

# Algoritmos de Agrupamento Hierárquicos

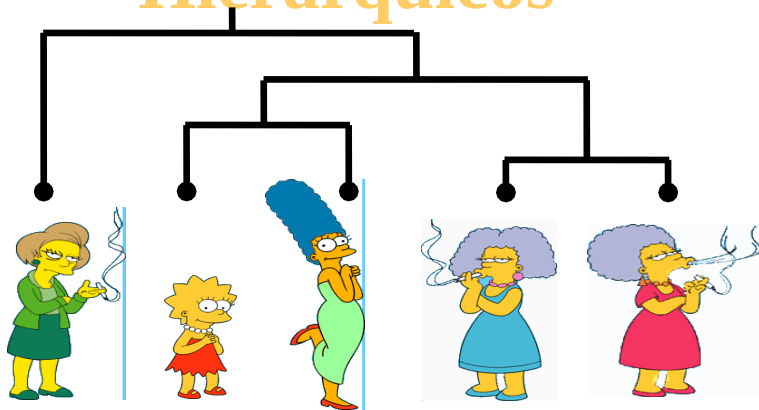
## Mineração de Dados

Ronaldo C. Prati

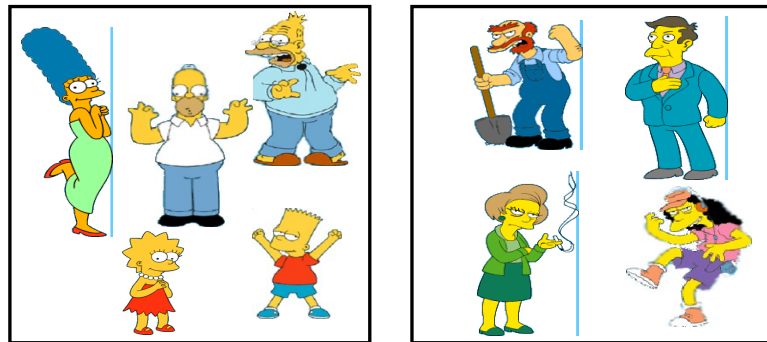
# Relembrando...

- **Agrupamento Particional:** constrói uma *partição* dos dados
- **Agrupamento Hierárquico:** constrói uma *hierarquia de partições*

## Hierárquicos



## Particionais



# Definição de Partição de Dados

- Consideremos um conjunto de  $N$  objetos a serem agrupados:  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$
- **Partição** (rígida): coleção de  $k$  grupos não sobrepostos  $\mathbf{P} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_k\}$  tal que:

$$\mathbf{C}_1 \cup \mathbf{C}_2 \cup \dots \cup \mathbf{C}_k = \mathbf{X}$$

$$\mathbf{C}_i \neq \emptyset$$

$$\mathbf{C}_i \cap \mathbf{C}_j = \emptyset \text{ para } i \neq j$$

- Exemplo:  $\mathbf{P} = \{ (\mathbf{x}_1), (\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6), (\mathbf{x}_2, \mathbf{x}_5) \}$

# Definição de Hierarquia

## ➤ Hierarquia (de partições de dados):

### ➤ Sequência de partições aninhadas

- Uma partição  $\mathbf{P}_1$  está *aninhada* em  $\mathbf{P}_2$  se cada componente (grupo) de  $\mathbf{P}_1$  é um subconjunto de um componente de  $\mathbf{P}_2$

## ➤ Exemplo:

$$\mathbf{P}_1 = \{ (\mathbf{x}_1), (\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6), (\mathbf{x}_2, \mathbf{x}_5) \}$$

$$\mathbf{P}_2 = \{ (\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6), (\mathbf{x}_2, \mathbf{x}_5) \}$$

## ➤ Contra-Exemplo:

$$\mathbf{P}_3 = \{ (\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6), (\mathbf{x}_2, \mathbf{x}_5) \}$$

$$\mathbf{P}_4 = \{ (\mathbf{x}_1, \mathbf{x}_2), (\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6), (\mathbf{x}_5) \}$$

# Definição de Hierarquia

## ➤ Uma hierarquia completa:

- Inicia ou termina com partição totalmente disjunta
  - **Disjoint clustering**: apenas grupos **atômicos** (*singletons*)
  - Exemplo:  $\mathbf{P} = \{ (\mathbf{x}_1), (\mathbf{x}_2), (\mathbf{x}_3), (\mathbf{x}_4), (\mathbf{x}_5), (\mathbf{x}_6) \}$ 
    - Também denominada “solução trivial”
- Inicia ou termina com partição totalmente conjunta
  - **Conjoint clustering**: grupo único com todos os objetos
  - Exemplo:  $\mathbf{P} = \{ (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6) \}$
- Geralmente possui  $N - 2$  partições intermediárias

Hierarquias são  
comumente usadas para  
organizar informação,  
como, por exemplo, num  
portal

Web Site Directory - Sites organized by subject

[Suggest your site](#)

**Business & Economy**

[B2B](#), [Finance](#), [Shopping](#), [Jobs](#)...

**Regional**

[Countries](#), [Regions](#), [US States](#)...

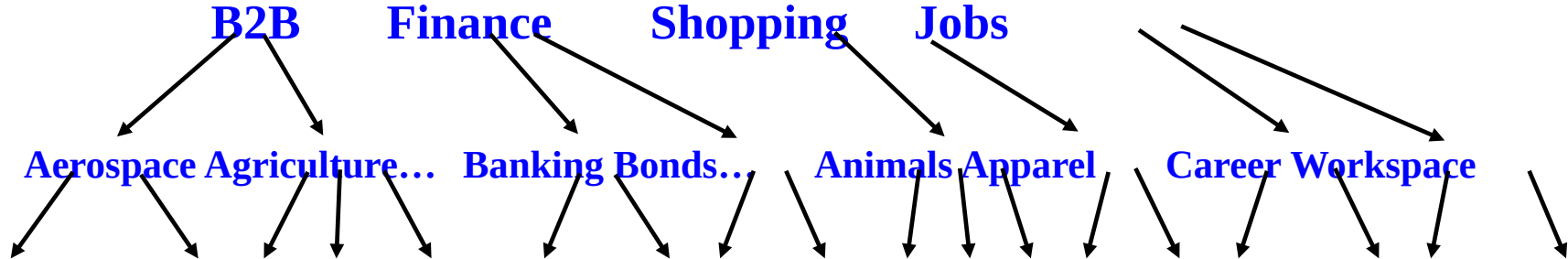
**Computers & Internet**

[Internet](#), [WWW](#), [Software](#), [Games](#)...

**Society & Culture**

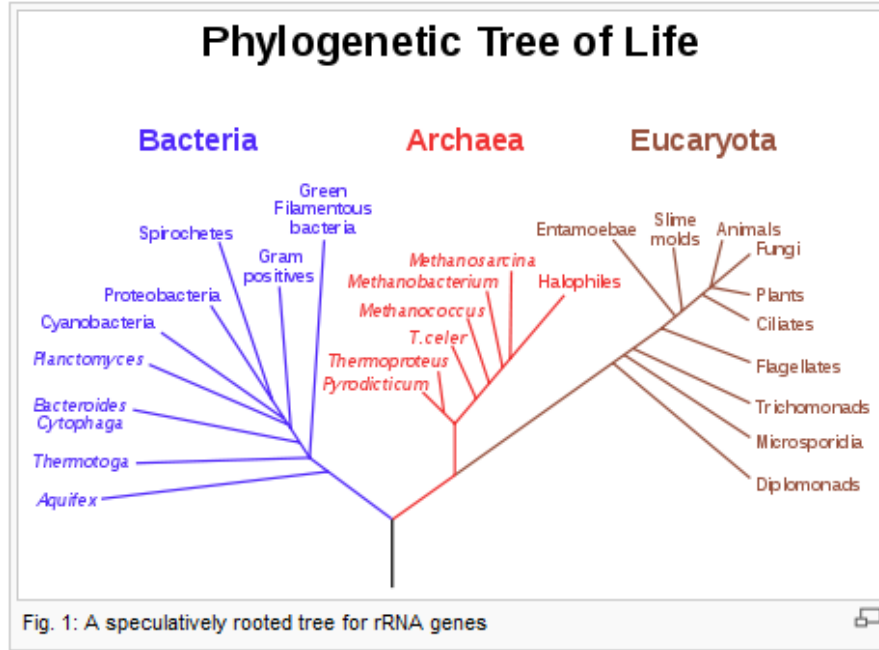
[People](#), [Environment](#), [Religion](#)...

**Business & Economy**



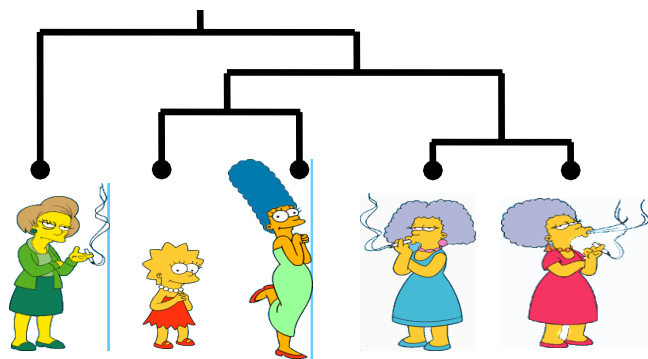
# Outro Exemplo:

## Árvores Filogenéticas em Biologia



[http://en.wikipedia.org/wiki/Phylogenetic\\_tree](http://en.wikipedia.org/wiki/Phylogenetic_tree)

# Métodos Clássicos para Agrupamento Hierárquico



## Bottom-Up (aglomerativos):

- Iniciar colocando cada objeto em um *cluster*
- Encontrar o melhor par de *clusters* para unir
- Unir o par de *clusters* escolhido
- Repetir até que todos os objetos estejam reunidos em um só *cluster*

## Top-Down (divisivos):

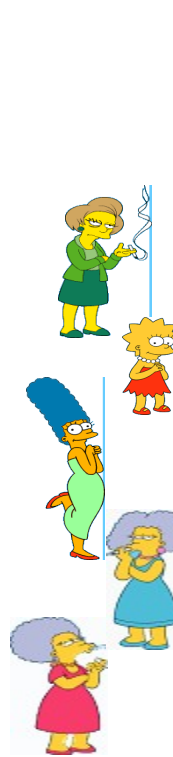
- Iniciar com todos objetos em um único *cluster*
- Sub-dividir o *cluster* em dois novos *clusters*
- Aplicar o algoritmo recursivamente em ambos, até que cada objeto forme um *cluster* por si só



Algoritmos hierárquicos  
podem operar somente sobre  
uma matriz de distâncias: são  
(ou podem ser) **relacionais**.

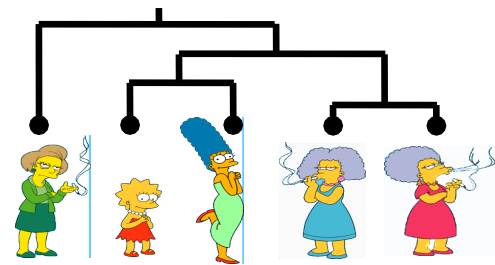
$$D\left(\begin{array}{c} \text{Mrs. Simpson} \\ \text{Lisa Simpson} \end{array}, \begin{array}{c} \text{Lisa Simpson} \end{array}\right) = 8$$

$$D\left(\begin{array}{c} \text{Marge Simpson} \\ \text{Bart Simpson} \end{array}, \begin{array}{c} \text{Bart Simpson} \end{array}\right) = 1$$

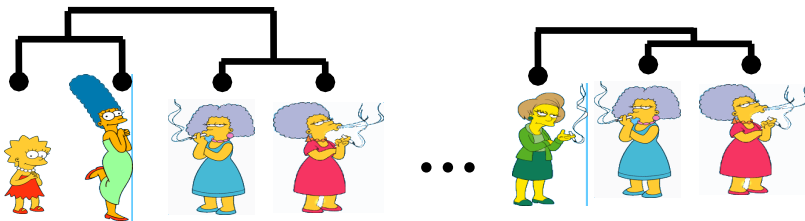


0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

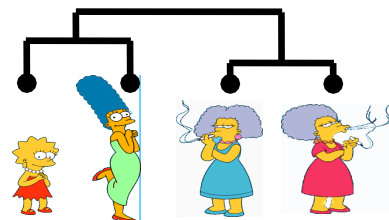
**Bottom-Up (aglomerativo):** Iniciando com cada objeto em seu próprio cluster, encontrar o melhor par de *clusters* para unir em um novo *cluster*. Repetir até que todos os *clusters* sejam fundidos em um único *cluster*.



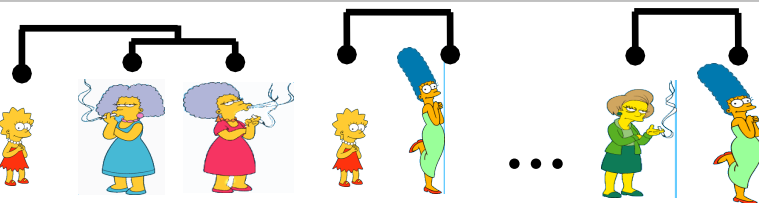
Considerar todas as uniões possíveis ...



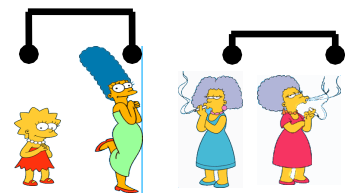
Escolher a melhor



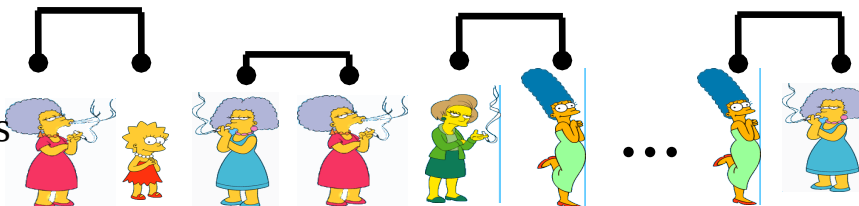
Considerar todas as uniões possíveis ...



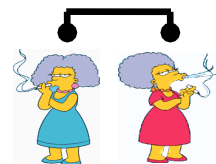
Escolher a melhor



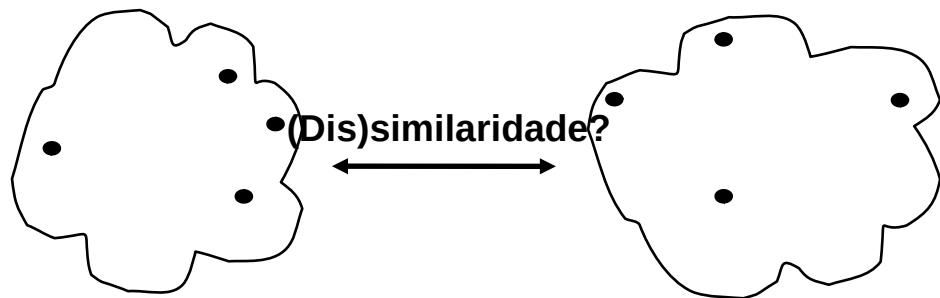
Considerar todas as uniões possíveis ...



Escolher a melhor



# Como definir similaridade entre grupos

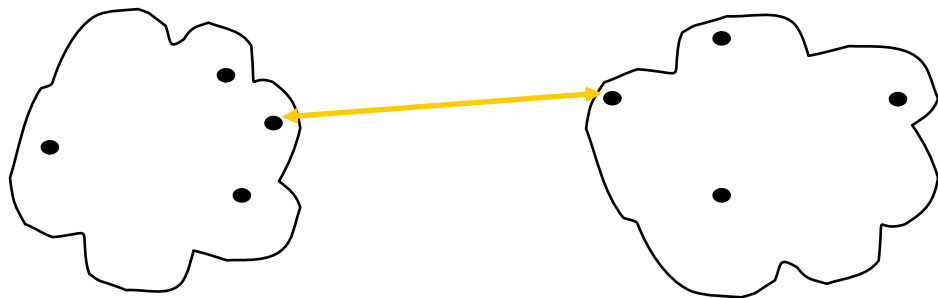


- MIN
- MAX
- Média do grupos
- Distância entre centróides
- Outros métodos
  - Ward's
  - ...

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						

Matriz de  
Proximidades

# Como definir similaridade entre grupos

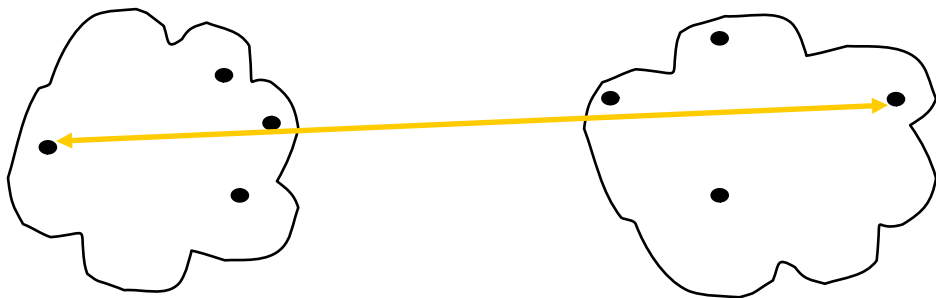


- **MIN**
- **MAX**
- Média do grupos
- Distância entre centróides
- Outros métodos
  - Ward's
  -

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						

Matriz de  
Proximidades

# Como definir similaridade entre grupos

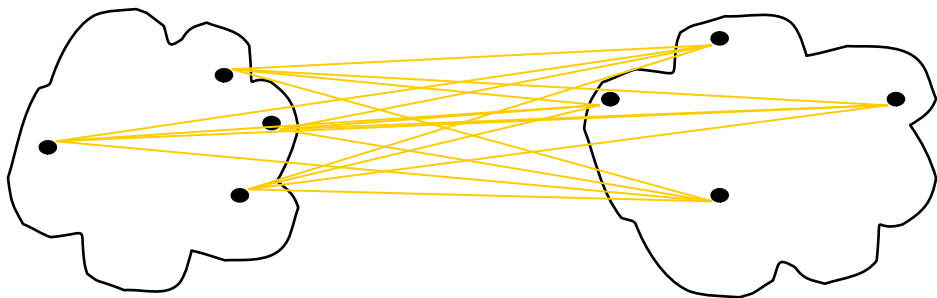


- MIN
- **MAX**
- Média do grupos
- Distância entre centróides
- Outros métodos
  - Ward's
  -

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						

Matriz de  
Proximidades

# Como definir similaridade entre grupos

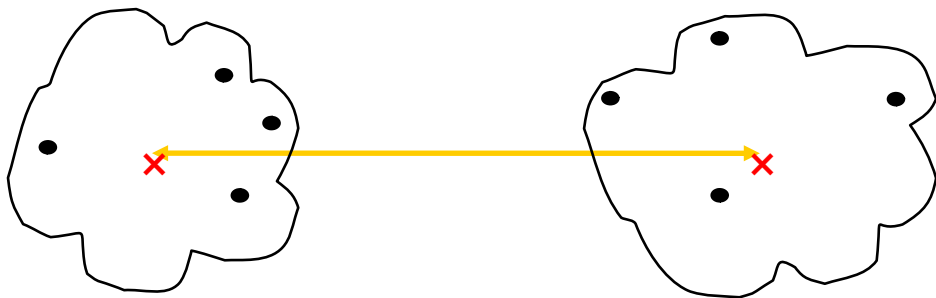


- MIN
- MAX
- **Média do grupos**
- Distância entre centróides
- Outros métodos
  - Ward's
  -

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						

Matriz de  
Proximidades

# Como definir similaridade entre grupos



- MIN
- MAX
- Média do grupos
- **Distância entre centróides**
- Outros métodos
  - Ward's
  -

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						

Matriz de  
Proximidades

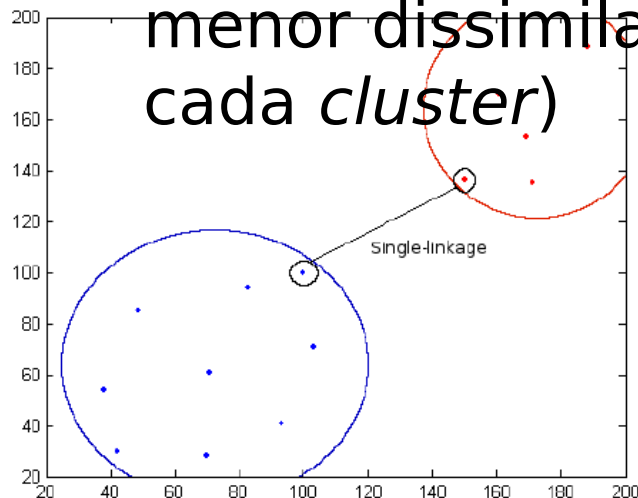
# Como Comparar os Clusters?

***Single Linkage*** , Min, ou Vizinho mais Próximo :

Dissimilaridade entre *clusters* é dada pela menor dissimilaridade entre 2 objetos (um de cada *cluster*)

***single link*** (Florek, 1951; Sneath, 1957)

Originalmente baseado em  
**Grafos**: menor aresta entre dois vértices de subconjuntos distintos





# Propriedade Útil

Propriedade da Função Mínimo (min):

$$\min\{\mathbf{D}\} = \min\{ \min\{\mathbf{D}_1\} , \min\{\mathbf{D}_2\} \}$$

$\mathbf{D}$ ,  $\mathbf{D}_1$  e  $\mathbf{D}_2$  são conjuntos de valores reais tais que  $\mathbf{D}_1 \cup \mathbf{D}_2 = \mathbf{D}$

Exemplo:

$$\min\{10, -3, 0, 100\} = \min \{ \min\{10, -3\}, \min\{0, 100\} \} = -3$$

Propriedade vale recursivamente (para  $\min\{\mathbf{D}_1\}$  e  $\min\{\mathbf{D}_2\}$ )

Utilidade para *Single-Linkage*

Dada a distância entre os grupos  $\mathbf{A}$  e  $\mathbf{B}$  e entre  $\mathbf{A}$  e  $\mathbf{C}$

É trivial calcular a distância entre  $\mathbf{A}$  e  $(\mathbf{B} \cup \mathbf{C})$ .

## Exemplo de Single Linkage: Método de Johnson (1967)

Consideremos a seguinte matriz de distâncias iniciais ( $\mathbf{D}_1$ ) entre 5 objetos  $\{1,2,3,4,5\}$ . Qual par de objetos será escolhido para formar o 1º *cluster* ?

$$D_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & & & & \\ & 2 & & & \\ & 6 & 5 & & \\ & 10 & 9 & 4 & 0 \\ & 9 & 8 & 5 & 3 & 0 \end{bmatrix} \end{matrix}$$

A menor distância entre objetos é  $d_{12}=d_{21}=2$ , indicando que estes dois objetos serão unidos em um *cluster*. Na sequência, calcula-se:

$$d_{(12)3} = \min\{d_{13}, d_{23}\} = d_{23} = 5;$$

$$d_{(12)4} = \min\{d_{14}, d_{24}\} = d_{24} = 9;$$

$$d_{(12)5} = \min\{d_{15}, d_{25}\} = d_{25} = 8;$$

Desta forma, obtém-se uma nova matriz de distâncias ( $\mathbf{D}_2$ ), que será usada na próxima etapa do agrupamento hierárquico:

$$D_2 = \begin{bmatrix} 12 & 0 & & & \\ 3 & 5 & 0 & & \\ 4 & 9 & 4 & 0 & \\ 5 & 8 & 5 & 3 & 0 \end{bmatrix}$$

Qual o novo *cluster* a ser formado?

Unindo os objetos **4** e **5** obtemos três clusters: {1,2}, {4,5}, {3}

Como  $d_{(12)3}$  já está calculada, calculamos na sequência:

$$d_{(12)(45)} = \min\{d_{(12)(4)}, d_{(12)(5)}\} = d_{(12)(5)} = 8$$

$$d_{(45)3} = \min\{d_{43}, d_{53}\} = d_{43} = 4$$

obtendo a seguinte matriz:

$$D_3 = \begin{matrix} & 12 & \begin{bmatrix} 0 \\ 5 \\ 8 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{matrix} 3 \\ 45 \end{matrix} & \begin{bmatrix} 3 \\ 8 \end{bmatrix} & \begin{bmatrix} 5 \\ 4 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{matrix}$$

\* Unir *cluster* {3} com {4,5};

\* Finalmente, unir todos os *clusters* em um único *cluster*

A sequência de partições obtidas neste exemplo é, portanto:

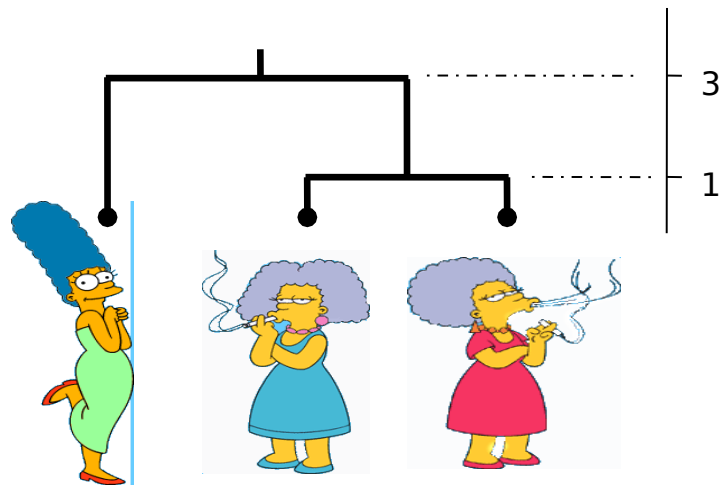
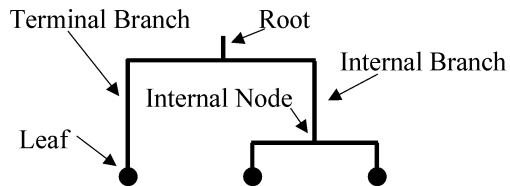
$$\{ (1), (2), (3), (4), (5) \} \rightarrow \{ (1, 2), (3), (4), (5) \} \rightarrow$$
$$\{ (1, 2), (3), (4, 5) \} \rightarrow \{ (1, 2), (3, 4, 5) \} \rightarrow \{ (1, 2, 3, 4, 5) \}$$

**Nota:** Para *single link*, a dissimilaridade entre 2 clusters pode ser computada naturalmente a partir da matriz atualizada na iteração anterior, sem necessidade da matriz original

Isso vale devido à propriedade da função *min* vista anteriormente

# Dendrograma

**Dendrograma:** Hierarquia + Dissimilaridades entre Clusters

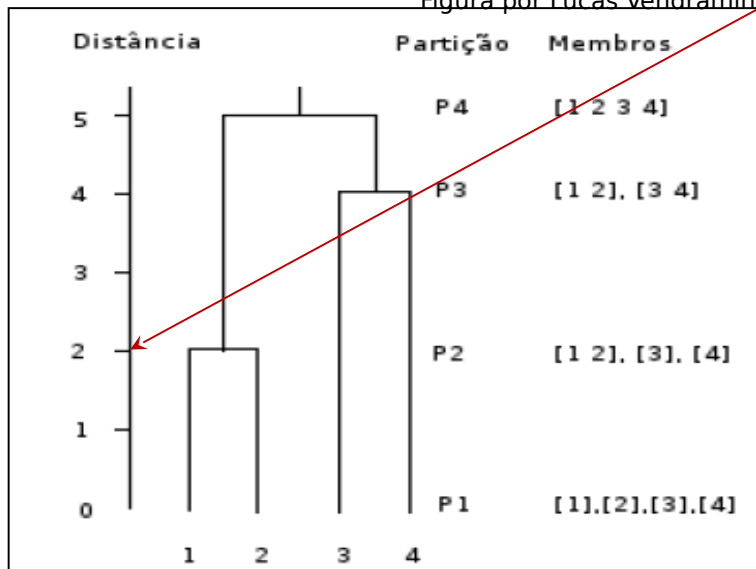


\* A dissimilaridade entre dois clusters (possivelmente **singletons**) é representada como a altura do nó interno mais baixo compartilhado

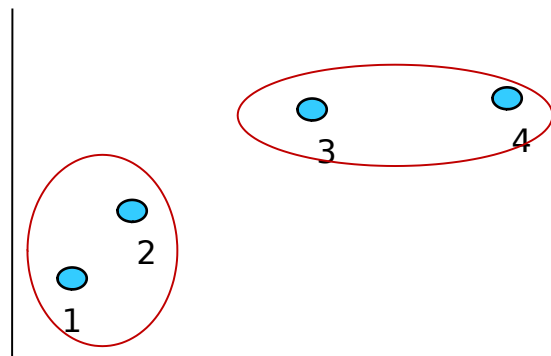
# Exemplo de Dendrograma

$$D = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 2 & 7 & 13 \\ 2 & 0 & 5 & 10 \\ 7 & 5 & 0 & 4 \\ 13 & 10 & 4 & 0 \end{bmatrix} \end{matrix}$$

Figura por Lucas Vendramin

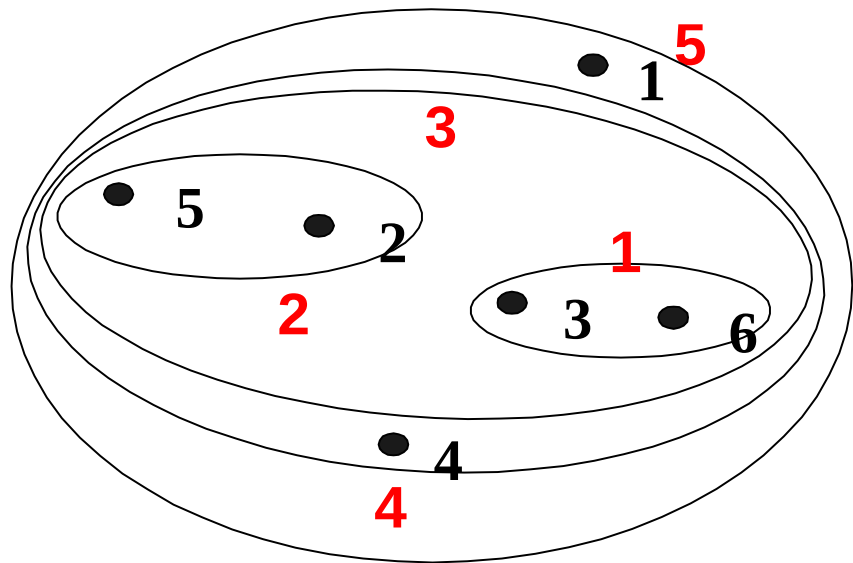


Dendrograma

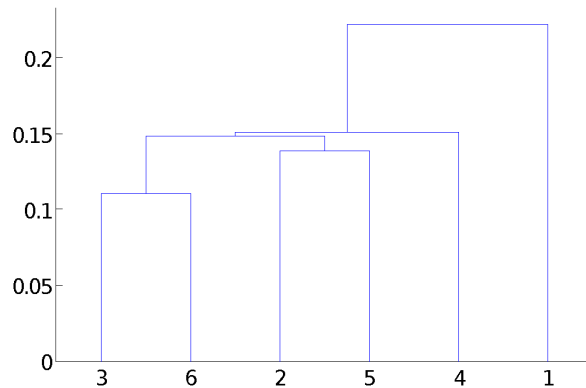


uma das partições  
aninhadas

# Outro Exemplo de Dendrograma



*Clusters Aninhados*

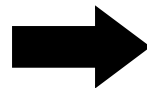
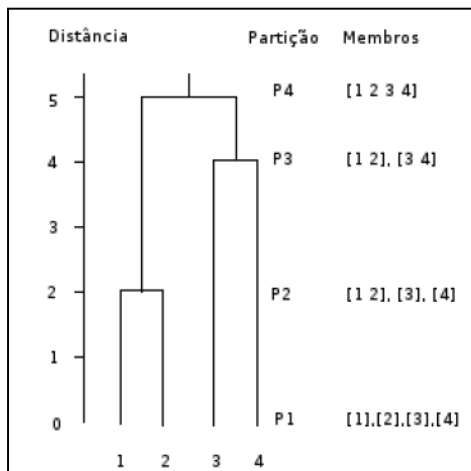


*Dendrograma*



# Cophenetic Matrix

Matriz com as dissimilaridades que levaram à união de cada par de objetos na base de dados. Exemplo:



$$C_p = \begin{bmatrix} 0 & 2 & 5 & 5 \\ 2 & 0 & 5 & 5 \\ 5 & 5 & 0 & 4 \\ 5 & 5 & 4 & 0 \end{bmatrix}$$

Esta matriz é importante para a validação de agrupamentos hierárquicos

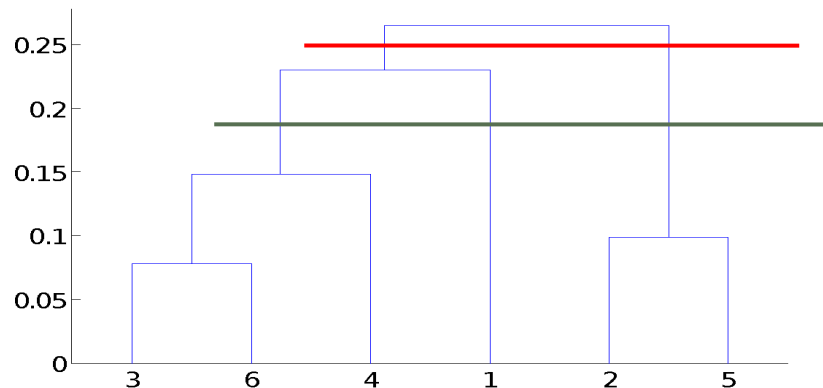
# Dendrogramas e Partições

- Partições são obtidas via **cortes** no dendrograma
  - cortes horizontais
  - no. de grupos da partição = no. de interseções

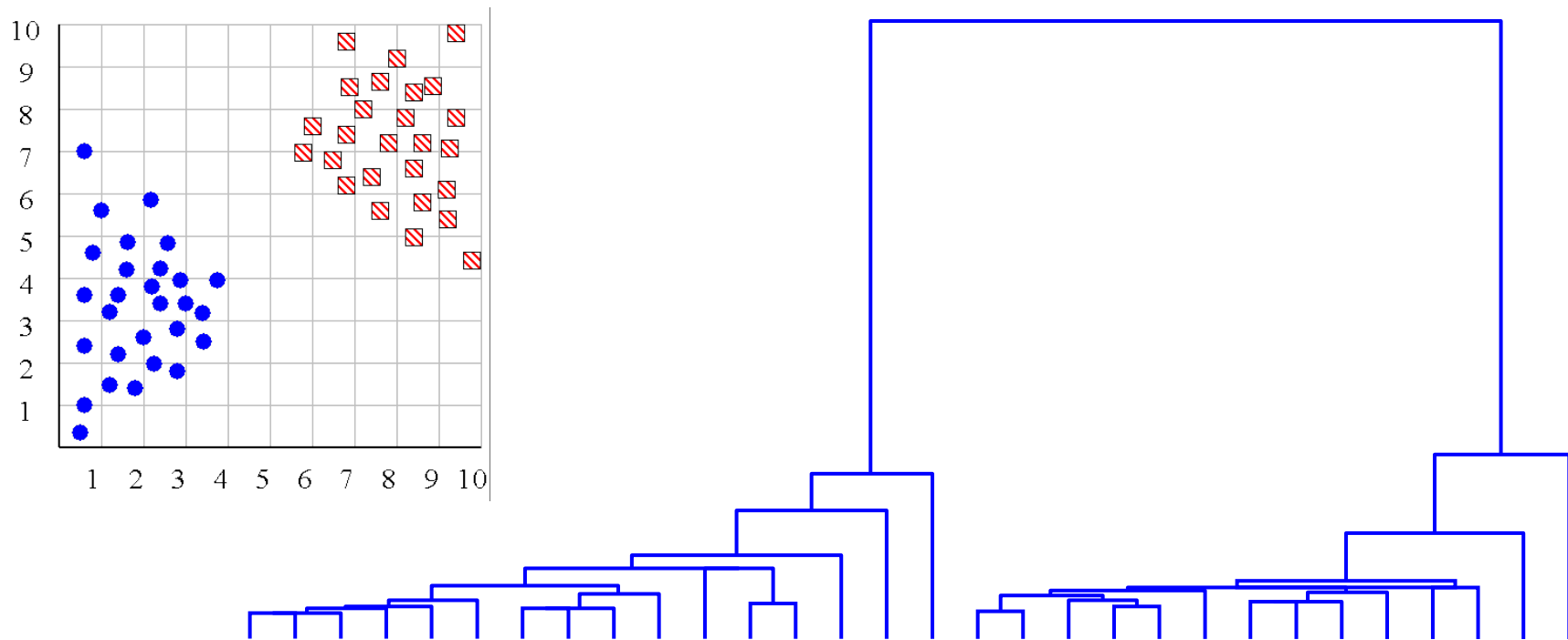
## ➤ Exemplos:

$$P_2 = \{ (\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6), (\mathbf{x}_2, \mathbf{x}_5) \}$$

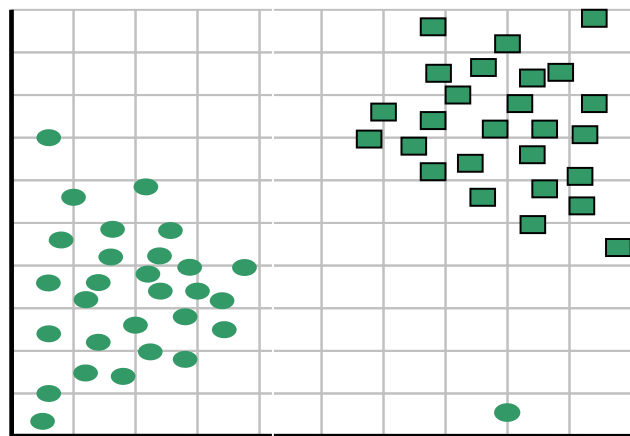
$$P_1 = \{ (\mathbf{x}_1), (\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6), (\mathbf{x}_2, \mathbf{x}_5) \}$$



Pode-se examinar o dendrograma para tentar estimar o número *mais natural* de clusters. No caso abaixo, existem duas sub-árvores bem separadas, sugerindo dois grupos de dados. Infelizmente, na prática, as distinções não são tão simples...

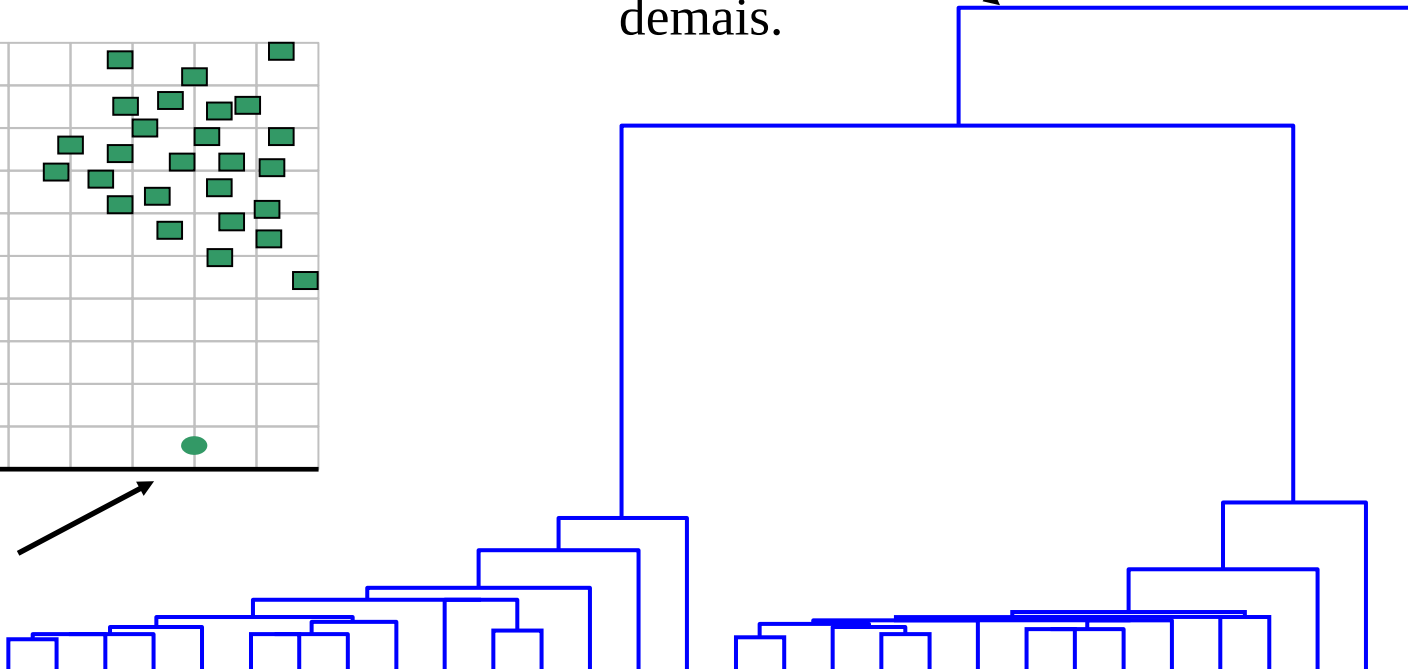


# Pode-se usar o dendrograma para tentar detectar *outliers*:



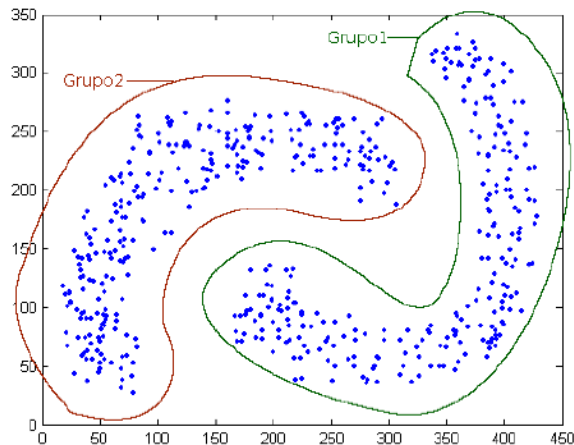
*Outlier*

Ramo isolado sugere que o objeto é muito diferente dos demais.



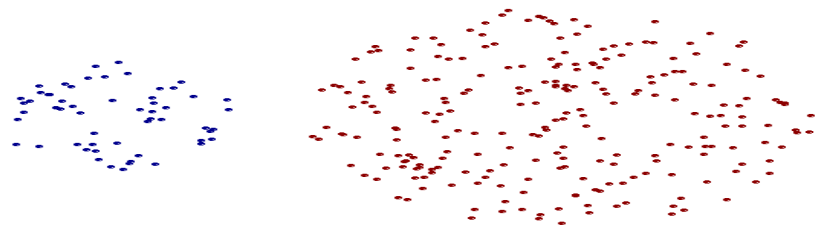
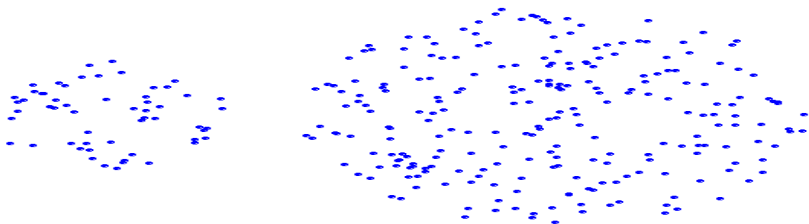
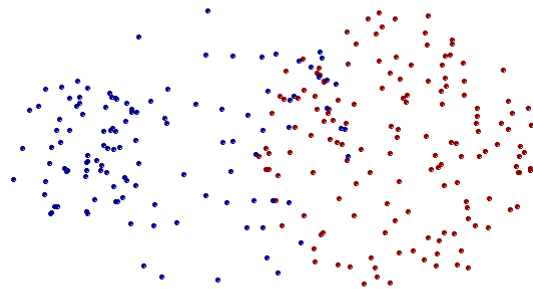
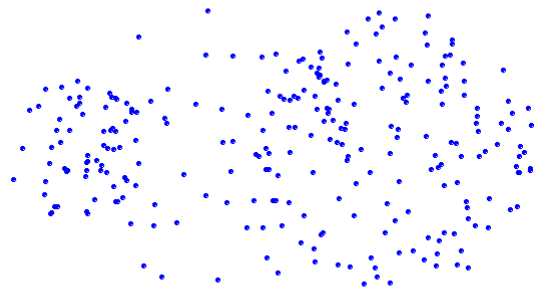
# Vantagens do MIN

- Capacidade de lidar com formas não globulares



# Limitações do MIN

- Sensibilidade a ruídos e outliers



Original Points

Two Clusters

# Como Comparar os Clusters?

***Complete Linkage*** , Max, ou Vizinho mais Distante:

Dissimilaridade entre *clusters* é dada pela maior dissimilaridade entre dois objetos (um de cada *cluster*)



***complete link*** (Sorensen, 1948)

Originalmente baseado em  
**Grafos**: maior aresta entre  
dois vértices de subconjuntos  
distintos

# Propriedade Útil

Propriedade da Função Máximo (max):

$$\max\{\mathbf{D}\} = \max\{ \max\{\mathbf{D}_1\} , \max\{\mathbf{D}_2\} \}$$

$\mathbf{D}$ ,  $\mathbf{D}_1$  e  $\mathbf{D}_2$  são conjuntos de valores reais tais que  $\mathbf{D}_1 \cup \mathbf{D}_2 = \mathbf{D}$

Exemplo:

$$\max\{10, -3, 0, 100\} = \max \{ \max\{10, -3\}, \max\{0, 100\} \} = 100$$

Propriedade vale recursivamente (para  $\max\{\mathbf{D}_1\}$  e  $\max\{\mathbf{D}_2\}$ )

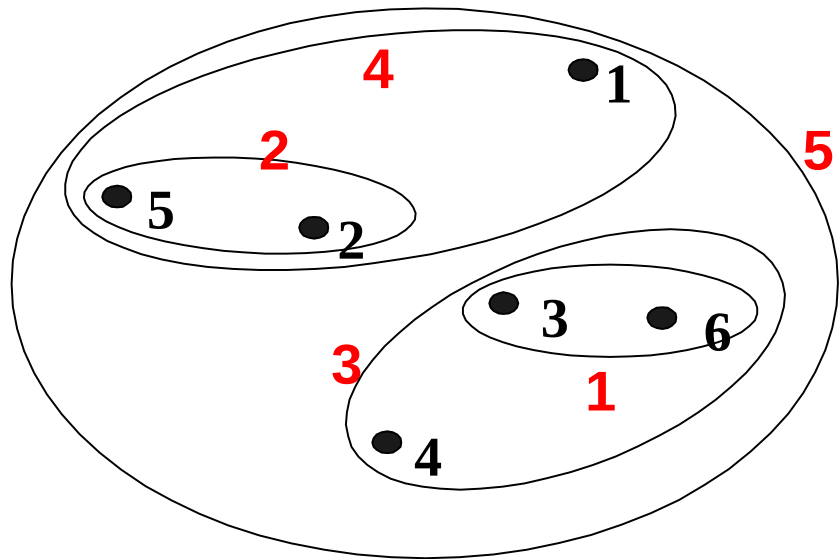
Utilidade para *Complete-Linkage*

Dada a distância entre os grupos  $\mathbf{A}$  e  $\mathbf{B}$  e entre  $\mathbf{A}$  e  $\mathbf{C}$

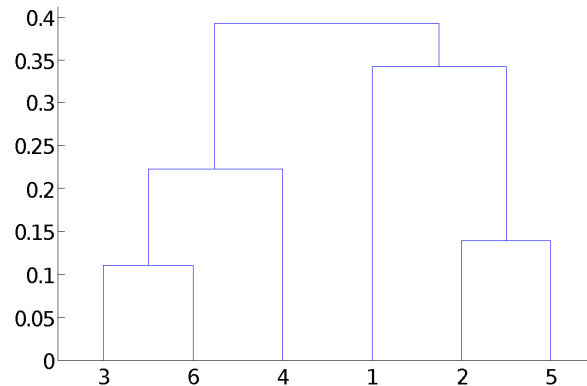
É trivial calcular a distância entre  $\mathbf{A}$  e  $(\mathbf{B} \cup \mathbf{C})$ .



# Agrupamento Hierárquico: MAX

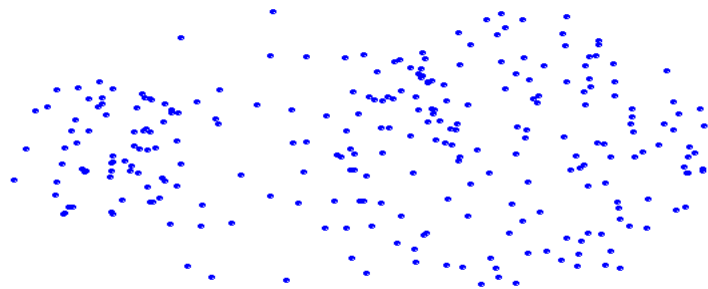


**Nested Clusters**

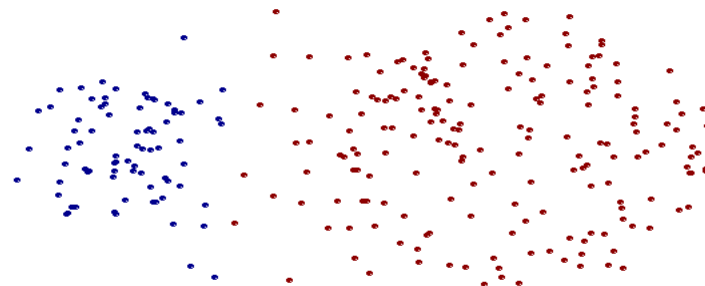


**Dendrogram**

# Vantagens do MAX



Pontos Originais

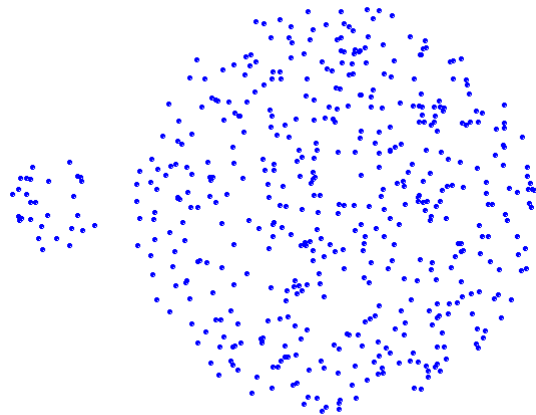


Dois Grupos

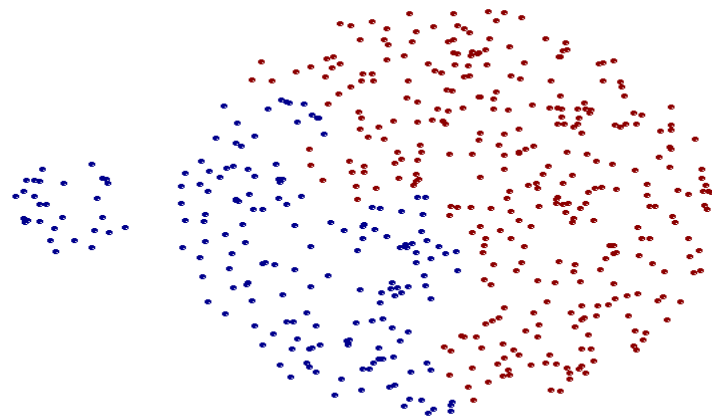
- Menos suscetível a ruído e *outliers*

# Limitações do MAX

Pontos Originais



Dois Grupos

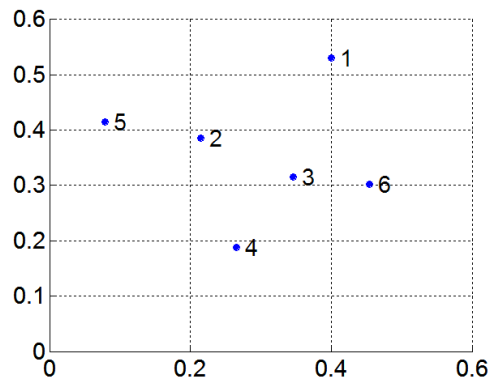


- Tendência a quebrar grupos grandes
- Enviesado para grupos globulares

# Group Average

- Dissimilaridade entre dois *clusters* é a média das dissimilaridades entre os objetos dos dois *clusters*

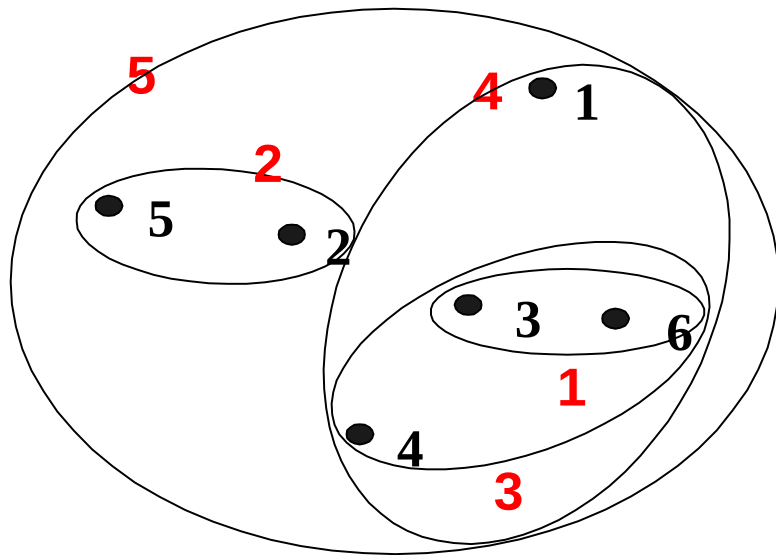
$$d(G_i, G_j) = \frac{\sum_{\mathbf{x}_n \in G_i} \sum_{\mathbf{x}_m \in G_j} d(\mathbf{x}_n, \mathbf{x}_m)}{|G_i| \times |G_j|}$$



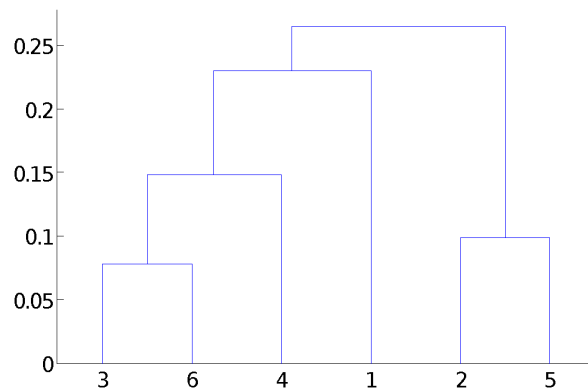
**Matriz de distância**

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

# Agrupamento hierárquico: Group Average



**Nested Clusters**



**Dendrogram**

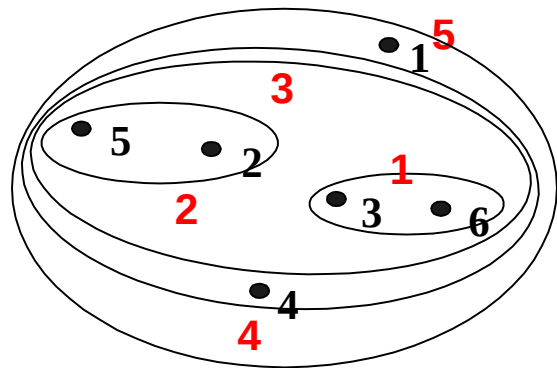
# Agrupamento hierárquico: Group Average

- Meio-termo entre Single e Complete Link
- Menos suscetível à *outliers* e ruído
- Enviesado para *clusters* globulares

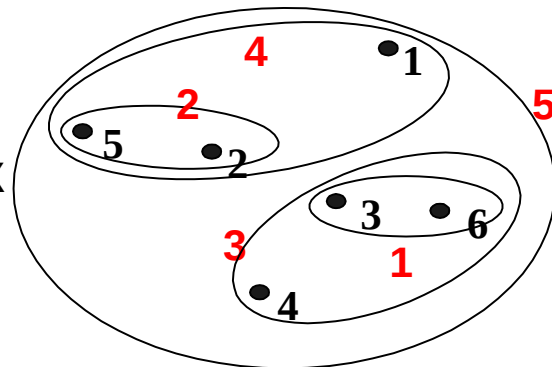
# Método de Ward

- Distância entre dois grupos baseado no aumento no erro quadrático quando os grupos são unidos
- Menos suscetível à ruídos e *outliers*
- Enviesado para *clusters* globulares
- Similar ao *k-means*, mas hierárquico
  - Pode ser usado para iniciar o *k-means*

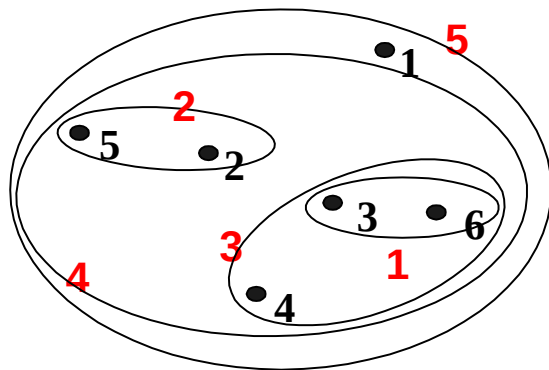
# Comparação entre os métodos



MIN

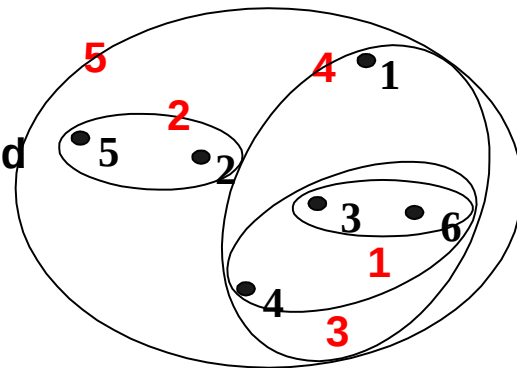


MAX



Group Average

Ward's Method





# Generalizando: Esquema Lance-Williams

Todos os algoritmos que vimos são instâncias de um modelo geral:

$d_{ik}$  = distância entre  $G_i$  e  $G_k$

$d_{(ij)k}$  = distância entre os grupos  $G_i$  e  $G_j$  e  $G_k$

$$d_{(ij)k} = \alpha_i d_{ik} + \alpha_j d_{jk} + \beta d_{ij} + \gamma |d_{ik} - d_{jk}|$$

Para single-linkage:  $\alpha_i = \alpha_j = 0,5$  e  $\gamma = -0,5$

# Referências

- Tan, P.-N., Steinbach, M., and Kumar, V., *Introduction to Data Mining*, **Capítulo 8**. Addison-Wesley, 2006
- Jain, A. K. and Dubes, R. C., *Algorithms for Clustering Data*, Prentice Hall, 1988
- Everitt, B. S., Landau, S., and Leese, M., *Cluster Analysis*, Arnold, 4<sup>th</sup> Edition, 2001.