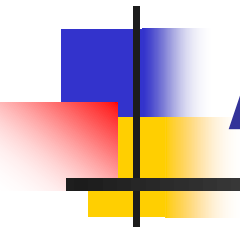
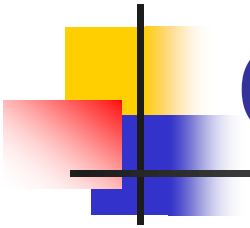


# SCC5895 – Análise de Agrupamento de Dados

## Algoritmos Hierárquicos: Parte II

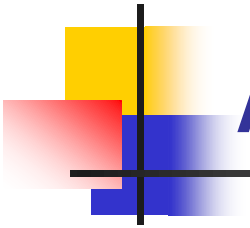
**Prof. Eduardo Raul Hruschka**

PPG-CCMC / ICMC / USP



# Créditos

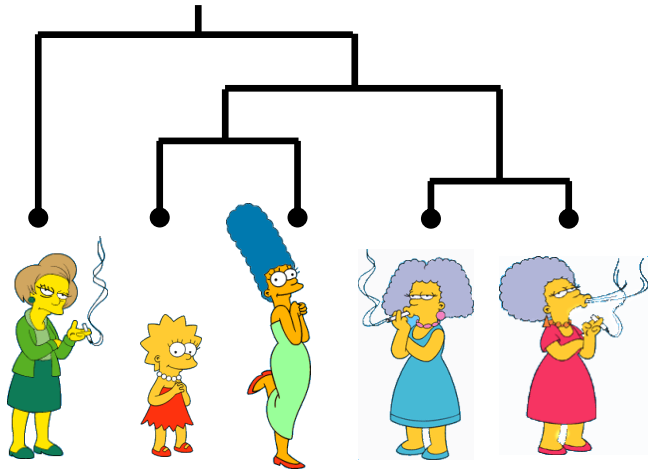
- Parte do material a seguir consiste de adaptações e extensões dos originais:
  - Elaborados por Eduardo R. Hruschka e Ricardo J. G. B. Campello
  - de (Tan et al., 2006)
  - de E. Keogh (SBBD 2003)
- Algumas figuras foram gentilmente cedidas por Lucas Vendramin



# Agenda

- Continuação de Algoritmos Hierárquicos
  - Average Linkage (UPGMA)
    - Variantes: WPGMA, UPGMC, WPGMC
  - Método de Ward
  - Esquema de Lance-Williams
  - Métodos Monótonos e Não Monótonos
  - Métodos Divisivos
    - Heurística de MacNaughton-Smith (DIANA)
    - Bisecting k-Means
    - Single Linkage via Árvores Geradoras Mínimas em Grafos
  - Complexidade Computacional

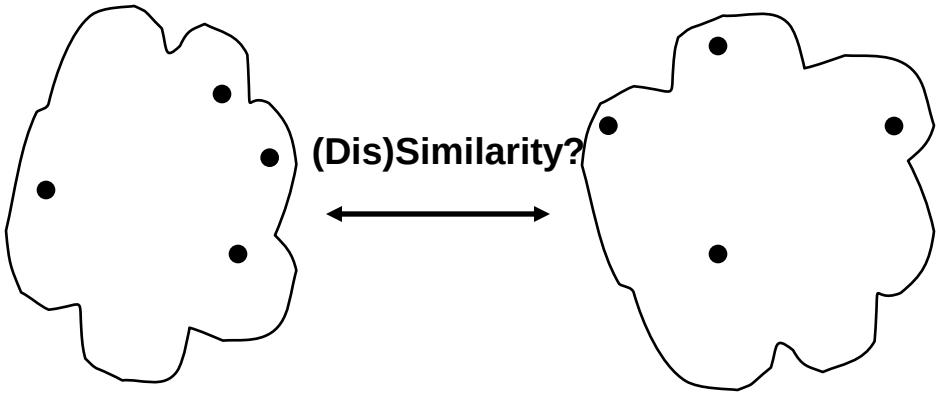
# Relembrando Agrupamento Hierárquico...



**Bottom-Up (métodos 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 (métodos 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

# How to Define Inter-Cluster (Dis)Similarity




- MIN (aula anterior)
- MAX (aula anterior)
- **Group Average**
- **Distance Between Centroids**
- **Other methods**
  - **Ward's**
  - **…**

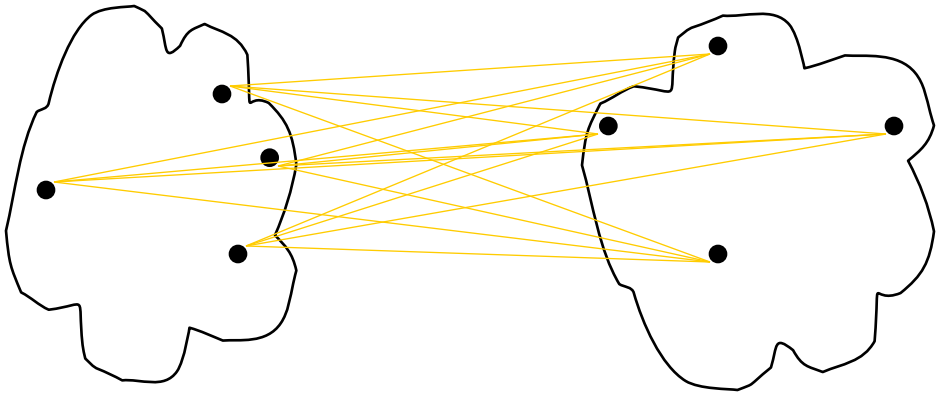
| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |

.

.

**Proximity Matrix**

# How to Define Inter-Cluster (Dis)Similarity



| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |

.

.

**Proximity Matrix**

- MIN
- MAX
- **Group Average**
- Distance Between Centroids
- Other methods
  - Ward's
  - …

# *Average Linkage* ou *Group Average*

- Distância entre *clusters* é dada pela distância média entre cada par de objetos (um de cada *cluster*)
- Também conhecido como **UPGMA** :
  - *Unweighted Pair Group Method using Arithmetic averages*
    - "unweighted" → cada par de objetos possui a mesma importância

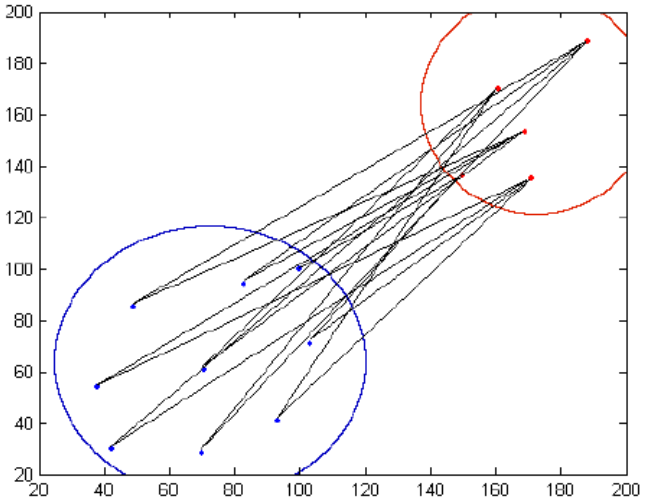



Figura por Lucas Vendramin

# Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

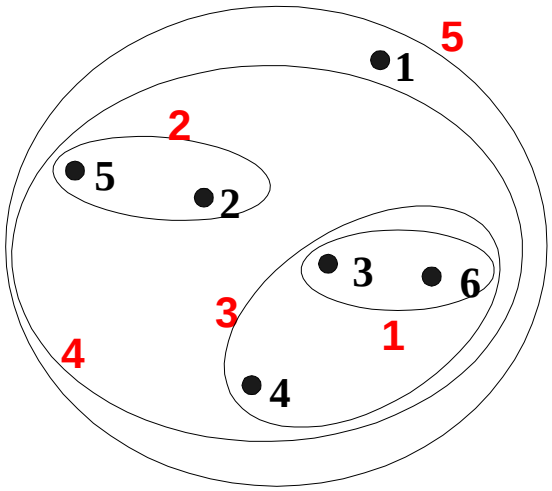
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

- Need to use average connectivity for scalability since total proximity favors large clusters

| | I1 | I2 | I3 | I4 | I5 |
|---|---|---|---|---|---|
| I1 | 1.00 | 0.90 | 0.10 | 0.65 | 0.20 |
| I2 | 0.90 | 1.00 | 0.70 | 0.60 | 0.50 |
| I3 | 0.10 | 0.70 | 1.00 | 0.40 | 0.30 |
| I4 | 0.65 | 0.60 | 0.40 | 1.00 | 0.80 |
| I5 | 0.20 | 0.50 | 0.30 | 0.80 | 1.00 |

**Exercício: Gerar a hierarquia.**

# Hierarchical Clustering: Group Average




Nested Clusters

**Exercício:** atribua valores de distância entre os pontos ao lado, que sejam condizentes com a figura, e monte o dendrograma.

# Hierarchical Clustering: Group Average

- **Group Average** represents a compromise between Single and Complete Link

- **Strengths**
  - Less susceptible to noise and outliers

- **Limitations**
  - Biased towards globular clusters

# Como Comparar os Clusters ?

- **Single** x **Complete** x **Average**:
  - Single Linkage:
    - Capaz de detectar clusters de formas complexas
    - No entanto, muito sensível a ruído nos dados (e.g. "pontes")

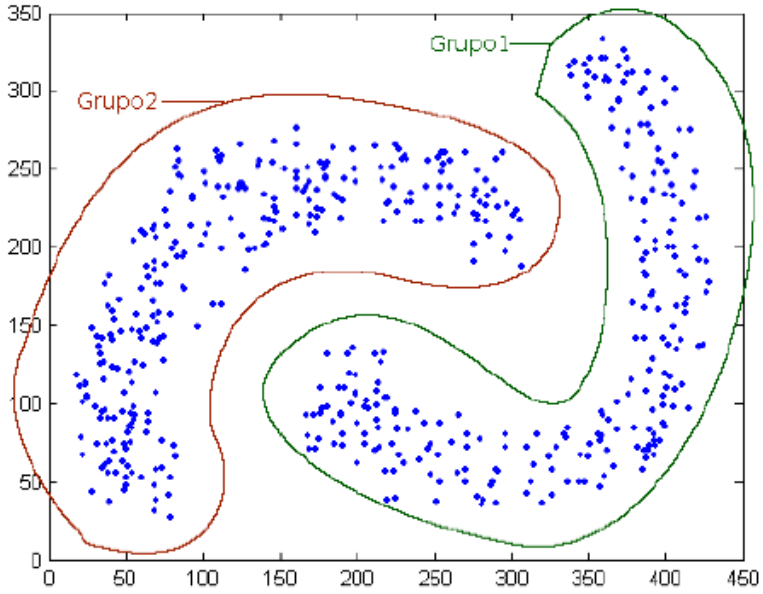


Figura por Lucas Vendramin

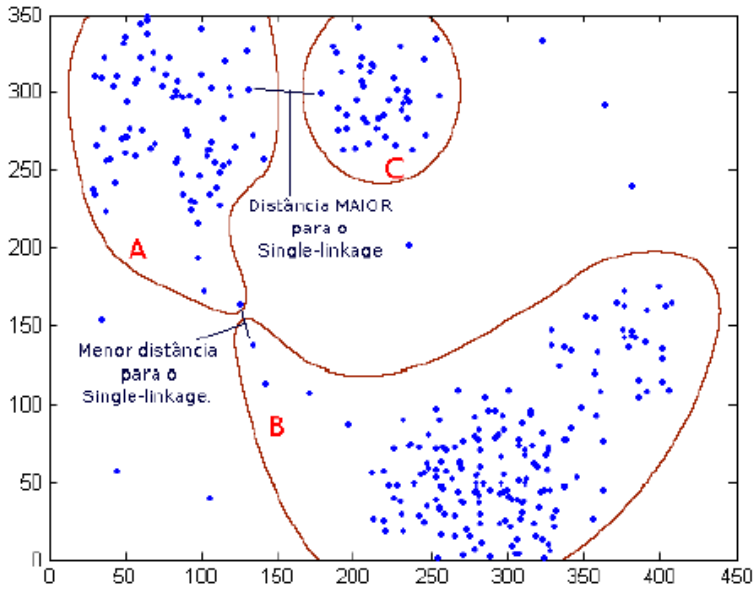


Figura por Lucas Vendramin

# Como Comparar os Clusters ?

- **Single** x **Complete** x **Average**:
  - Complete Linkage:
    - Reduz sensibilidade a ruído (e.g. pontes entre clusters)
    - No entanto:
      - aumenta risco de separar clusters grandes
      - perde capacidade de detecção de formas complexas
        - favorece clusters globulares
  - Average Linkage:
    - Também favorece clusters bem comportados (globulares)
    - Mas é muito menos sensível (mais robusto) a ruído e outliers.

# Atualização da Matriz de Proximidades

- Para fins de atualização da matriz de (dis)similaridade em **average linkage**, o cálculo da (dis)similaridade entre um novo cluster (dado pela união de outros dois) e os demais deve considerar o no. de objetos em cada cluster envolvido
  - já que average linkage calcula uma média.
- Especificamente, sendo $|\mathbf{C}_i|$ o número de objetos em um cluster $\mathbf{C}_i$ e $d(\mathbf{C}_i,\mathbf{C}_j)$ a (dis)similaridade entre dois clusters $\mathbf{C}_i$ e $\mathbf{C}_j$, é simples mostrar que (vide Lance-Williams / exercícios):

$$d\left(\mathbf{C}_i, \mathbf{C}_j \cup \mathbf{C}_k\right) = \frac{\left|\mathbf{C}_j\right|}{\left|\mathbf{C}_j\right| + \left|\mathbf{C}_k\right|} d\left(\mathbf{C}_i, \mathbf{C}_j\right) + \frac{\left|\mathbf{C}_k\right|}{\left|\mathbf{C}_j\right| + \left|\mathbf{C}_k\right|} d\left(\mathbf{C}_i, \mathbf{C}_k\right)$$

# Exercício:

- Obtenha o dendrograma completo de execução do método average linkage (**UPGMA**) sobre a matriz de distâncias abaixo
  - Mostre passo a passo a matriz atualizada (via fórmula do slide anterior)

$$\mathbf{D}_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 5 & 0 & & \\ 10 & 9 & 4 & 0 & \\ 9 & 8 & 5 & 3 & 0 \end{bmatrix} \end{array}$$

- Apresente também a *cophenetic matrix* correspondente

# Variante de Average Linkage

- Método **WPGMA**:
  - *Weighted Pair Group Method using Arithmetic averages*

$$d\left(\mathbf{C}_i, \mathbf{C}_j \cup \mathbf{C}_k\right) = \frac{d\left(\mathbf{C}_i, \mathbf{C}_j\right) + d\left(\mathbf{C}_i, \mathbf{C}_k\right)}{2}$$

    - Média aritmética simples das (dis)similaridades entre os grupos
      - não leva em conta as cardinalidades (no. objetos) dos grupos
    - Mesma importância aos grupos, independente dos tamanhos
      - equivale a dar maior importância (**peso**) às (dis)similaridades envolvendo os objetos do grupo de menor tamanho ($\mathbf{C}_j$ ou $\mathbf{C}_k$)
        - reduz o peso dos objetos do grupo de maior tamanho

# Método de Ward (1963)

- Método baseado na minimização do **Critério de Erro Quadrático** (variâncias intra-grupos) a cada nova partição:

$$J = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in \mathbf{C}_i} d\left(\mathbf{x}_j, \bar{\mathbf{x}}_i\right)^2$$

onde $d$ = Euclidiana e $\bar{\mathbf{x}}_i$ é o centróide do i-ésimo cluster:

$$\bar{\mathbf{x}}_i = \frac{1}{\left|\mathbf{C}_i\right|} \sum_{\mathbf{x}_i \in \mathbf{C}_i} \mathbf{x}_i$$

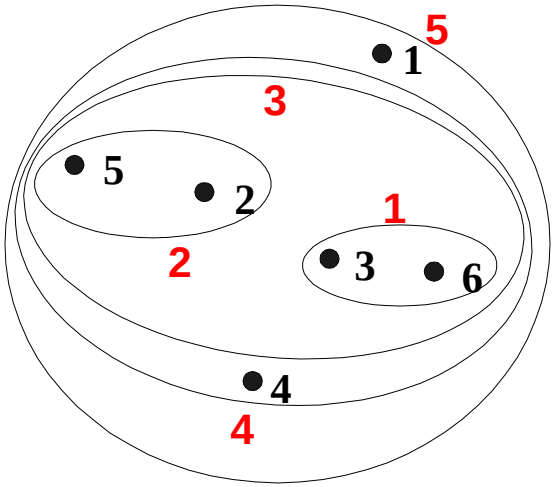
# Método de Ward

- "Dissimilaridade" entre cada par de grupos $\mathbf{C}_i$ e $\mathbf{C}_j$
  - definida como a variação no critério *J* da partição corrente se esses grupos forem unidos para formar a partição seguinte na sucessão hierárquica
  - unir os 2 grupos mais similares significa minimizar o crescimento das variâncias intra-grupos a cada nível da hierarquia
- Vide fórmula de atualização no esquema Lance-Williams...

# Método de Ward

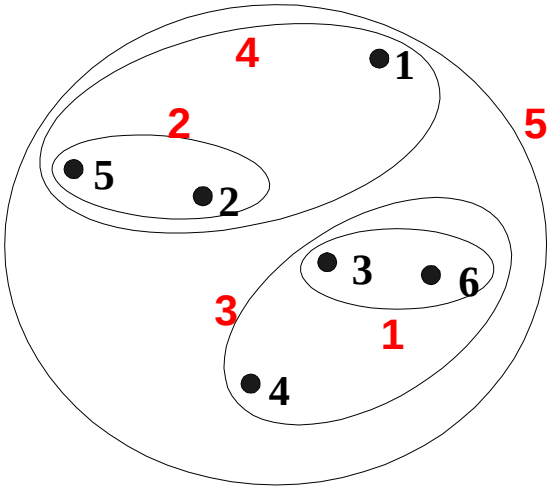
- **Limitações**:
  - Assim como Average Linkage, tende a gerar clusters globulares
  - Fórmula de atualização só possui interpretação para dados descritos por atributos numéricos e distância Euclidiana
- **Vantagens**:
  - Similar a Average Linkage em robustez a ruído e outliers
  - "Análogo hierárquico" do k-means (mesma função objetivo)
    - pode ser usado para inicializar k-means
  - **Jain & Dubes (1988):** "*Several of the comparative studies discussed in Section 3.5.2 conclude that Ward's method, also called the minimum variance method, outperforms other hierarchical clustering methods*"

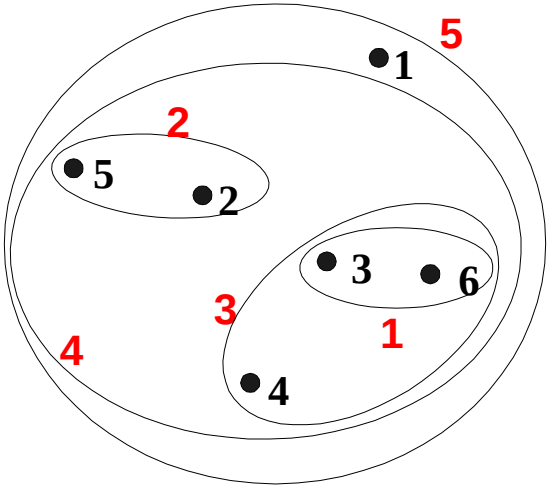
# Hierarchical Clustering: Comparison




MIN

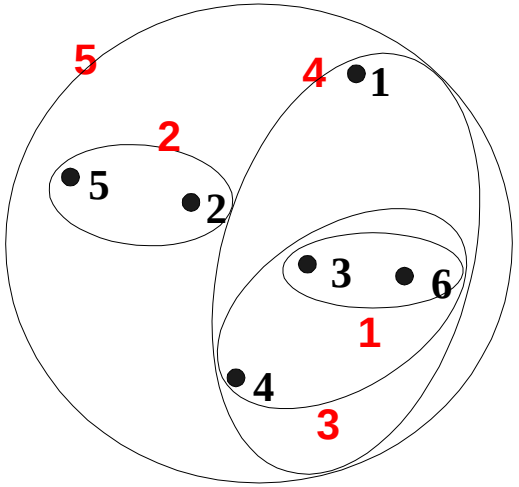
MAX

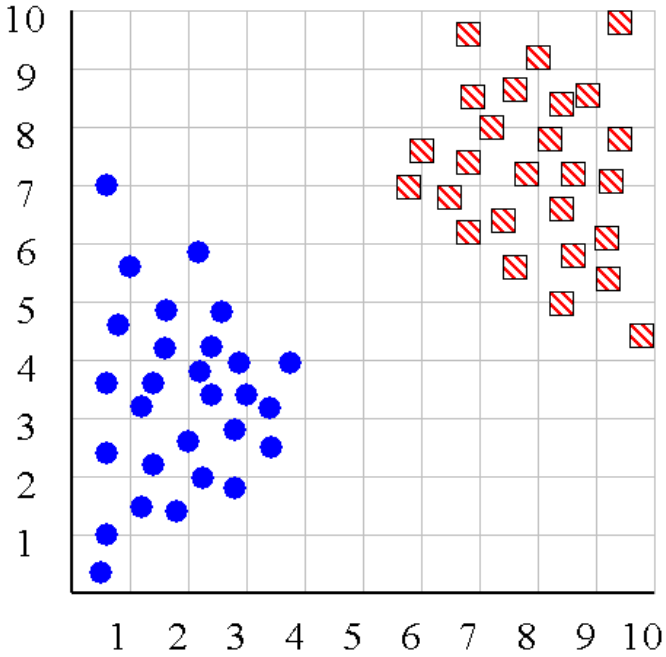


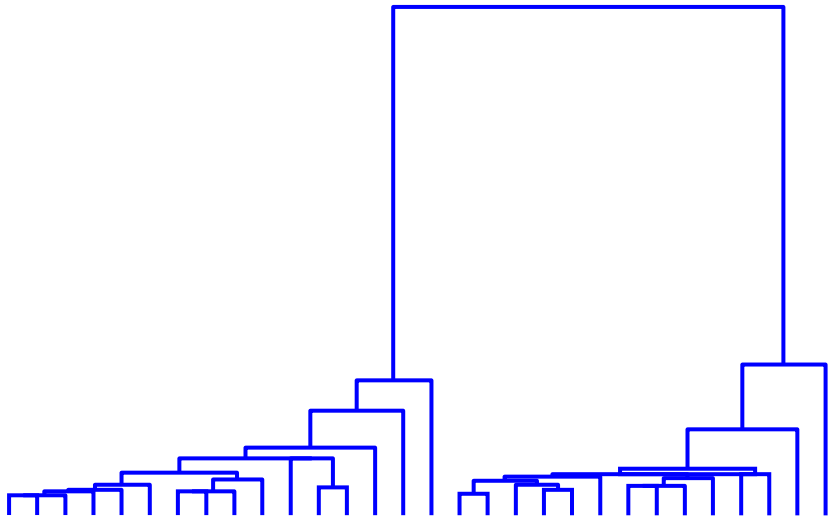




Group Average

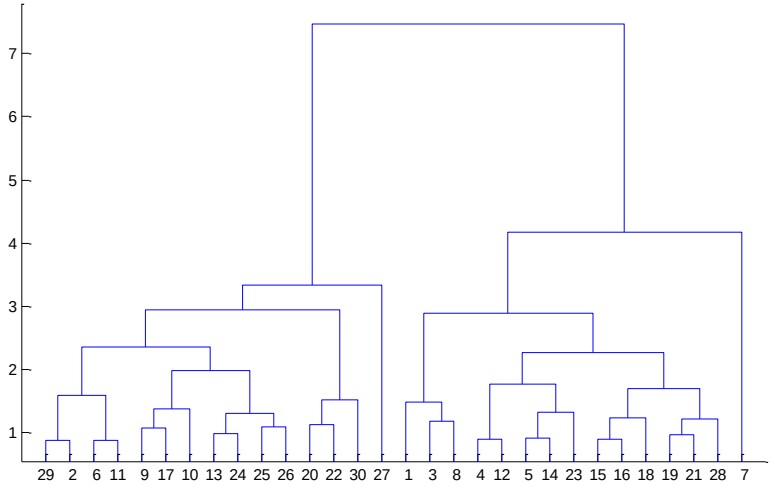
Ward's Method



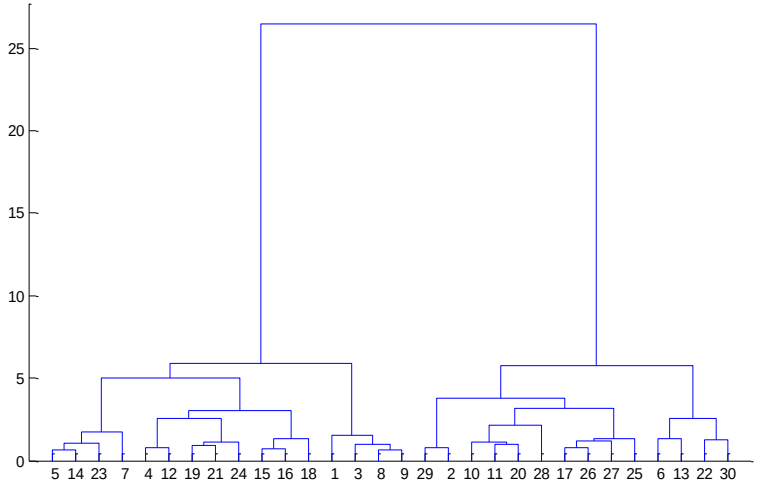


Vinculação Simples *(Single Linkage)*




Vinculação Média (*Average Linkage)*




Vinculação de *Ward*

# Esquema de Lance-Williams (1967)

- **Formulação Parametrizada** que abrange todos os métodos vistos anteriormente

$$d(\mathbf{C}_i, \mathbf{C}_j \cup \mathbf{C}_k) = \alpha_j d(\mathbf{C}_i, \mathbf{C}_j) + \alpha_k d(\mathbf{C}_i, \mathbf{C}_k) + \beta d(\mathbf{C}_j, \mathbf{C}_k) + \gamma \left| d(\mathbf{C}_i, \mathbf{C}_j) - d(\mathbf{C}_i, \mathbf{C}_k) \right|$$

onde $|\cdot|$ significa valor absoluto

- Algoritmo aglomerativo unificado
  - atualização configurável da matriz de proximidades

# Esquema de Lance-Williams (1967)

| | $\alpha_j$ | $\alpha_k$ | $\beta$ | $\gamma$ |
|---|---|---|---|---|
| **Single-Linkage** | 1/2 | 1/2 | 0 | $-1/2$ |
| **Complete-Linkage** | 1/2 | 1/2 | 0 | 1/2 |
| **UPGMA** | $N_j / (N_j + N_k)$ | $N_k / (N_j + N_k)$ | 0 | 0 |
| **WPGMA** | 1/2 | 1/2 | 0 | 0 |
| **UPGMC** | $N_j / (N_j + N_k)$ | $N_k / (N_j + N_k)$ | $-(N_j N_k) / (N_j + N_k)^2$ | 0 |
| **WPGMC** | 1/2 | 1/2 | $-1/4$ | 0 |
| **Ward's** | $(N_j+N_i) / (N_j+N_k+N_i)$ | $(N_k+N_i) / (N_j+N_k+N_i)$ | $-N_i / (N_j+N_k+N_i)$ | 0 |

- NOTAS:
  - $N_i$, $N_j$ e $N_k$ são as quantidades de objetos nos grupos $\mathbf{C}_i$, $\mathbf{C}_j$ e $\mathbf{C}_k$, respectivamente
  - Métodos de centróides (UPGMC e WPGMC) subsumem dist. Euclidiana ao quadrado

# Hierarchical Clustering: Time and Space requirements

- $O(N^2)$ **space** since it uses the proximity matrix
  - N is the number of points

- $O(N^3)$ **time** in many cases
  - There are N steps and, at each step, the proximity matrix must be updated and searched
  - Complexity can be reduced for some approaches

# Métodos Divisivos

- Iniciam com um único *cluster*, que é sub-dividido em 2
- Recursivamente sub-divide cada um dos 2 *clusters*
  - Até que cada objeto constitua um **singleton**
- Em geral, são menos usados que os aglomerativos
  - É mais simples unir 2 *clusters* do que dividir...
    - número de modos para dividir N objetos em 2 *clusters* é $(2^{N-1} – 1)$. Por exemplo, para N=50 existem $5.63x10^{14}$ maneiras de se obter dois *clusters* !
- Questão:
  - *Como dividir um cluster ?*

- Heurística de **MacNaughton-Smith et al.** (1964):
  - Para um dado *cluster*, escolher o objeto mais distante dos demais
    - Este formará o *novo cluster*
  - Para cada objeto, calculam-se as distâncias médias deste aos objetos do *cluster* original e aos objetos do *novo cluster*
  - O objeto mais próximo do *novo cluster* e mais distante do *cluster* original é transferido para o *novo cluster* ; repete-se o processo

- Exemplo (Everitt et al., 2001):

$$\mathbf{D} = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{bmatrix} 0 & & & & & & \\ 10 & 0 & & & & & \\ 7 & 7 & 0 & & & & \\ 30 & 23 & 21 & 0 & & & \\ 29 & 25 & 22 & 7 & 0 & & \\ 38 & 34 & 31 & 10 & 11 & 0 & \\ 42 & 36 & 36 & 13 & 17 & 9 & 0 \end{bmatrix}$$

- Para este exemplo, objeto "1" é o mais distante (*novo cluster* – A)
- Demais objetos permanecem no *cluster principal* (*cluster* – B)
- *Clusters* obtidos: A={1} e B={2,3,4,5,6,7}
- Sejam $D_A$ e $D_B$ as distâncias médias de um objeto de B em relação aos objetos de A e B, respectivamente:

Mais próximos de A do que de B

| Objetos B | $D_A$ | $D_B$ | $D_B$-$D_A$ |
|---|---|---|---|
| 2 | 10 | 25 | 15,0 |
| 3 | 7 | 23,4 | 16,4 |
| 4 | 30 | 14,8 | -15,2 |
| 5 | 29 | 16,4 | -12,6 |
| 6 | 38 | 19,0 | -19,0 |
| 7 | 42 | 22,2 | -19,8 |

Objeto escolhido para mudar de *cluster*

Desta forma, obtemos os *clusters* {1,3} e {2,4,5,6,7}

Repetindo o processo temos ...

| Objetos B | $D_A$ | $D_B$ | $D_B$-$D_A$ |
|---|---|---|---|
| 2 | 8,5 | 29,5 | 12,0 |
| 4 | 25,5 | 13,2 | -12,3 |
| 5 | 25,5 | 15,0 | -10,5 |
| 6 | 34,5 | 16,0 | -18,5 |
| 7 | 39,0 | 18,7 | -20,3 |

*Mudar* para A

Novos *clusters*: {1,3,2} e {4,5,6,7}.

Próximo passo: todos ($D_B$-$D_A$) negativos;

Pode-se então repetir o processo em cada *cluster* acima, separadamente...

# Heurística de MacNaughton-Smith

- **Exercício**:
  - Aplicar o algoritmo hierárquico divisivo com heurística de MacNaughton-Smith et al. na seguinte base de dados:

$$\mathbf{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 5 & 0 & & \\ 10 & 9 & 4 & 0 & \\ 9 & 8 & 5 & 3 & 0 \end{bmatrix} \end{array}$$

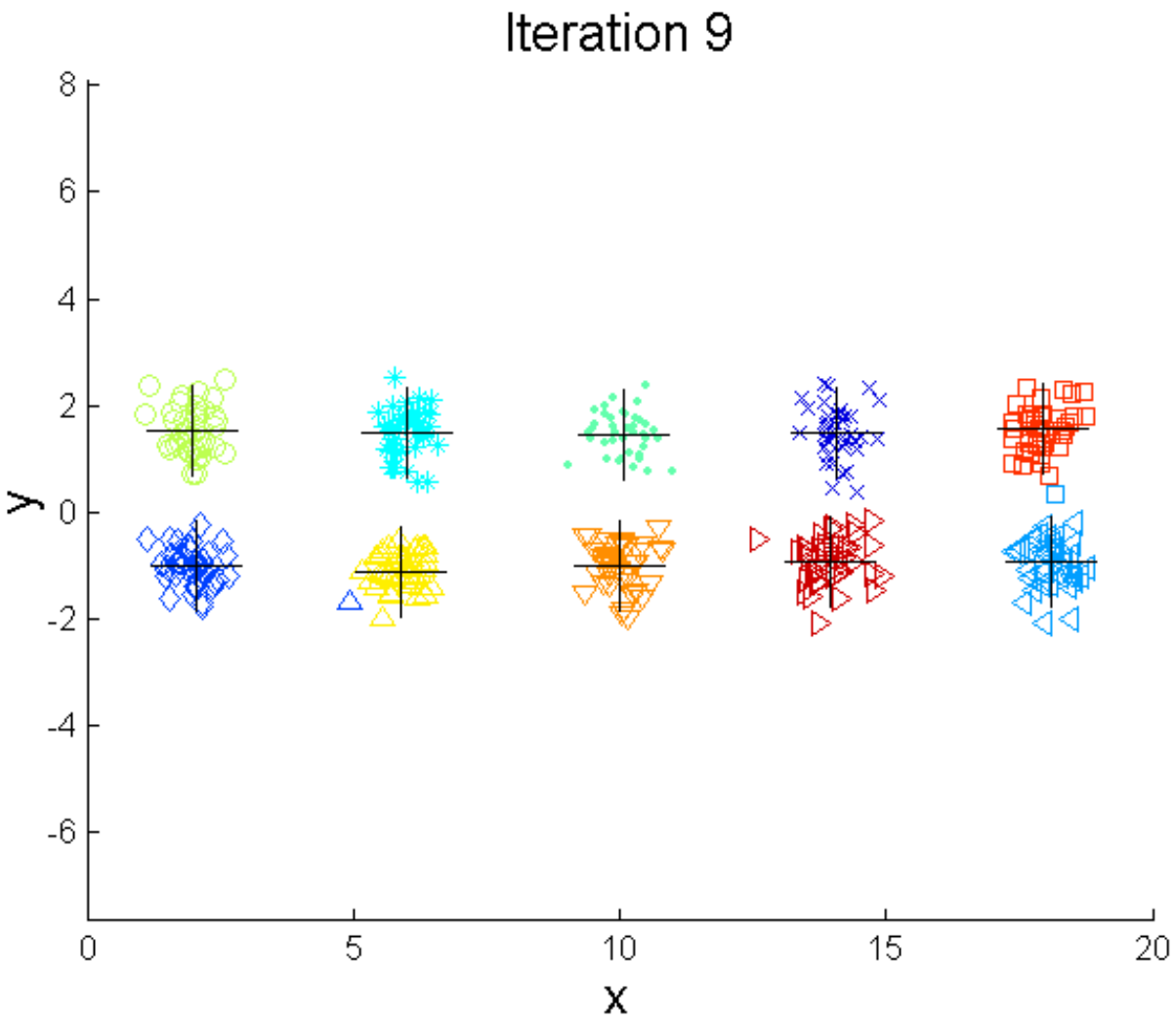
# Bisecting K-means

- Bisecting K-means algorithm
  - Variant of **K-means** that can produce a **partitional** or a **hierarchical** clustering

1: Initialize the list of clusters to contain the cluster containing all points.
2: **repeat**
3: Select a cluster from the list of clusters
4: **for** $i = 1$ to $number\_of\_iterations$ **do**
5: Bisect the selected cluster using basic K-means
6: **end for**
7: Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: **until** Until the list of clusters contains $K$ clusters

$$SSE(\mathbf{C}_i) = \sum_{\mathbf{x}_j \in \mathbf{C}_i} d(\mathbf{x}_j, \bar{\mathbf{x}}_i)^2 \quad \rightarrow \quad \text{Sum of Squared Errors (para o grupo } \mathbf{C}_i\text{)}$$

# Bisecting K-means Example



# Bisecting K-Means

- Note que fazendo K = N (no. total de objetos) no passo 8 do algoritmo, obtemos uma hierarquia completa
- No passo 3, a seleção do grupo a ser bi-seccionado pode ser feita de diferentes maneiras
  - Utiliza-se algum critério de avaliação de qualidade dos grupos, para eleger o "pior". Por exemplo:
    - Diâmetro máximo (sensível a outliers)
    - SSE normalizado pelo no. de objetos do grupo (mais robusto)
    - Critérios de avaliação de grupos individuais que consideram os objetos nos demais grupos (veremos posteriormente no curso)

# Bisecting K-Means

- **Complexidade Computacional**
  - k-means roda em O(Nkn)*, mas, como k = 2, tem-se O(Nn)
  - Assumimos por simplicidade que *no_of_iterations* = 1 no passo 4
  - **Pior Caso**: cada divisão separa apenas 1 objeto dos demais
    - O( Nn + (N-1)n + (N-2)n + ... + 2n) $\rightarrow$ **$O(N^2n)$**
  - **Melhor Caso**: cada divisão separa o grupo de forma balanceada
    - Árvore binária com $\log_2$ N níveis, cada um somando N objetos
    - **$O(n\ N \log_2 N)$**



* Assumindo distância com complexidade linear no no. de atributos

# Bisecting K-Means

- Um problema deste algoritmo é que as divisões via execução de k-means (discutido posteriormente no curso) com k = 2 grupos podem "quebrar" grupos naturais
  - Essas quebras não poderão ser corrigidas. Exemplo:

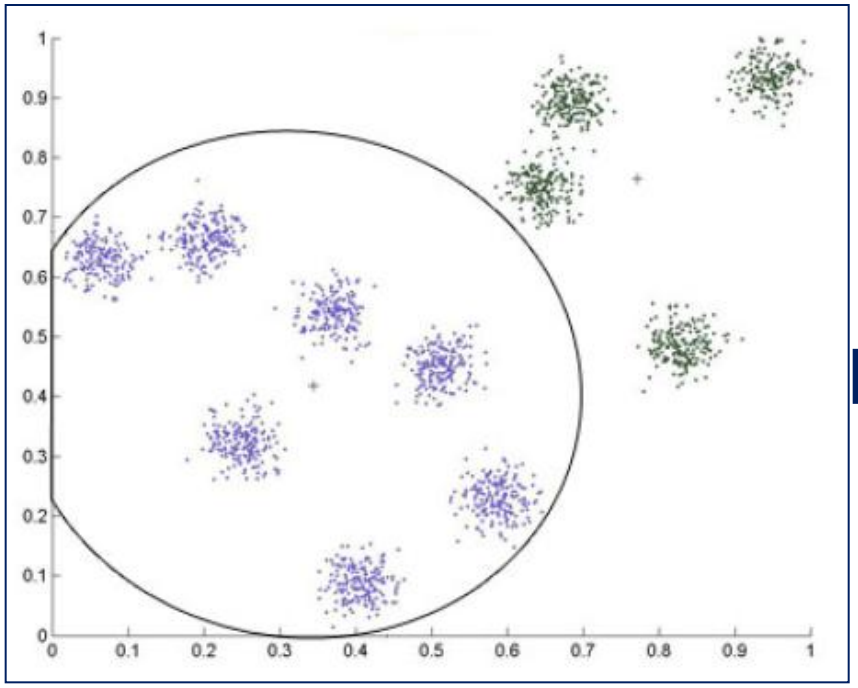



Figura por André Fontana

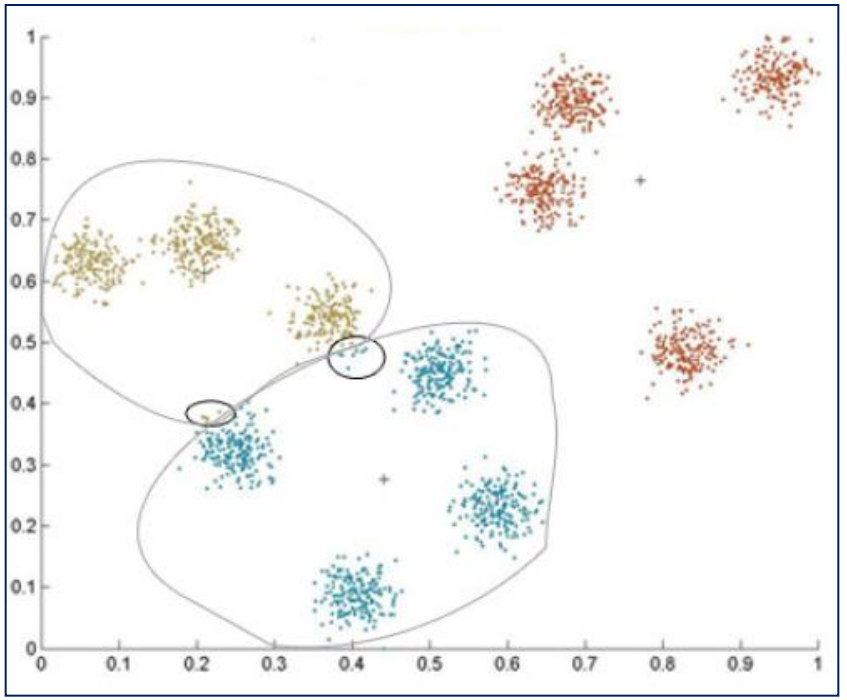
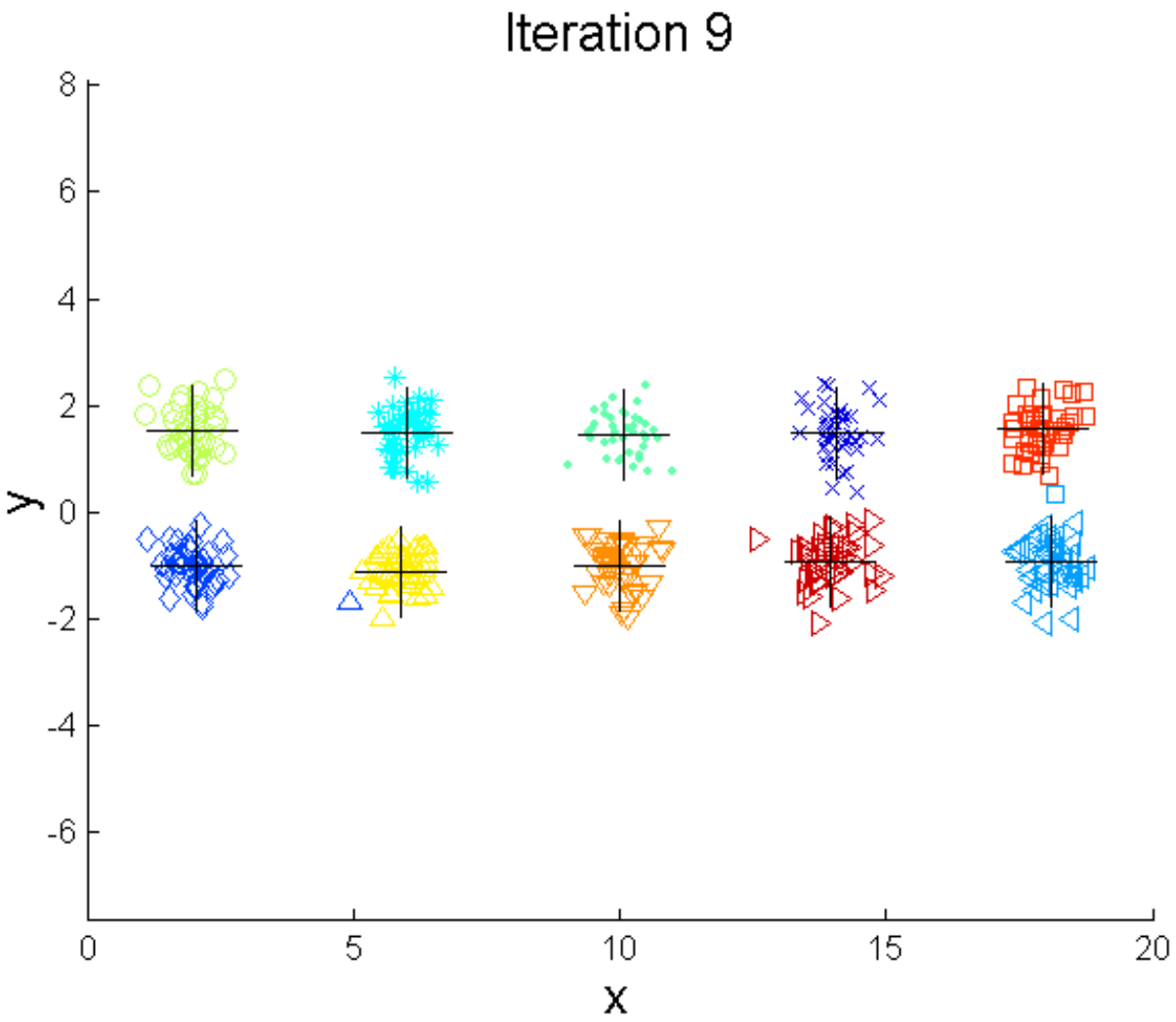



Figura por André Fontana

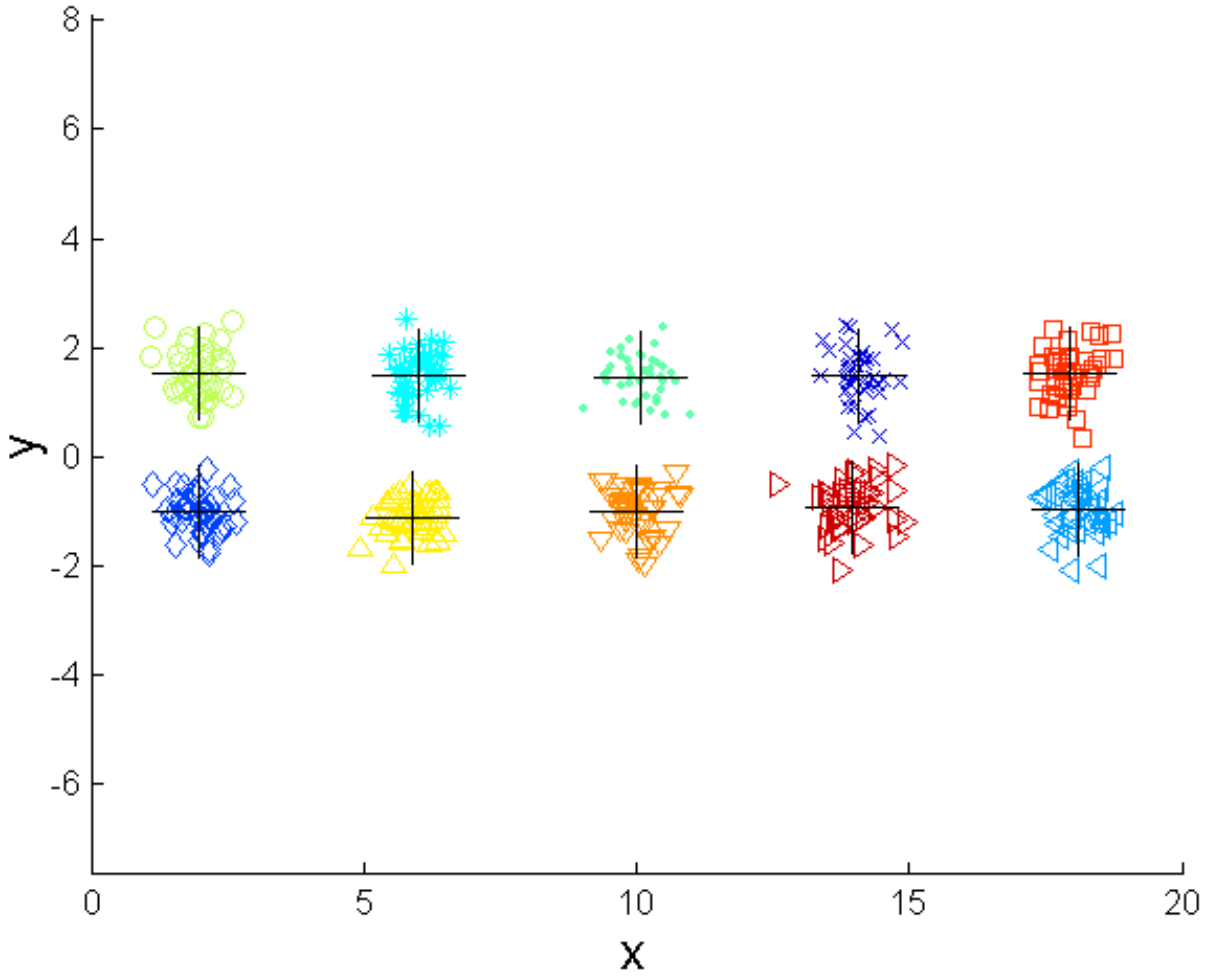
# Bisecting K-Means

- **Nota**: se queremos uma partição com k' grupos:
  - ao invés da hierarquia
  - podemos refinar a solução obtida com k' grupos
    - executando o próprio k-means com k = k'
- No exemplo anterior:
  - refinar a solução com 10 grupos executando k-means com k = 10
  - protótipos iniciais iguais aos finais obtidos via bisecting k-means
  - vide figuras a seguir

# Bisecting K-means Example

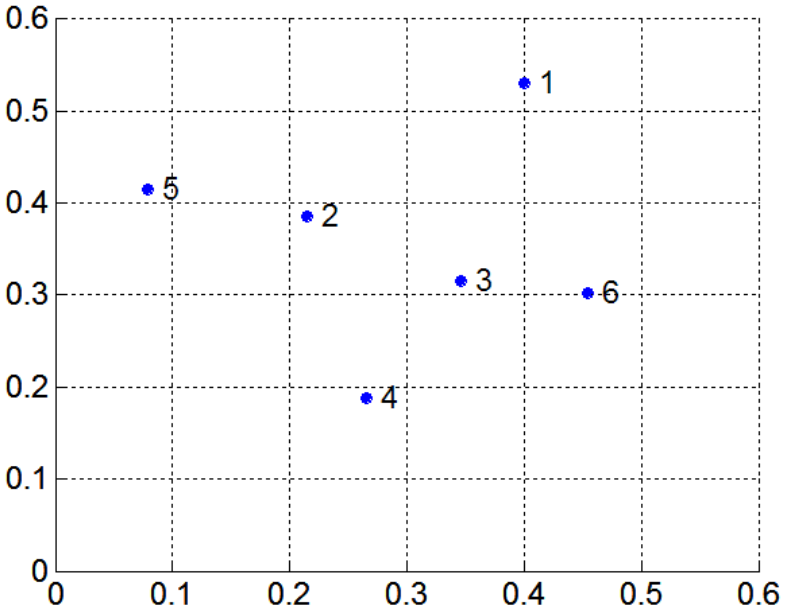


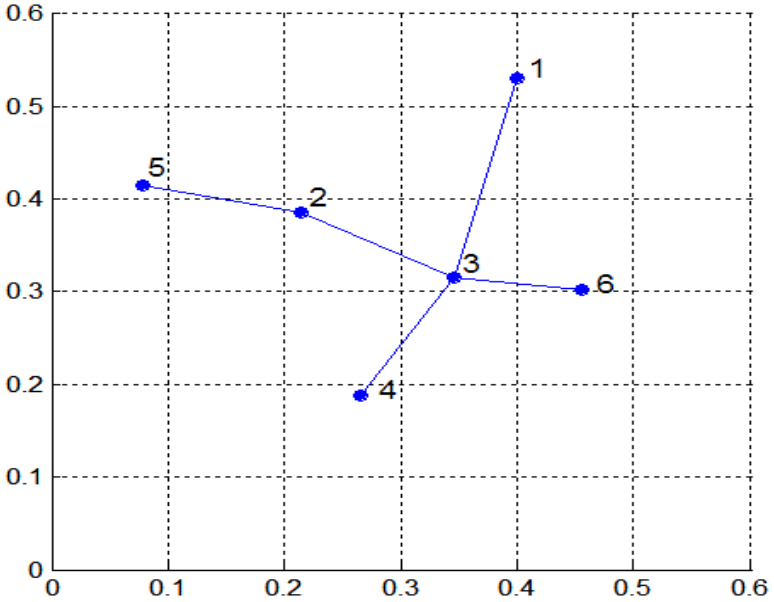
# Bisecting K-Means Example



# MST: Divisive Single-Linkage Clustering

- Build MST (Minimum Spanning Tree) for the **proximity graph**
  - Start with a tree that consists of any point
  - In successive steps, look for the closest pair of points (p, q) such that one point (p) is in the current tree but the other (q) is not
  - Add q to the tree and put an edge between p and q



# MST: Divisive Single-Linkage Clustering

- Use MST for constructing hierarchy of clusters

**Algorithm 7.5** MST Divisive Hierarchical Clustering Algorithm

1: Compute a minimum spanning tree for the proximity graph.
2: **repeat**
3: Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
4: **until** Only singleton clusters remain

# Single-Linkage via MSTs

- O método para construção de Árvores Geradoras Mínimas (MSTs) descrito anteriormente é chamado de **Algoritmo de Prim**
- Outro método similar que também pode ser utilizado é o **Algoritmo de Kruskal**
  - Comece com cada objeto sendo uma MST
  - Forme uma nova MST conectando as duas MSTs mais próximas através da menor aresta entre elas
  - Repita até que se tenha uma única MST

# Single-Linkage via MSTs

- Observe a relação direta entre o algoritmo de Kruskal para MSTs e o algoritmo *single-linkage*
- De fato, o procedimento divisivo subsequente à construção da MST é desnecessário nesse caso
  - MSTs parciais correspondem a grupos
  - Aresta mínima entre duas MSTs unidas a cada passo corresponde à distância entre dois grupos unidos
- Armazenando as MSTs parciais e as arestas de ligação, obtemos o **single-linkage aglomerativo**
  - torna mais evidente a relação entre single-linkage e MSTs

# Sumário dos Métodos Hierárquicos

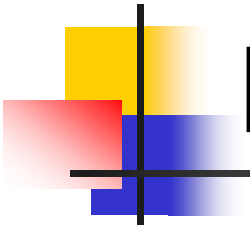
• **No. de Clusters**: não necessitam especificar o número de clusters *a priori*, mas de qualquer forma é necessário selecionar *a posteriori* ...

• **Procedimento Guloso**: não se pode reparar o que foi feito num passo anterior – não necessariamente leva à solução ótima

• **Escalabilidade**: complexidade de tempo $\Omega(N^2)$; $N$ = no. de objetos

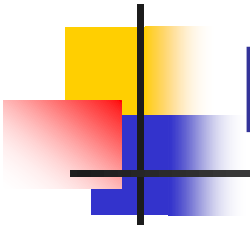
• **Interpretabilidade**: Produz uma hierarquia, que é aquilo desejado em muitas aplicações (e.g. taxonomia), e permite análise de outliers

• **Cálculo Relacional**: Não demandam matriz de dados original



# Leitura Sugerida

- Seções 3.1 e 3.2 de (Jain & Dubes, 1988)



# Referências


- 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, 4th Edition, 2001
- Tan, P.-N., Steinbach, M., and Kumar, V., *Introduction to Data Mining*, Addison-Wesley, 2006
- Gan, G., Ma, C., and Wu, J., Data Clustering: Theory, Algorithms and Applications, ASA SIAM, 2007
- Gordon, A. D., "Hierarchical Classification", Em Arabie et al. (Eds.), *Clustering and Classification*, World Scientific, 1996