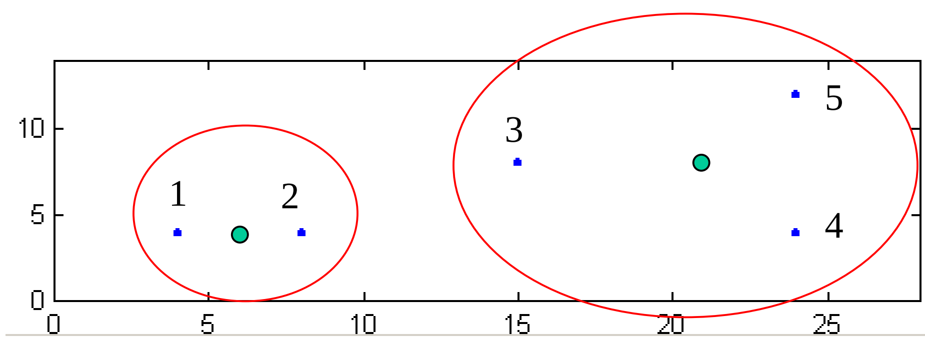
Uma vez que algumas amostras mudaram de cluster, repetir o cálculo dos centroides e prosseguir:

$$\mu_1 = \left(\frac{4+8}{2}, 4\right) = (6,4)$$

$$\mu_2 = \left(\frac{15+24+24}{3}, \frac{8+4+12}{3}\right) = (21,8)$$

Nova iteração:

	Centroide mais próximo
1	(6,4)
2	(6,4)
3	(21,8)
4	(21,8)
5	(21,8)



- **Algoritmo k-médias**

- **Diferença em relação ao Forgy:**
  - **Os centroides dos clusters são recalculados logo que uma amostra é adicionada a um luster**
  - **Só executa dois passos sobre os dados**

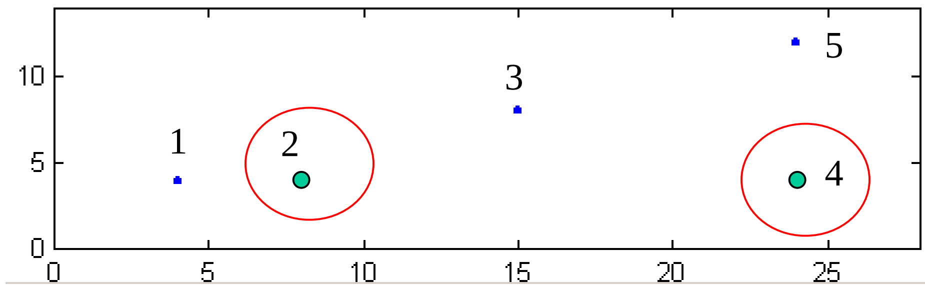
- **Input:**
  - **C: número de clusters**
  - **C padrões seleccionados aleatoriamente, designados pontos semente**

1. **Começar com C clusters, cada um com uma amostra semente. Para cada uma das N-C amostras restantes, encontrar o centroide mais próximo. Colocar a amostra nesse cluster e recalcular o centroide do cluster alterado.**
2. **Ir ao longo de todas as amostras uma segunda vez. Para cada amostra, encontrar o centroide mais próximo e colocá-la nesse cluster (neste passo não são recalculados os centroides).**

Clustering k-médias: C=2

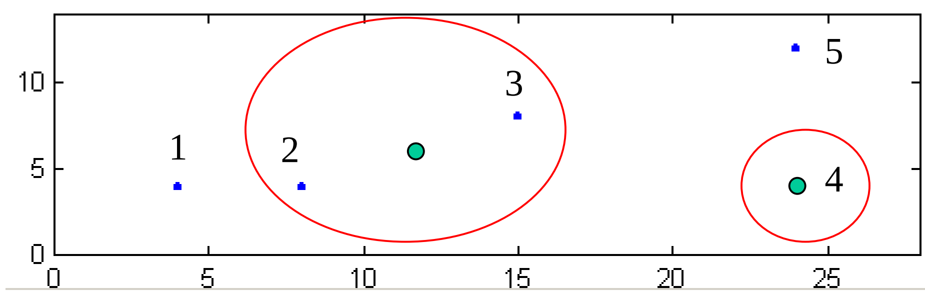
Começar por ex. com os clusters {(8,4)} e {(24,4)}

	x	y
2	8	4
4	24	4
3	15	8
1	4	4
5	24	12



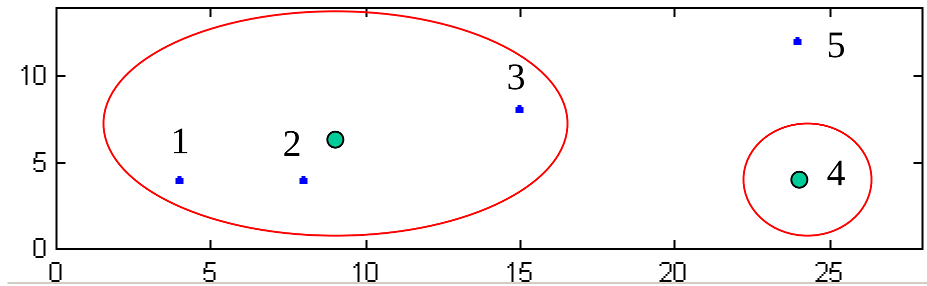
Cluster	Centroide
(8,4)	(8,4)
(24,4)	(24,4)

Ponto seguinte: (15,8) -> está mais perto de (8,4)



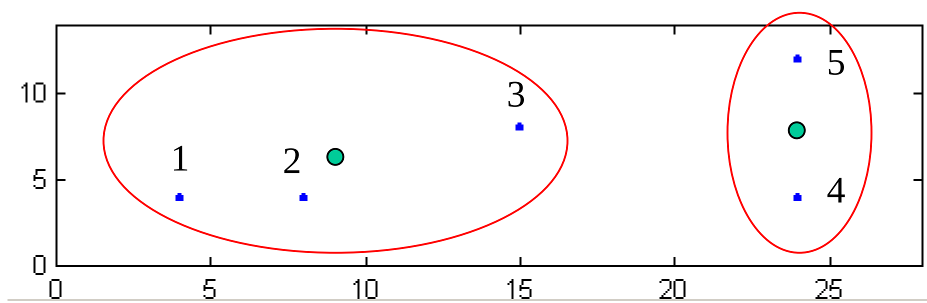
Cluster	Centroide
(8,4)(15,8)	(11.5,6)
(24,4)	(24,4)

Ponto seguinte: (4,4) -> está mais perto de (11.5,6)



Cluster	Centroide
(8,4)(15,8)(4,4)	(9,5.3)
(24,4)	(24,4)

Ponto seguinte: (24,12) -> está mais perto de (24,4)



Cluster	Centroide
(8,4)(15,8)(4,4)	(9,5.3)
(24,4)(24,12)	(24,8)

Passo 2: não há necessidade de alteração dos clusters.