Técnicas de Clustering: Algoritmos K-means e Aglomerative

Danilo Oliveira, Matheus Torquato

Centro de Informática
Universidade Federal de Pernambuco

9 de outubro de 2012

Sumario

Introdução

- Classificar é um dos métodos mais comuns no cotidiano.
- Em síntese, consiste num processo de atribuir rótulos a objetos que possuem características semelhantes.
- Constitue um dos alicerces da linguagem natural.

Por quê classificar?

- Organizar massas de dados;
- Aumentar a eficiência na recuperação da informação.

Por quê classificar?

- Crescimento da complexidade de alguns conjuntos de dados;
- Necessidade de padrões bem-definidos para tomada de decisão.

Por quê classificar?

- Crescimento da complexidade de alguns conjuntos de dados;
- Necessidade de padrões bem-definidos para tomada de decisão.

Clustering!

Clustering

- O objetivo principal é fornecer um método para efetuar uma classificação estável e objetiva;
- Nomenclatura adotada
  - **Análise de cluster**;
  - **Taxonomia numérica** - Biologia;
  - ***Q Analysis*** - Psicologia;
  - **Reconhecimento de padrões não-supervisionado** - Inteligência Artificial;
  - **Segmentação** - Pesquisa de mercado

Introdução

Definição

É o procedimento de classificar objetos de dados em diferentes grupos, baseados nos atributos de cada objeto e seus relacionamentos, utilizando algum método numérico. O objetivo é que os objetos em um mesmo grupo sejam similares (ou relacionados) e diferentes (ou não relacionados) com os objetos de outros grupos.

Entendendo *clustering*

- Suponha a seguinte situação:
  - Existe uma rede de sorveterias que vende sorvetes de baunilha e chocolate;
  - Deseja-se avaliar as regiões onde mais sorvetes foram vendidos.

	Chocolate	Baunilha
Sorveteria 1	12	6
Sorveteria 2	15	16
Sorveteria 3	18	17
Sorveteria 4	10	8
Sorveteria 5	8	7
Sorveteria 6	9	6
Sorveteria 7	12	9
Sorveteria 8	20	18

Introdução

Problemas na definição precisa do que constitui um grupo:

(a) Original points.

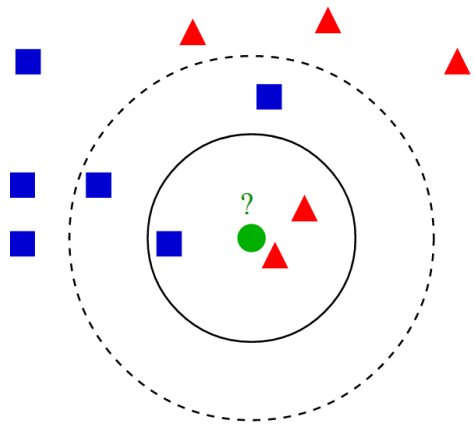
(b) Two clusters.

(c) Four clusters.

(d) Six clusters.

Introdução

- Classificação supervisionada X Classificação não supervisionada
  - Técnicas de clustering são relacionadas com técnicas que também tem como objetivo classificar objetos em diferentes categorias, mas que fazem uso de objetos previamente classificados para descobrir o grupo de objetos desconhecidos. Esse tipo de abordagem é denominado **classificação supervisionada**

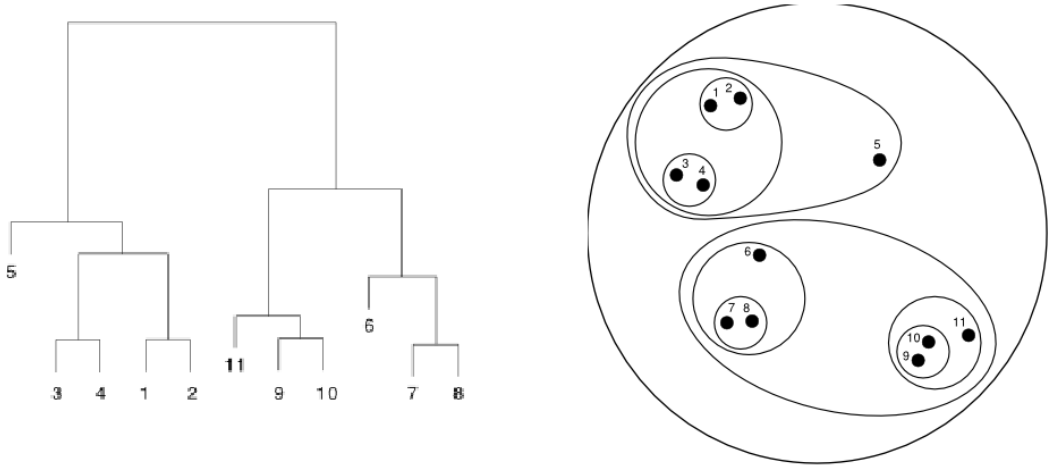


  - As técnicas de clustering, por sua vez, não possuem nenhum tipo de conhecimento prévio, ou dados de "treinamento", portanto entram na categoria das técnicas de classificação **não supervisionadas**

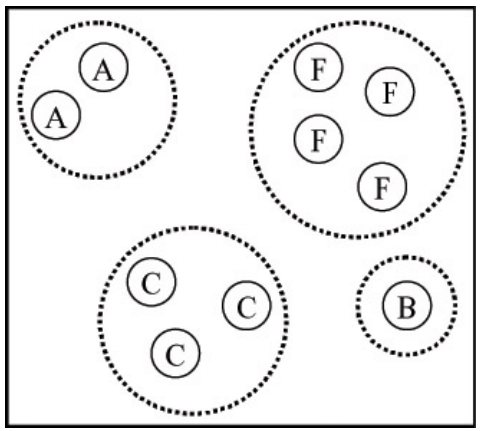
Tipos de clustering

- Hierárquico versus Particional

Hierárquico:

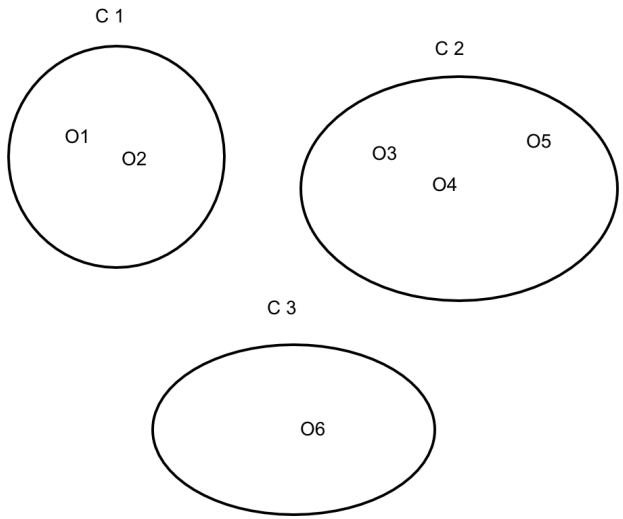


Particional:

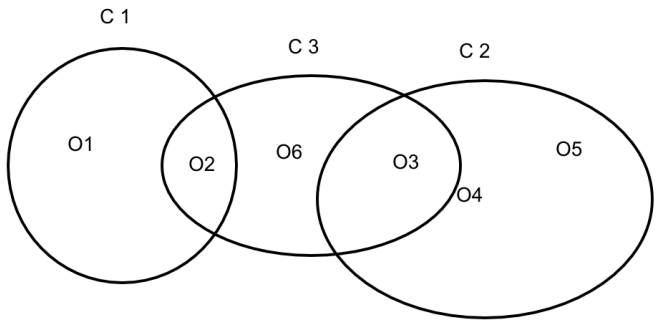


Tipos de clustering

- Exclusivo versus Com sobreposição versus Fuzzy
  - **Exclusivo** - Atribui cada elemento a um cluster

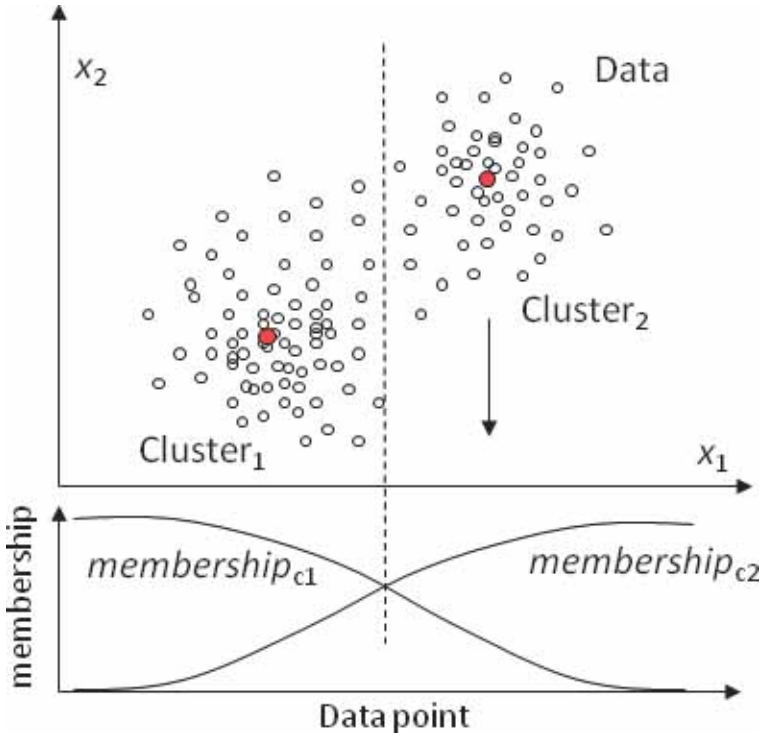


  - **Com sobreposição ou não exclusivo**- Um elemento pode pertencer à mais de um cluster



Tipos de clustering

- Exclusivo versus Com sobreposição versus Fuzzy
  - **Fuzzy** - Um elemento pode estar em qualquer cluster, com um determinado peso, que varia entre 0 (definitivamente não pertence) a 1 (absolutamente pertence). O somatório de todos os pesos deve ser igual à 1.



Tipos de clustering

- Completo versus Parcial
  - **Completo - Todo elemento é atribuído a algum cluster**
  - **Parcial - Alguns elementos podem não ser atribuídos à nenhum cluster, sendo tratados como "ruído"**

Características

- O processo de *clustering* exige métodos que apresentem as seguintes características [1]:
  - Ser capaz de lidar com dados com alta dimensionalidade;
  - Ser "escalável" com o número de dimensões e com a quantidade de elementos a serem agrupados;
  - Habilidade para lidar com diferentes tipos de dados;
  - Capacidade de definir agrupamentos de diferentes tamanhos e formas;
  - Exigir o mínimo de conhecimento para determinação dos parâmetros de entrada;
  - Ser robusto à presença de ruído;
  - Apresentar resultado consistente independente da ordem em que os dados são apresentados.

Sumario

Medidas de similaridade

- Para poder agrupar conjuntos de objetos é necessário medir a similaridade entre eles.
- Esta medida é obtida com um cálculo de distância entre os objetos.
- Métodos para cálculo de distância:
  - **Distância Euclidiana**
  - **Distância Euclidiana Quadrática**
  - **Distância *Manhattan***
  - **Distância *Chebychev***

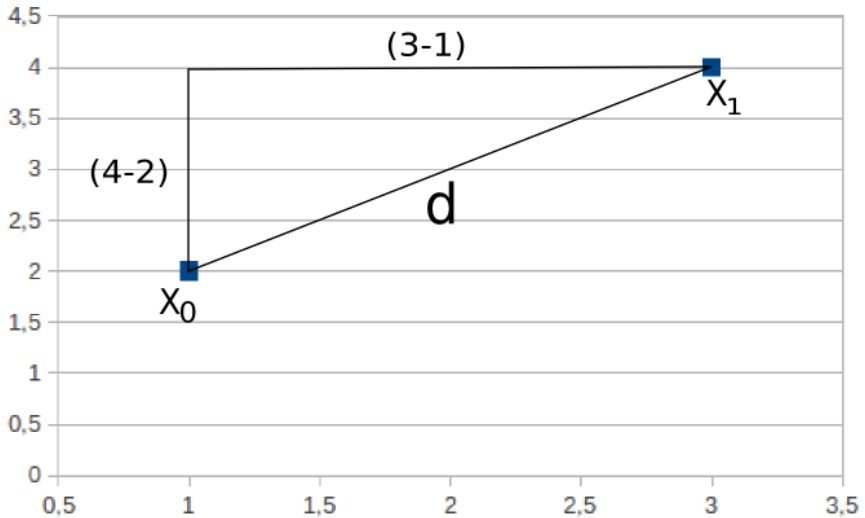
Distância Euclidiana

- Distância geométrica no espaço multidimensional.
- Tendo-se X = $[X_1, X_2, ..., X_p]$ e Y = $[Y_1, Y_2, ..., Y_p]$, a distância entre os pontos é definida por:

$$d_{xy} = \sqrt{(X_1 - Y_1)^2 + (X_1 - Y_1)^2 + ...(X_p - Y_p)^2} = \sqrt{\sum_{i=1}^{p}(X_i - Y_i)^2}$$

Exemplo

- Calcular a distância entre os elementos $X_0 = (1,2)$ e $X_1 = (3,4)$ (espaço euclidiano)



$$d_{x_0 x_1} = \sqrt{(3-1)^2 + (4-2)^2} = \sqrt{8} = 2,83$$

Distância euclidiana quadrática

- Mais sensível que a distância euclidiana.
- Definida pela expressão

$$d_{xy} = (X_1 - Y_1)^2 + (X_2 - Y_2)^2 + ...(X_p - Y_p)^2 = \sum_{i=1}^{p}(X_i - Y_i)^2$$

- Considerando-se os mesmos pontos X0 e X1 do exemplo anterior, observa-se a intensificação da distância:

$$d_{x_0 x_1} = (3 - 1)^2 + (4 - 2)^2 = 8$$

Mais distâncias

- **Manhattan** - Sensível a minimizações.

$$d_{xy} = |X_1 - Y_1| + |X_2 - Y_2| + ...|X_p - Y_p| = \sum_{i=1}^{p} |X_i - Y_i|$$

- **Chebychev** - Diferentes se ao menos uma dimensão é diferente.

$$d_{xy} = maximo(|X_1 - Y_1| + |X_2 - Y_2| + ...|X_p - Y_p|)$$

Matriz de similaridade

- São utilizadas em clustering para determinar a distância entre os elementos
- As distâncias são armazenadas na **matriz de similaridade**;
- Esta matriz é simétrica e utiliza, em geral a distância euclidiana.

Matriz de similaridade - Exemplo

ELEMENTO	X	Y
1	4	3
2	2	7
3	4	7
4	2	3
5	3	5
6	6	1

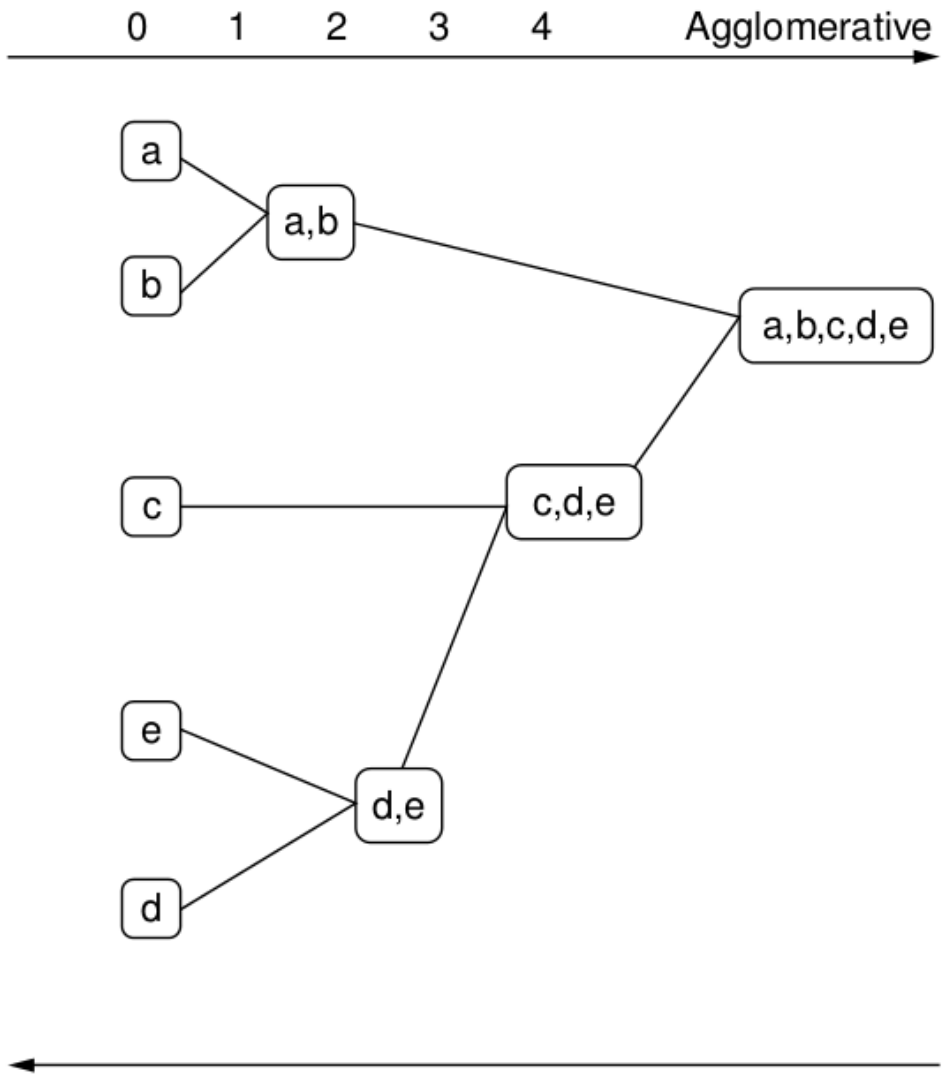
$$D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & 4{,}47 & 4 & 2 & 2{,}24 & 2{,}83 \\ 4{,}47 & 0 & 2 & 4 & 2{,}24 & 7{,}21 \\ 4 & 2 & 0 & 4{,}47 & 2{,}24 & 6{,}32 \\ 2 & 4 & 4{,}47 & 0 & 2{,}24 & 4{,}47 \\ 2{,}24 & 2{,}24 & 2{,}24 & 2{,}24 & 0 & 5 \\ 2{,}83 & 7{,}21 & 6{,}32 & 4{,}47 & 5 & 0 \end{bmatrix}$$

Sumario

Clustering hierárquico

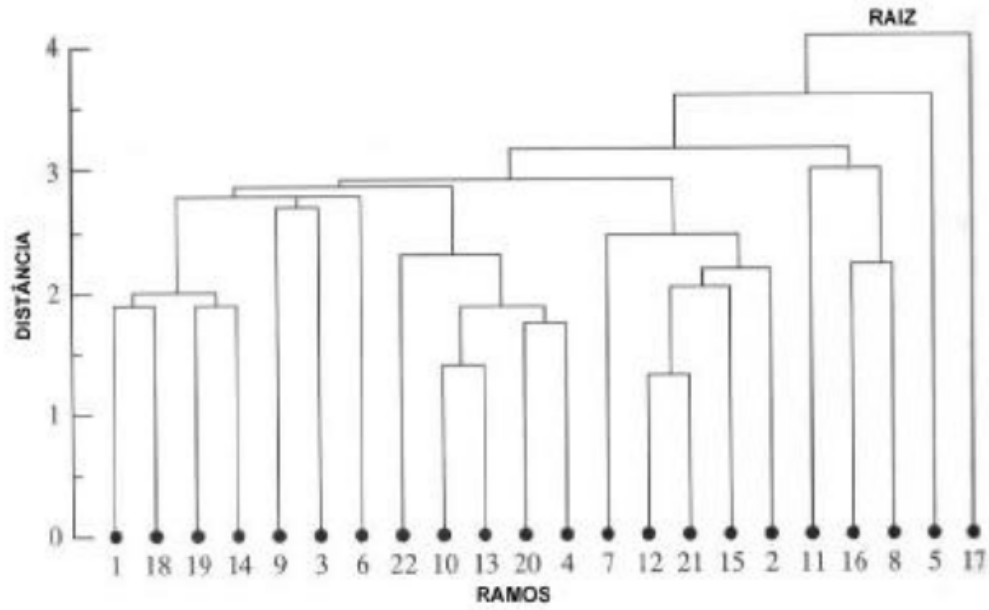
- Consiste em uma série de:
  - Agrupamentos sucessivos - **Algoritmo Agglomerative**
  - Divisões de elementos - **Algoritmo Divisive**.
- **Vantagem** - Implementação simples.
- **Desvantagem** - Uma vez feita a divisão ou união de objetos e/ou clusters, é irrevogável.

Clustering hierárquico



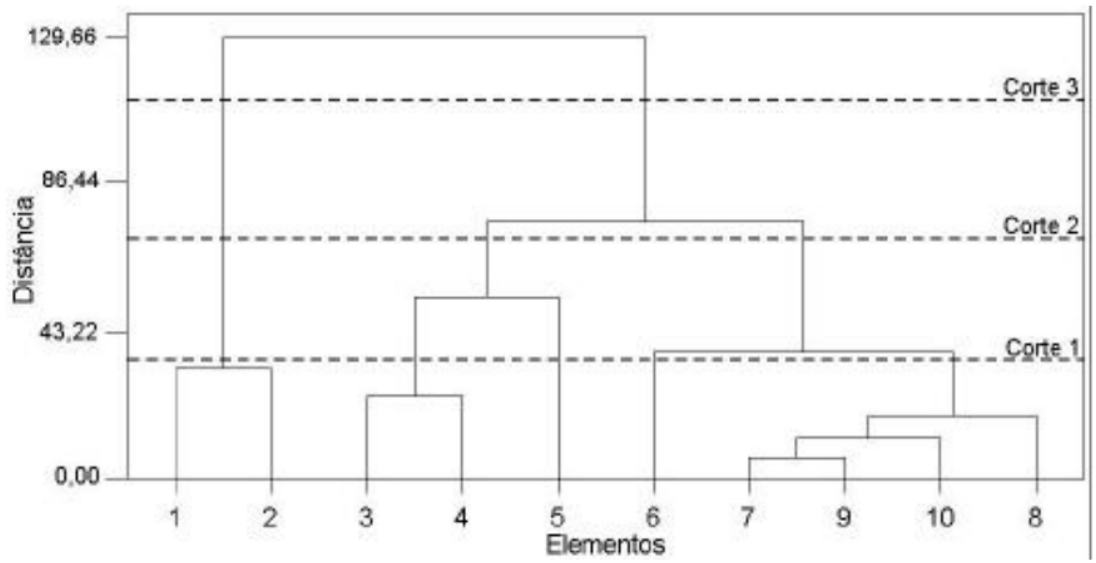
Clustering hierárquico

- A representação dos *clusters* obtidos através do método hierárquico é geralmente apresentado em forma de um diagrama bi-dimensional chamado de **dendrograma** ou **diagrama de árvore** [2].
- Neste tipo de diagrama, os ramos representam os elementos e a raiz o agrupamento de todos eles.



Clustering hierárquico

- Deve-se definir no dendrograma uma distância de corte.
  - Atribuir limites para os grupos.
- É necessário ter conhecimento prévio sobre a estrutura de dados observada;
- A atribuição de distâncias de corte é subjetiva.



Clustering hierárquico

- Outro método para representação de *clusters* hierárquicos é a utilização do diagrama de cluster aninhado.
- A técnica pode ser utilizada em conjuntos de pontos bi-dimensionais.

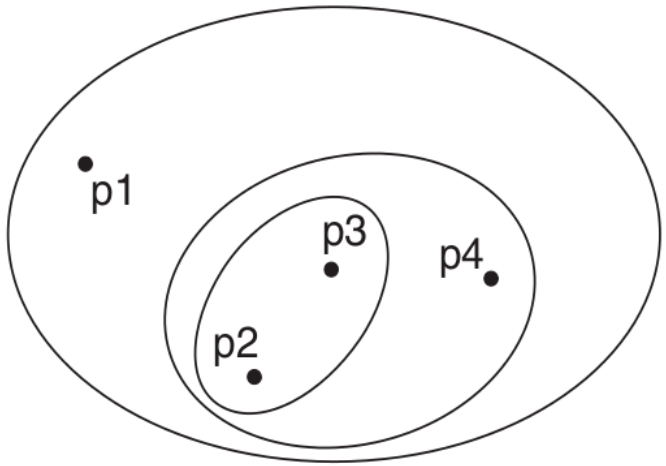


Figura: Diagrama de cluster aninhado

Como ligar os grupos?

- Dentro dos métodos aglomerativos existe uma variedade de critérios para definir a distância entre os grupos [2].
- A maioria dos métodos utilizam conceitos de agrupamento aglomerativo:
  1. Método de ligação (*single linkage*, *complete linkage*, *average linkage*, *median linkage*) ;
  2. Métodos de centróide;
  3. Métodos de soma de erros quadráticos ou variância (método de *Ward*).

Clustering hierárquico

- O padrão seguido por algoritmos de clustering hierárquico é exibido a seguir.
- As diferenças entre os métodos ocorre no passo 2.3 onde a função distância é definida de acordo com cada método [2].

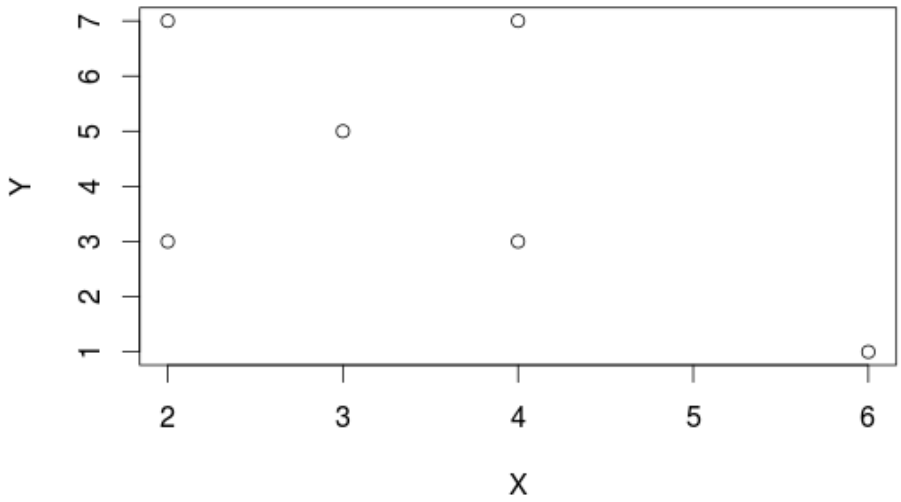
Algoritmo agglomerative

Entrada: Uma base de dados com N elementos.
Saída: Um conjunto de grupos.

1. Iniciar com N grupos, contendo um elemento em cada grupo e uma matriz de similaridade $D_N x_N$
2. Repitir
   1. Localizar a menor distância d_{uv} (maior similaridade);
   2. Atualizar a matriz D, retirando os elementos U e V;
   3. Atualizar a matriz D, adicionando as novas distâncias do grupo (U,V);
   4. Até N-1, quando todos elementos estarão em um único grupo.

Entendendo os algoritmos de clustering hierárquico

- Consideremos esta distribuição de pontos antes de estudarmos os algoritmos.



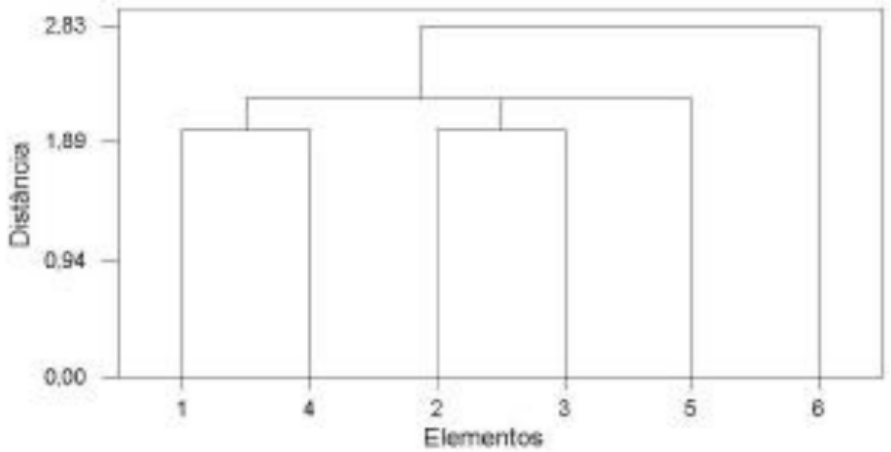
$$D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & 4{,}47 & 4 & 2 & 2{,}24 & 2{,}83 \\ 4{,}47 & 0 & 2 & 4 & 2{,}24 & 7{,}21 \\ 4 & 2 & 0 & 4{,}47 & 2{,}24 & 6{,}32 \\ 2 & 4 & 4{,}47 & 0 & 2{,}24 & 4{,}47 \\ 2{,}24 & 2{,}24 & 2{,}24 & 2{,}24 & 0 & 5 \\ 2{,}83 & 7{,}21 & 6{,}32 & 4{,}47 & 5 & 0 \end{bmatrix}$$

Método Single Linkage ou ligação por vizinho mais próximo

- Ligação por vizinho mais próximo

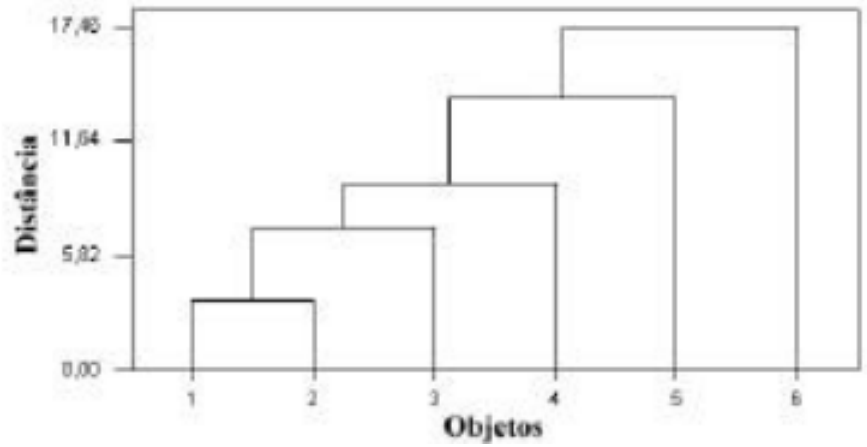
$$d_{UV}W = min(d_{UW}, d_{VW})$$

- Após as iterações do algoritmo na matriz de exemplo obtemos os seguinte dendrograma.



Método Single Linkage ou ligação por vizinho mais próximo

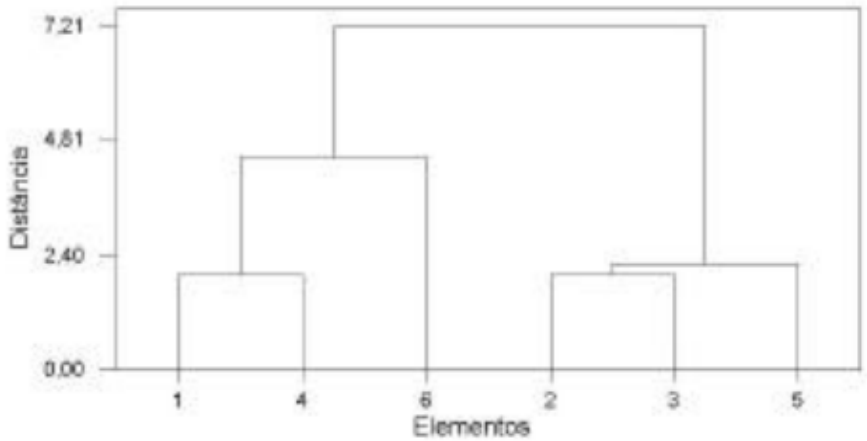
- Características [3]
  - Grupos muito próximos podem não ser identificados;
  - Permite detectar grupos de formas não-elípticas;
  - Apresenta pouca tolerância a ruído, pois tem tendência a incorporar os ruídos em um grupo já existente;
  - Apresenta bons resultados tanto para distâncias Euclidianas quanto para outras distâncias;
  - Tendência a formar longas cadeias (encadeamento).



Método Complete Linkage ou ligação por vizinho mais distante

- Ligação por vizinho mais distante

$$d_{UV}W = max(d_{UW}, d_{VW})$$



Método Complete Linkage ou ligação por vizinho mais distante

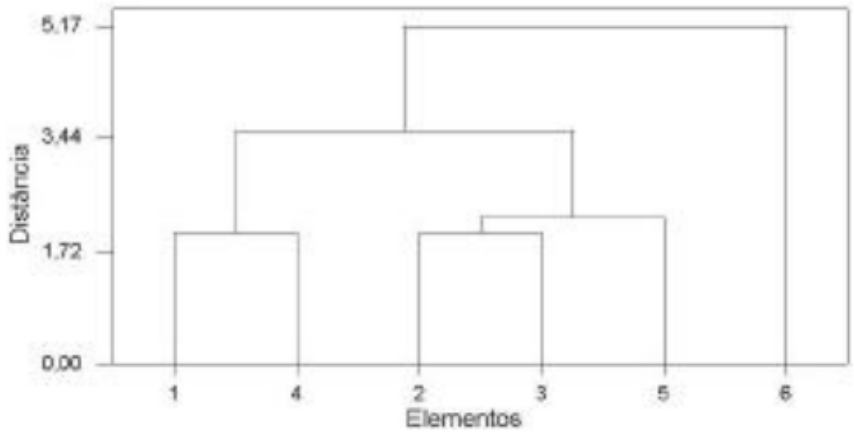
- Algumas características desse método são [4]:
  - Bons resultados tanto para distâncias Euclidianas quanto para outras distâncias;
  - Tendência a formar grupos compactos;
  - Os ruídos demoram a serem incorporados ao grupo.

Método Average Linkage ou ligação por média

- Ligação por média das distâncias

$$d_{UV}W = \frac{N_u.d_{UW} + N_v.d_{VW}}{N_u + N_v}$$

- Onde:
  - N_u e N_V -> número de objetos em U e V, respectivamente;
  - d_{UW} e d_{VW} -> distâncias entre os elementos UW e VW, respectivamente;

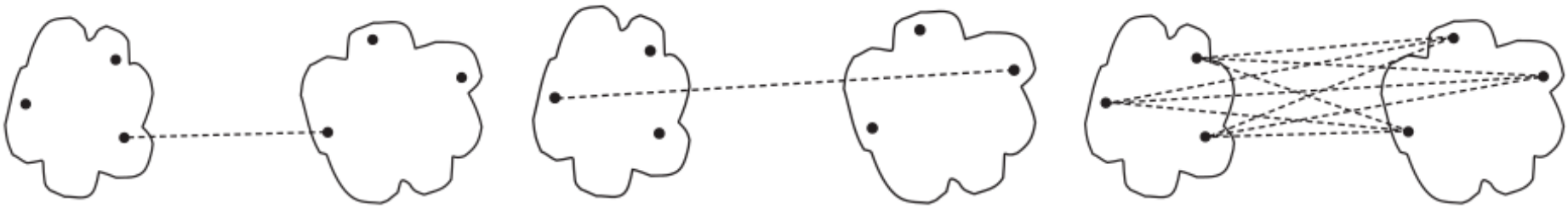


Método Average Linkage ou ligação por média

- Algumas características desse método são [4]:
  - Menor sensibilidade à ruídos que o os métodos anteriores;
  - Bons resultados tanto para distâncias Euclidianas quanto para outras distâncias;
  - Tendência a formar grupos com número de elementos similares.

Resumindo...

- Representação dos métodos de modo gráfico.



(a) MIN (single link.) (b) MAX (complete link.) (c) Group average.

Resumindo...

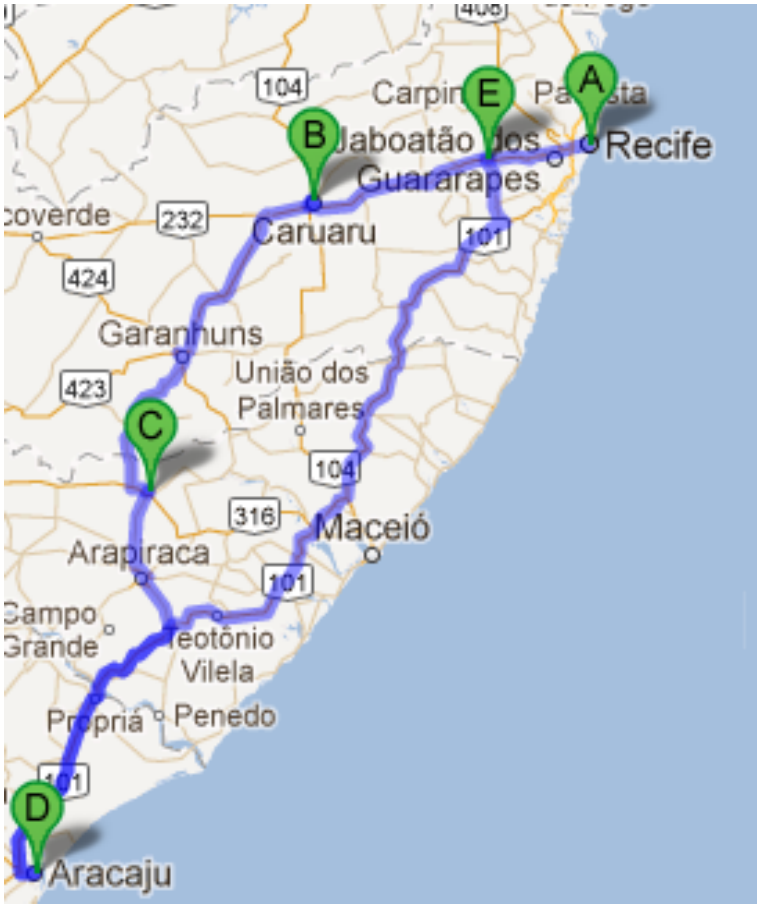
- Existem mais métodos para ligação dos objetos.

MÉTODO	DISTÂNCIA	CARACTERÍSTICAS
Ligação por vizinho mais próximo	$d_{(UV)W} = \min(d_{UW}, d_{VW})$	Sensibilidade à ruídos. Encadeamento.
Ligação por vizinho mais distante	$d_{(UV)W} = \max(d_{UW}, d_{VW})$	Tendência a formar grupos compactos.
Ligação por média	$d_{(UV)W} = \frac{(N_u . d_{UW} + N_v . d_{VW})}{N_u + N_v}$	Tendência a formar grupos com número de elementos similares.
Ligação por centróide	$d_{(UV)W} = \frac{N_U . d_{UW} + N_V . d_{VW}}{N_U + N_V} - \frac{N_U . N_V . d_{UV}}{(N_U + N_V)^2}$	Robustez à ruídos. Reversão.
Ligação por mediana	$d_{(UV)W} = \frac{d_{UW} + d_{VW}}{2} - \frac{d_{UV}}{4}$	Robustez à ruídos.
***Ligação de* Ward**	$d_{(UV)W} = \frac{((N_W + N_U) . d_{UW} + (N_W + N_V) . d_{VW} - N_W . d_{UV})}{N_W + N_U + N_V}$	Sensibilidade à ruídos.

Figura: Resumo dos métodos. Extraído de [2]

Exemplo prático

- Vamos praticar!
- Considerando o mapa abaixo, encontrar quais são as cidades mais próximas.



Exemplo prático

- A matriz de distancias entre as cidades.

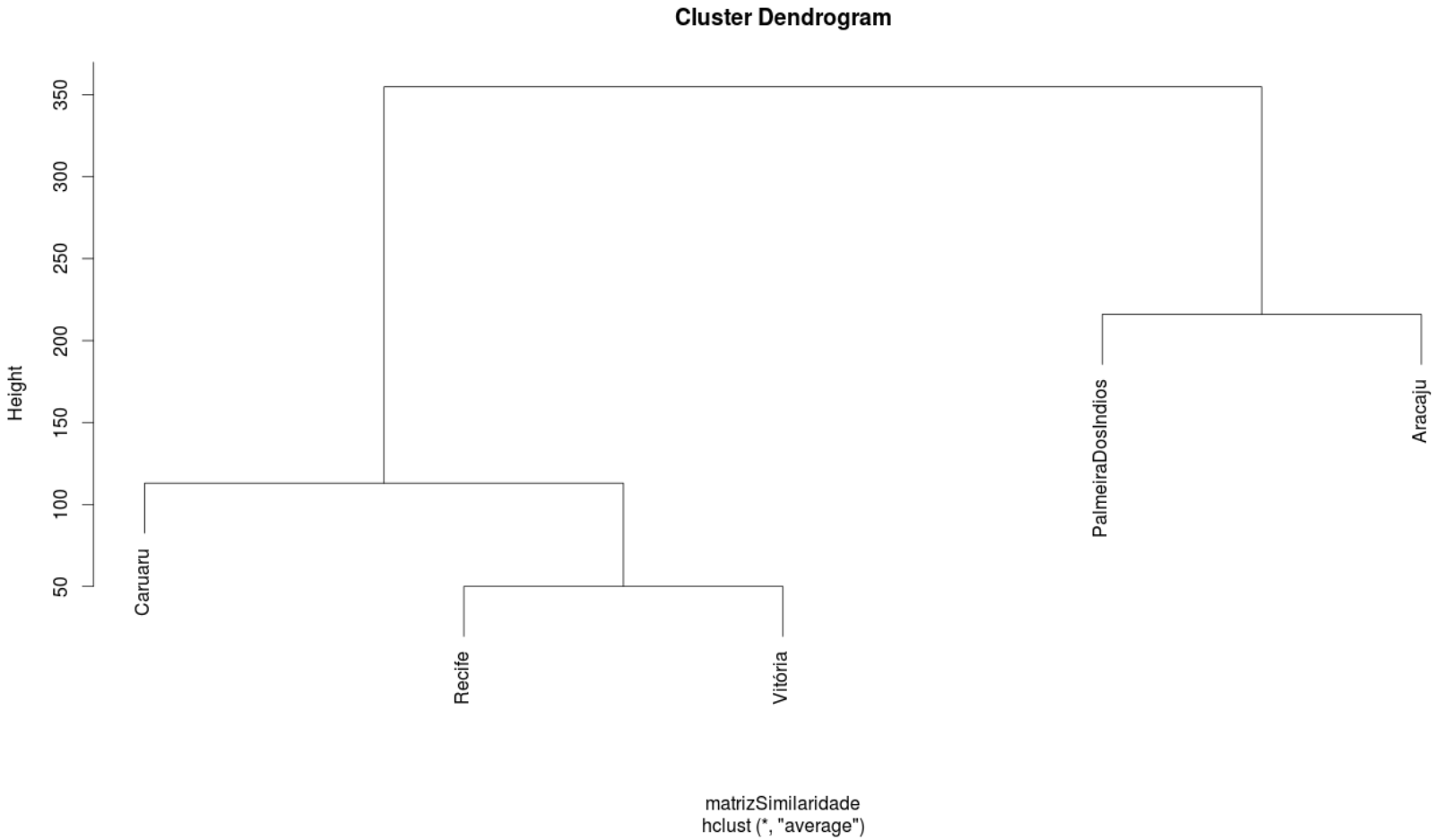
Recife	0	138	50	315	495
Caruaru	138	0	88	184	400
Vitória	50	88	0	267	468
Palmeira dos Índios	315	184	267	0	216
Aracaju	495	400	468	216	0

Utilizando o R

- Funções
  - **read.csv()** -> Importa um arquivo csv para um dataframe;
    Recebe como parâmetros o arquivo e o separador.
  - **as.dist() e dist()** -> Criação da matriz de similaridade;
    Recebe o data-frame como parâmetro.
  - **hclust()** -> Executa o algoritmo de clustering hierarquico, obecendo ao método passado.
    Recebe a matriz de similaridade e o método utilizado para construção do dendrograma.
  - **plot()** -> Plotar o dendrograma.

Exemplo prático

- Utilizando o método de ligação por média temos o seguinte resultado.



Sumario

K-means

- Características
  - Método particional, completo, exclusivo e baseado em protótipo de clustering
  - Não requer a computação de todas as possiveis distâncias entre objetos
  - Você precisa definir de antemão o número K de clusters

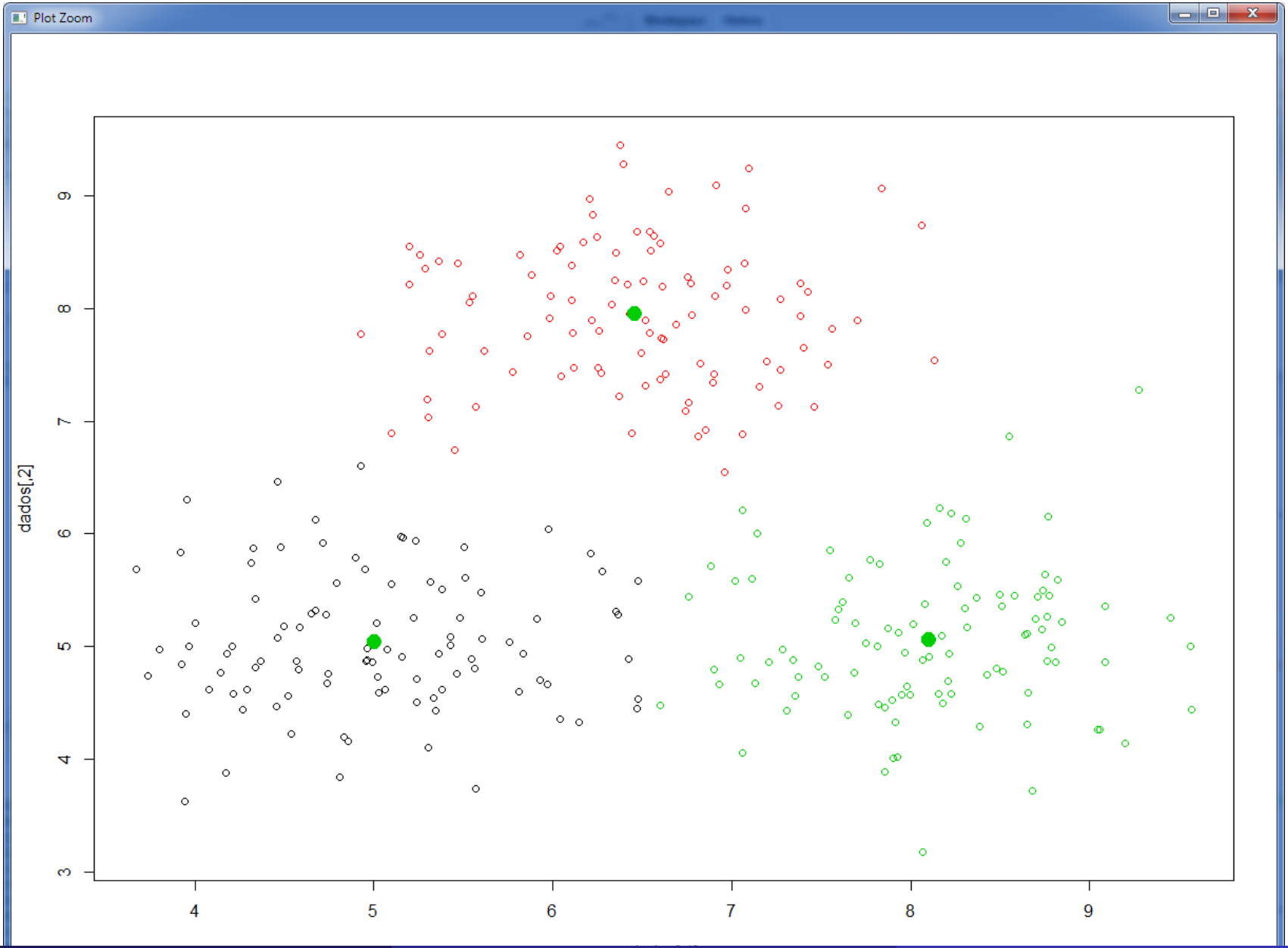
Algoritmo K-means

1. Escolha um número k de *clusters*
2. Escolha k pontos iniciais para serem utilizados como estimativas dos centróides
3. Examine cada ponto da série e coloque-o no *cluster* cujo centróide que estiver mais próximo. A posição do centróide é recalculada cada vez que um novo ponto é adicionado ao *cluster*
4. Repita o passo 3 até que não haja mudança no *cluster* ou um número máximo de passos seja executado

Utilizando o k-means da API do R

```
require(MASS)

# Criando três massas de dados distintas
X <- mvrnorm(n = 100, mu = c(5,5),
    Sigma = matrix(c(1/2, 0, 0, 1/2), nrow = 2, ncol = 2))
Y <- mvrnorm(n = 100, mu = c(6.5,8),
    Sigma = matrix(c(1/2, 0, 0, 1/2), nrow = 2, ncol = 2))
Z <- mvrnorm(n = 100, mu = c(8, 5),
    Sigma = matrix(c(1/2, 0, 0, 1/2), nrow = 2, ncol = 2))

# Unindo as três massas de dados em uma só amostra
dados <- rbind(X, Y, Z)


plot(dados)

#Aplicando o algoritmo k-means para k=3
kmeans.res <- kmeans(dados, centers=3)

#Plotando os pontos com uma cor para distinguir os clusters
plot(dados, col=kmeans.res$cluster)

#Plotando os centroides de cada cluster
points(kmeans.res$centers, cex=2, col=11, pch=19)
```

K-means



K-means

- Soma dos erros quadrados (SSE)
  - Considere um conjunto de dados cuja medida de proximidade é a distância Euclidiana. A **soma dos erros quadrados** é uma medida de qualidade do resultado da operação de clustering

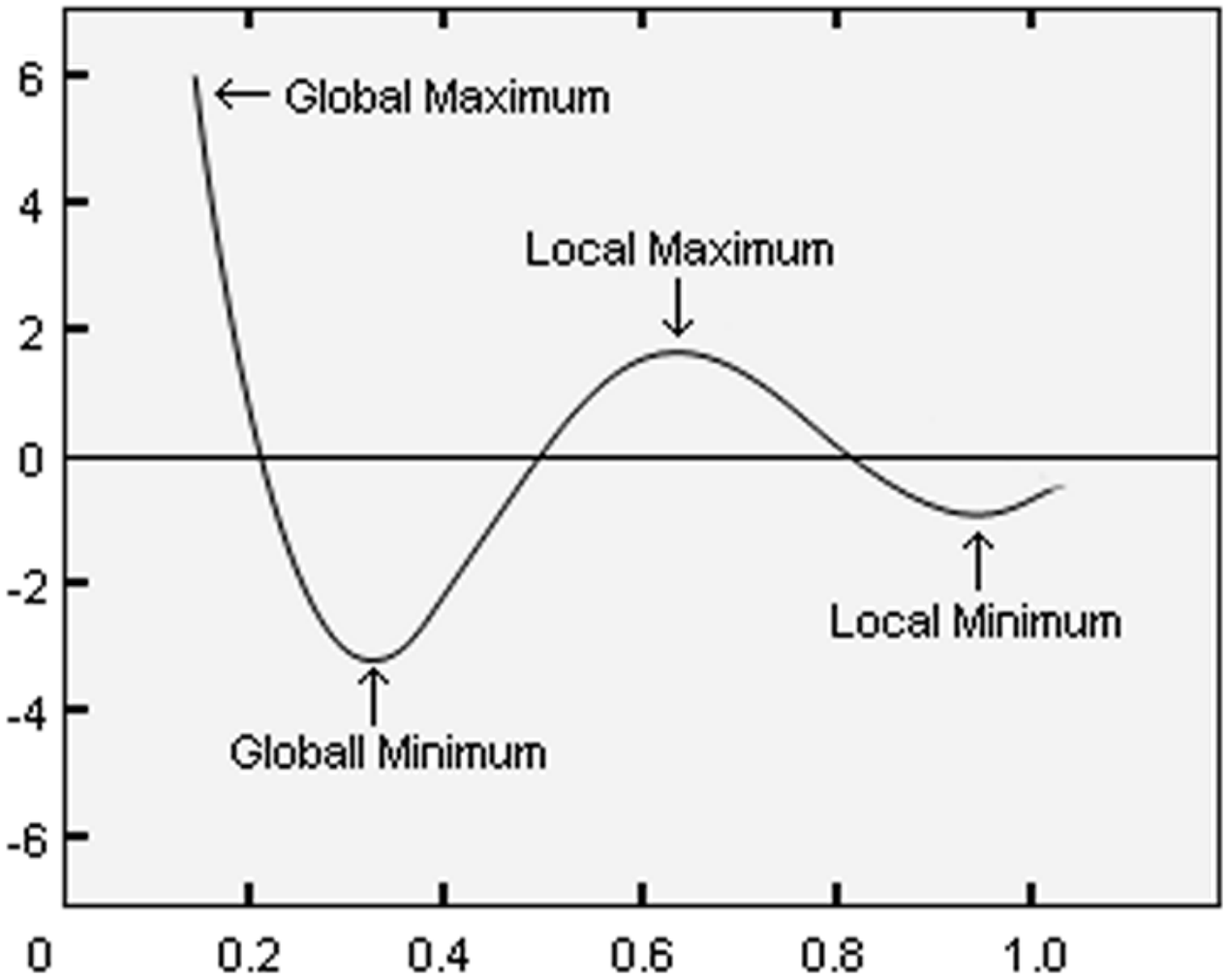
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(c_i, x)^2$$

  - Dados dois diferentes resultados que são produzidos por duas execuções diferentes do K-means, nós preferimos aquela que resultar no menor SSE
  - O K-means de fato consegue otimizar o SSE, mas pode levar à um mínimo local

Obtendo SSE no R

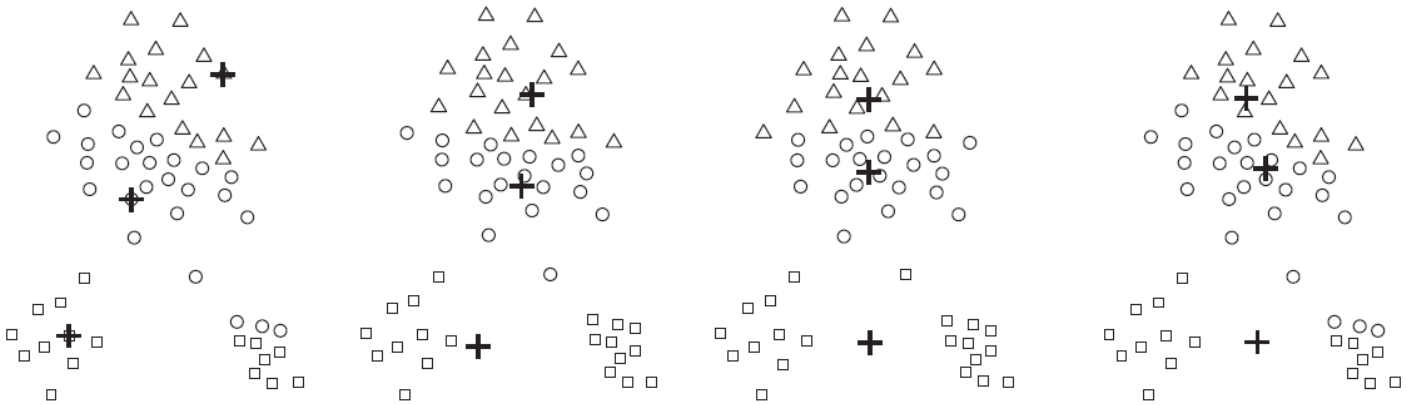
```
km = kmeans(x, n_clusters)
print(km$withinss) # imprime SSE de cada cluster
print(km$tot.withinss) # imprime SSE total
```

K-means


← Global Maximum
Local Maximum
Local Minimum
Globall Minimum
6
4
2
0
-2
-4
-6
0
0.2
0.4
0.6
0.8
1.0

K-means

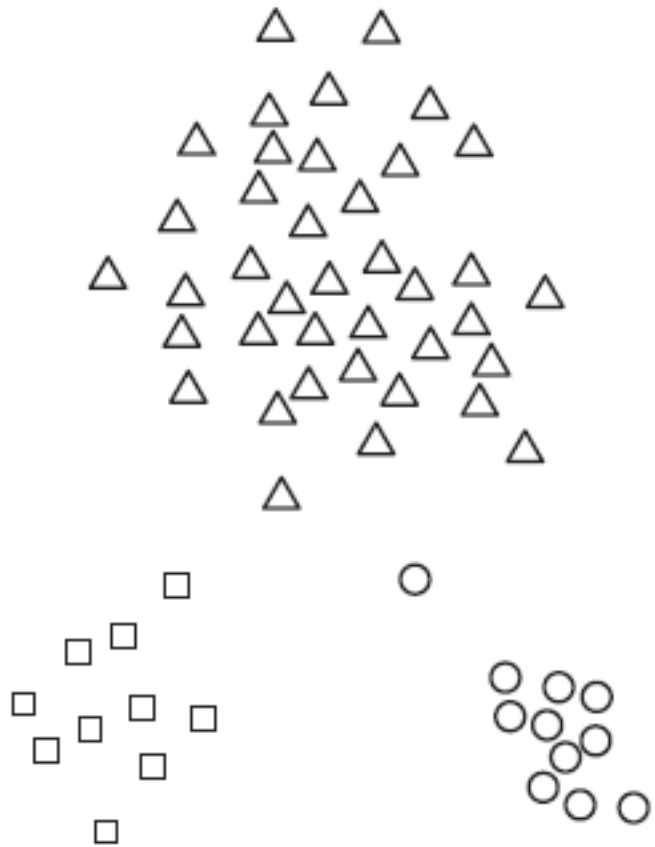
- Escolhendo centróides iniciais
  - O algoritmo do K-Means é bastante sensível à escolha dos pontos iniciais. Exemplo:



Iteration 1. Iteration 2. Iteration 3. Iteration 4.

K-means

- Escolhendo centróides iniciais
  - O mínimo global para o exemplo anterior é:



K-means++

- K-means++ foi proposto para solucionar o problema de selecionar um bom conjunto de valores iniciais para o k-means
- Motivação: problemas fundamentais do k-means tradicional:
  - Encontrar a solução ótima é um problema NP-Completo
  - Sensível a outliers
  - O resultado fornecido pelo K-means pode ser ruim em relação à solução ótima
- Vantagens do K-means++
  - Melhora o tempo de execução do K-means
  - Melhora a qualidade do resultado do K-means
  - Resultados melhoram a medida que o número de clusters aumentam

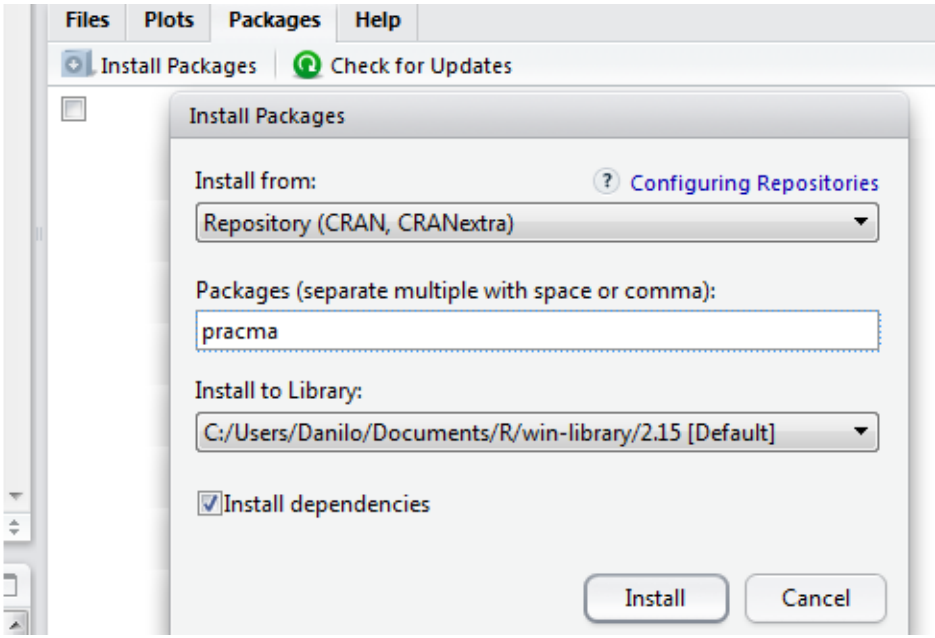
K-means++

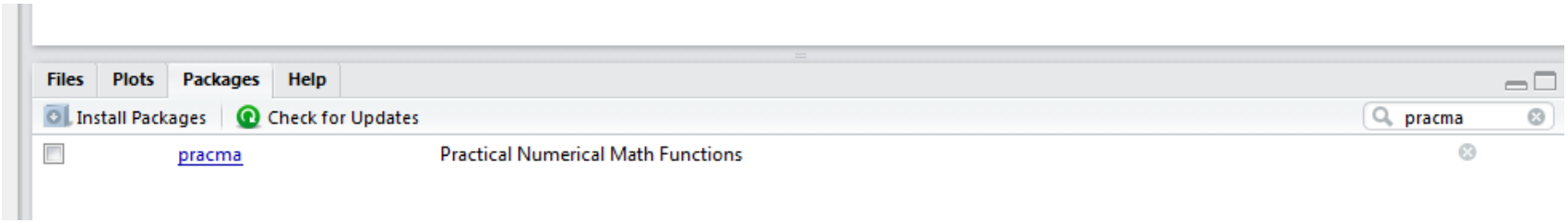
Algoritmo K-means++

1. Escolha um centróide uniformemente aleatoriamente entre os pontos de dados
2. Para cada ponto de dados x, calcule $D(x)$, a distância entre x e o centróide mais próximo que já tenha sido escolhido
3. Escolha um novo ponto de dados aleatoriamente como um novo centróide, usando uma distribuição de probabilidade ponderada onde um ponto x é escolhido com probabilidade proporcional a $D(x)^2$
4. Repita os passos 2 e 3 até que K centróides tenham sido escolhidos
5. Agora que os centróides iniciais foram escolhidos, continue usando o k-means tradicional

K-means++

- Implementação em R - Função *kmeanspp*, no pacote *pracma* - *Practical Numerical Math Functions*
- Instalando pacote *pracma* no R-Studio



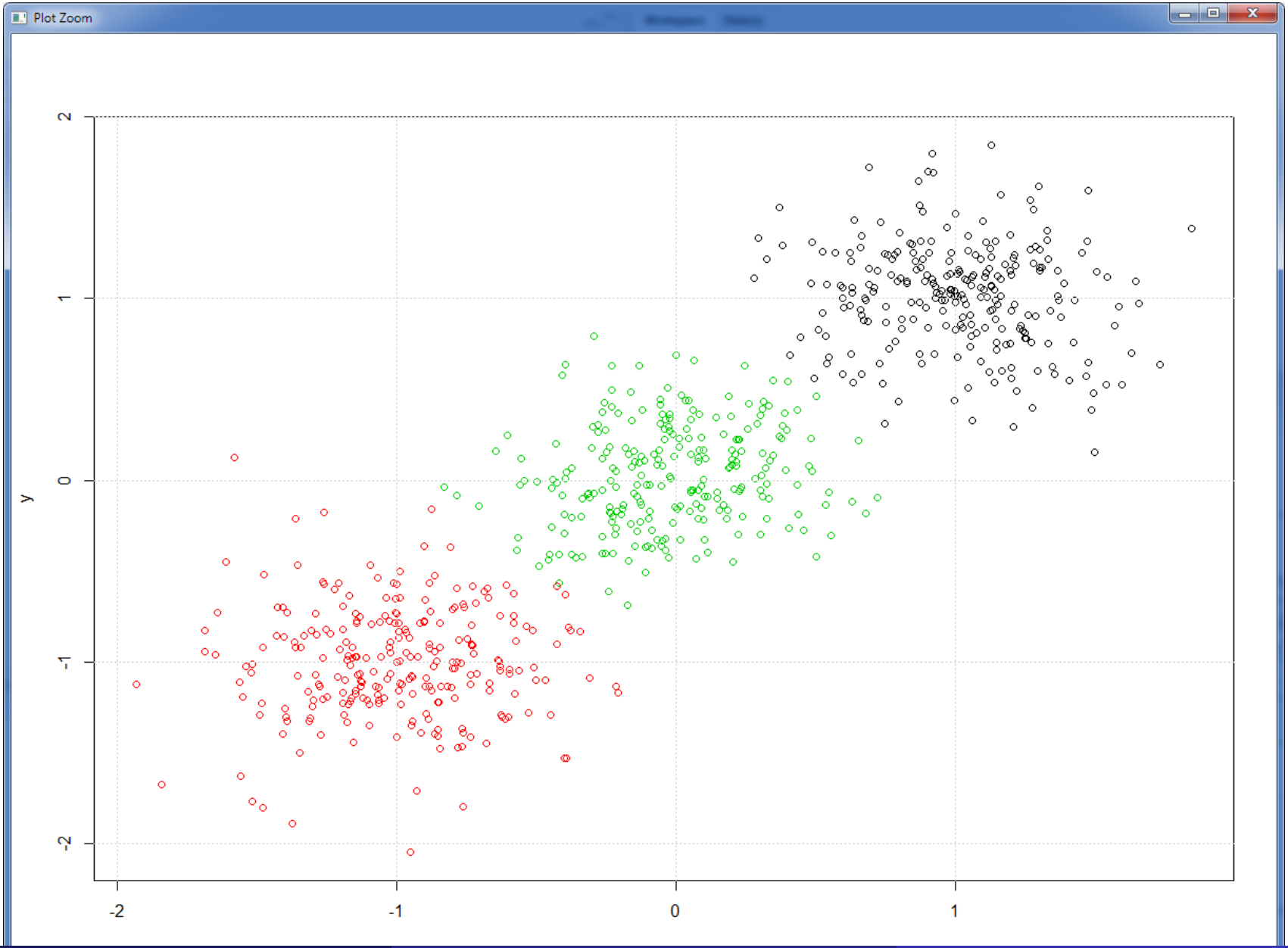


K-means++

Utilizando o k-means da API do R

```
X <- rbind(matrix(rnorm(500, mean = 0,  sd = 0.3), ncol = 2),
           matrix(rnorm(500, mean = 1,  sd = 0.3), ncol = 2),
           matrix(rnorm(500, mean = -1, sd = 0.3), ncol = 2))
colnames(X) <- c("x", "y")
cl <- kmeanspp(X, 3)
## Not run:
plot(X, col = cl$cluster)
points(cl$centers, col = 1:3)
grid()
## End(Not run)
```

K-means++



Final Message

Friends don't let friends use k-means.

Clustering na ferramenta WEKA

- WEKA - Waikato Environment for Knowledge Analysis
  - Pacote de software desenvolvido na Universidade de Waikato, Nova Zelândia, que oferece várias funcionalidades de data mining/aprendizagem de máquina
  - Vantagens de usar o WEKA para clustering:
    - O WEKA é capaz de lidar com objetos formados por uma mistura de atributos numéricos e categóricos
    - O algoritmo empregado pelo WEKA também normaliza os dados automaticamente quando computa a distância entre objetos

Clustering na ferramenta WEKA

- WEKA versus R

Clustering		
EM	√	√
KMeans	√	√
XMeans	√	
COBWEB (hierarchical)	√	
OPTICS	√	
Farthest first clustering	√	
Hierarchical clustering		√
Agglomerative nesting		√
Fuzzy C-means clustering		√
Bagged clustering		√
Cluster ensembles		√
Convex clustering		√

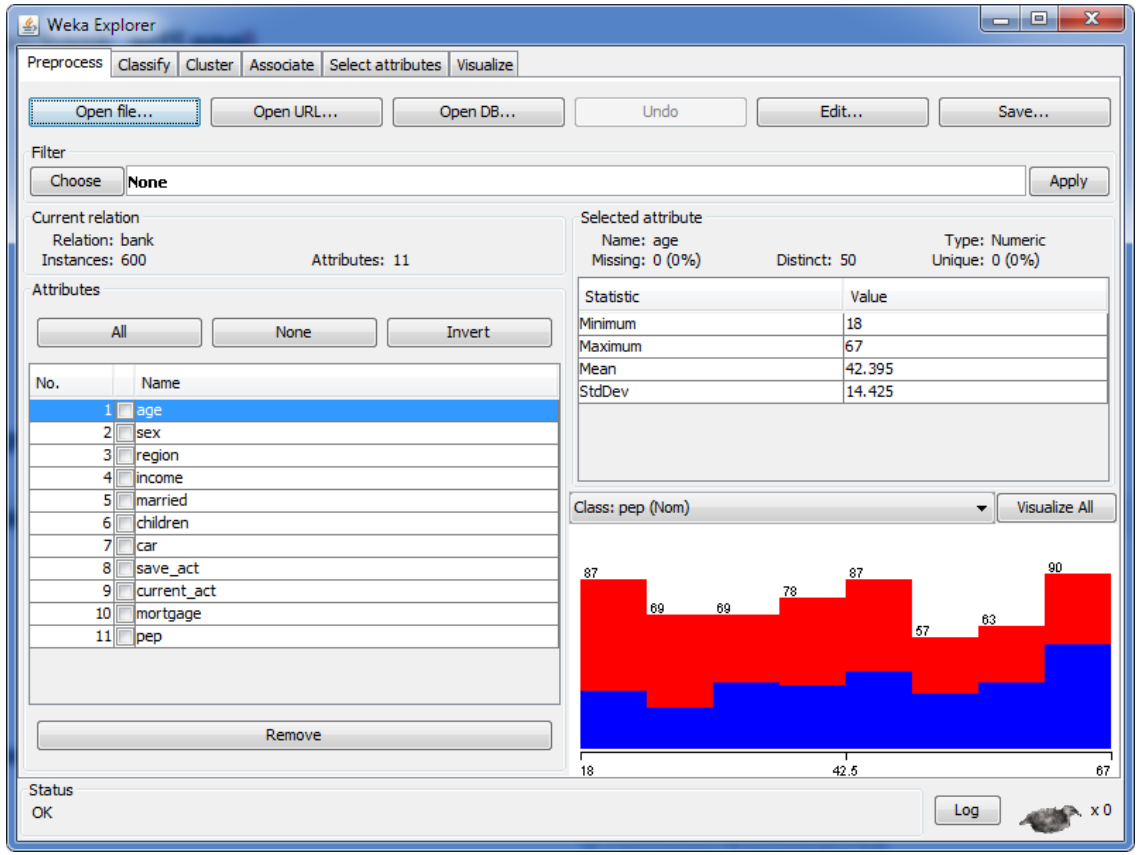
Clustering na ferramenta WEKA

- K-Means na ferramenta WEKA
  - Aceita como entrada um arquivo com extensão .arff, com o seguinte layout:

```
@relation bank

@attribute age numeric
@attribute sex {FEMALE,MALE}
@attribute region {INNER_CITY,TOWN,RURAL,SUBURBAN}
@attribute income numeric
@attribute married {NO,YES}
@attribute children {0,1,2,3}
@attribute car {NO,YES}
@attribute save_act {NO,YES}
@attribute current_act {NO,YES}
@attribute mortgage {NO,YES}
@attribute pep {YES,NO}

@data

48,FEMALE,INNER_CITY,17546,NO,1,NO,NO,NO,NO,YES
40,MALE,TOWN,30085.1,YES,3,YES,NO,YES,YES,NO
51,FEMALE,INNER_CITY,16575.4,YES,0,YES,YES,YES,NO,NO
23,FEMALE,TOWN,20375.4,YES,3,NO,NO,YES,NO,NO
57,FEMALE,RURAL,50576.3,YES,0,NO,YES,NO,NO,NO
57,FEMALE,TOWN,37869.6,YES,2,NO,YES,YES,NO,YES
22,MALE,RURAL,8877.07,NO,0,NO,NO,YES,NO,YES
```

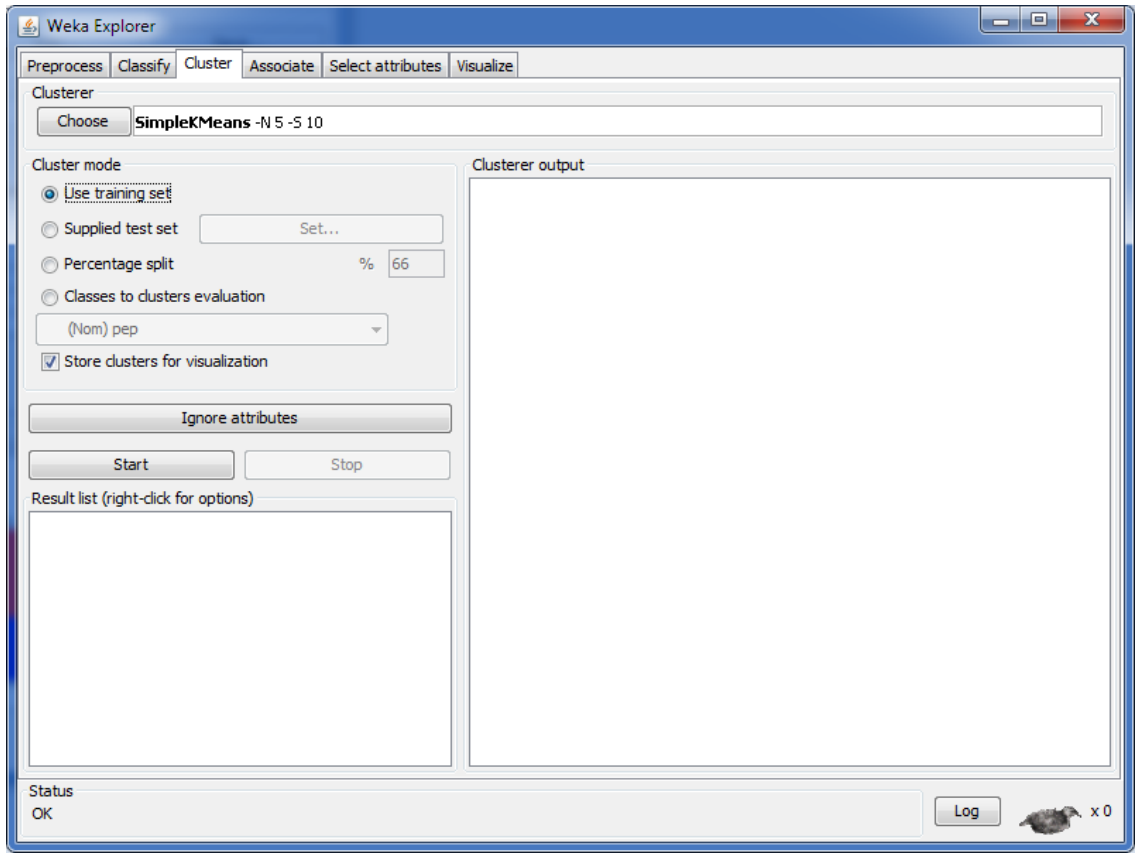
Clustering na ferramenta WEKA

- K-Means na ferramenta WEKA
  - Abrindo a seção "Explorer"da tela inicial do Weka, carregamos o arquivo "bank.arff"



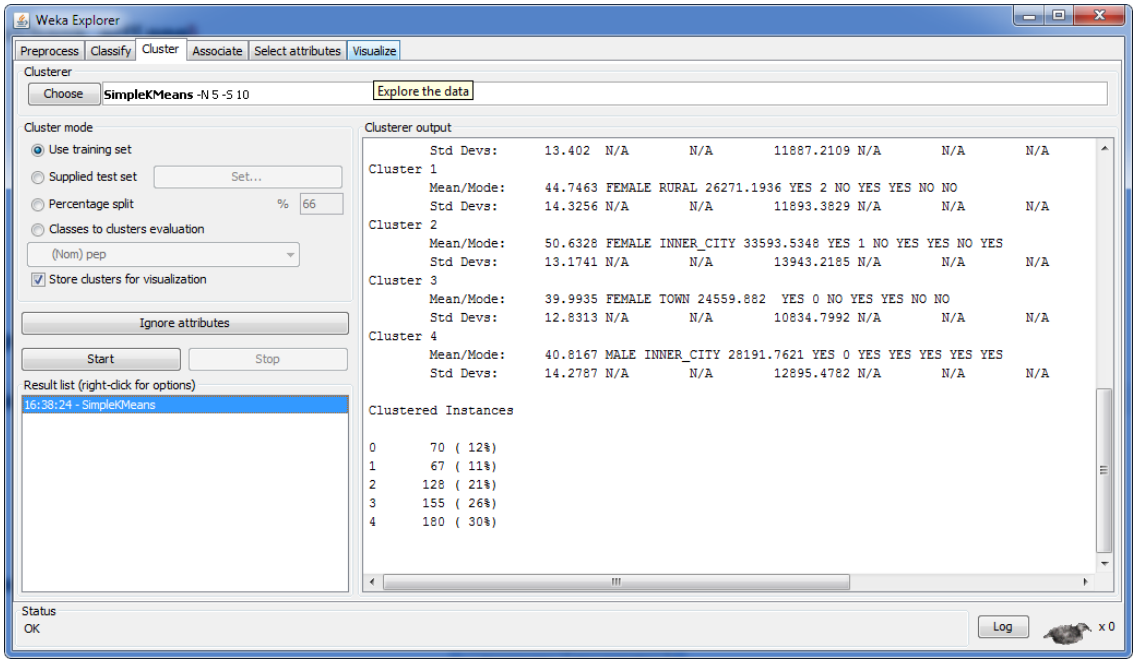
Clustering na ferramenta WEKA

- K-Means na ferramenta WEKA
  - Na aba "Cluster"selecionamos o algoritmo de clusterização (SampleKMeans) e o número de clusters (5)



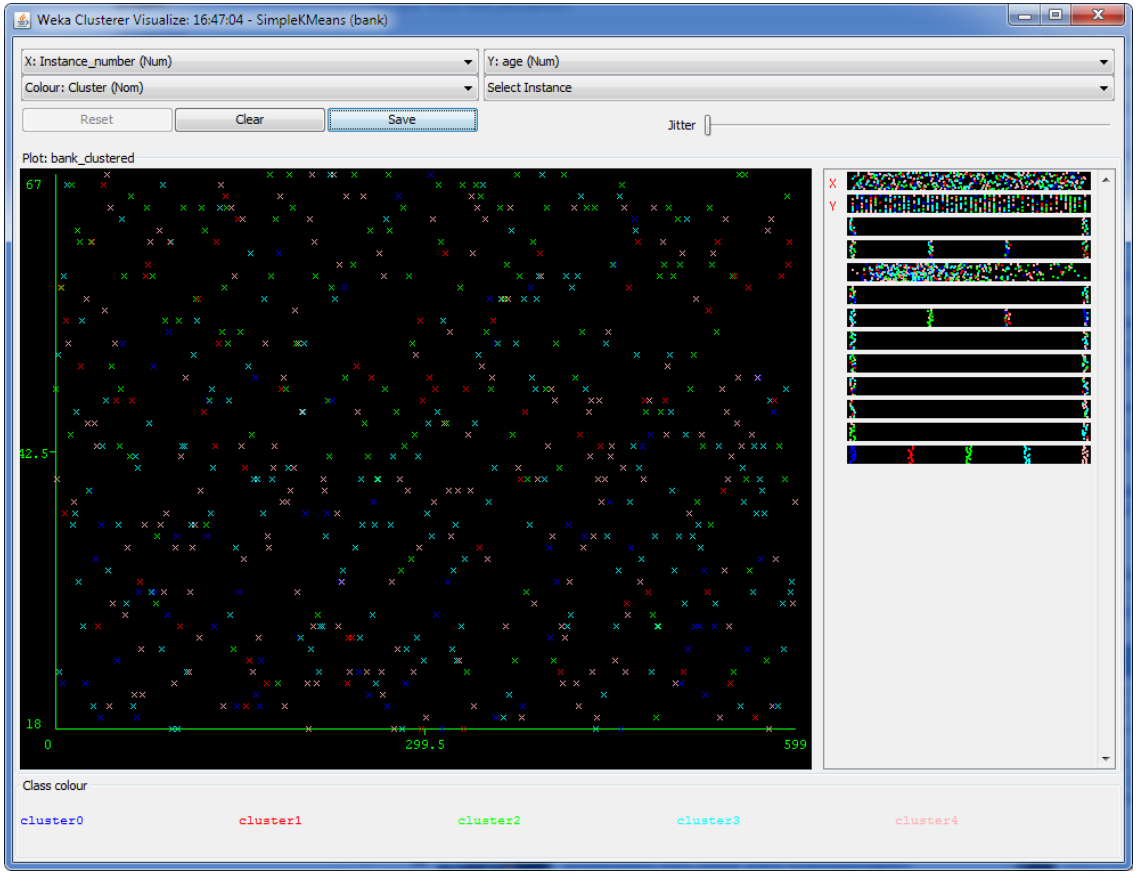
Clustering na ferramenta WEKA

- K-Means na ferramenta WEKA
  - Executando o algoritmo e visualisando o sumário dos resultados



Clustering na ferramenta WEKA

- K-Means na ferramenta WEKA
  - Visualisando os resultados



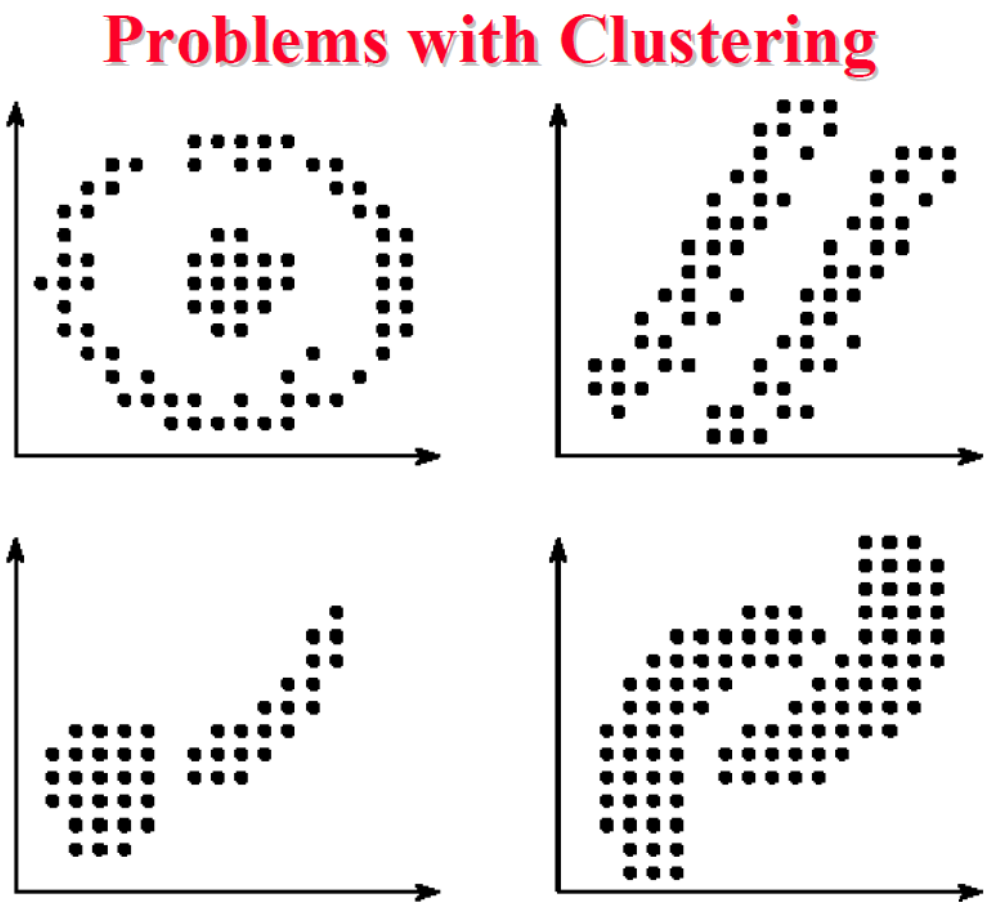
Clustering na ferramenta WEKA

- K-Means na ferramenta WEKA
  - Resultados salvos

```
@relation bank_clustered

@attribute Instance_number numeric
@attribute age numeric
@attribute sex {FEMALE,MALE}
@attribute region {INNER_CITY,TOWN,RURAL,SUBURBAN}
@attribute income numeric
@attribute married {NO,YES}
@attribute children {0,1,2,3}
@attribute car {NO,YES}
@attribute save_act {NO,YES}
@attribute current_act {NO,YES}
@attribute mortgage {NO,YES}
@attribute pep {YES,NO}
@attribute Cluster {cluster0,cluster1,cluster2,cluster3,cluster4}

@data
0,48,FEMALE,INNER_CITY,17546,NO,1,NO,NO,NO,NO,YES,cluster2
1,40,MALE,TOWN,30085.1,YES,3,YES,NO,YES,YES,NO,cluster4
2,51,FEMALE,INNER_CITY,16575.4,YES,0,YES,YES,YES,NO,NO,cluster3
3,23,FEMALE,TOWN,20375.4,YES,3,NO,NO,YES,NO,NO,cluster3
4,57,FEMALE,RURAL,50576.3,YES,0,NO,YES,NO,NO,NO,cluster1
5,57,FEMALE,TOWN,37869.6,YES,2,NO,YES,YES,NO,YES,cluster2
6,22,MALE,RURAL,8877.07,NO,0,NO,NO,YES,NO,YES,cluster0
7,58,MALE,TOWN,24946.6,YES,0,YES,YES,YES,NO,NO,cluster3
8,37,FEMALE,SUBURBAN,25304.3,YES,2,YES,NO,NO,NO,NO,cluster1
```

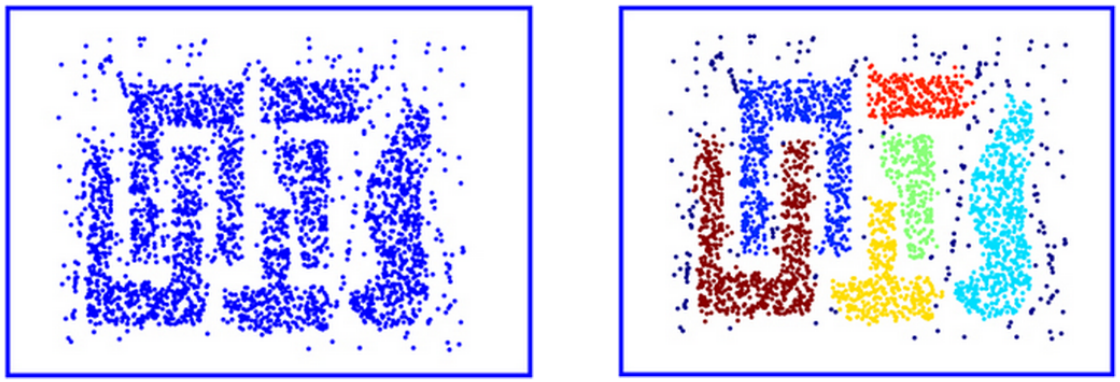
Sumario

Clustering baseado em densidade



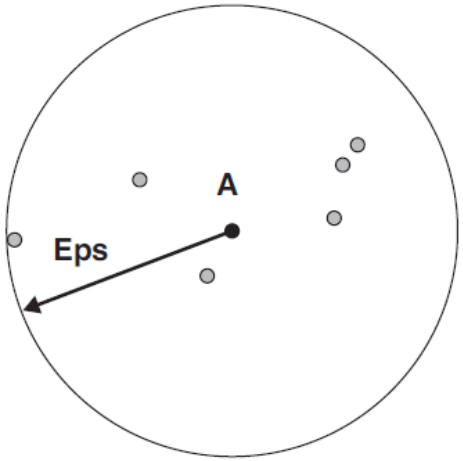
Clustering baseado em densidade

- Métodos de clustering baseados em densidade visam localizar regiões de alta densidade separadas umas das outras por regiões da baixa densidade.



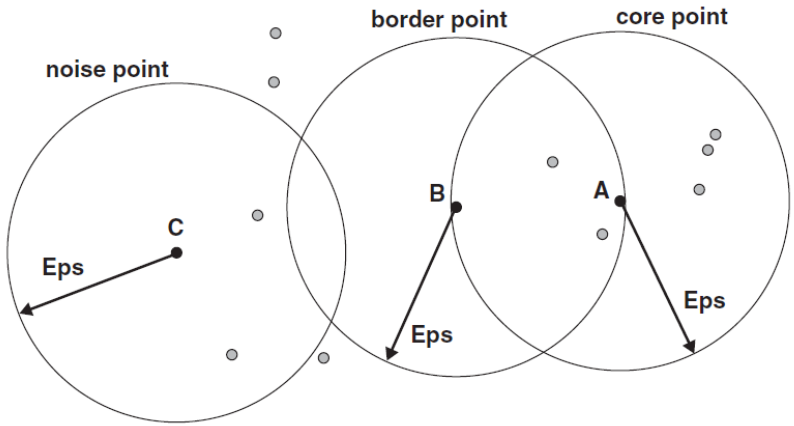
DBSCAN

- DBSCAN é um algoritmo que se baseia no número de pontos dentro de um determinado raio como métrica de densidade de um ponto



DBSCAN

- Dependendo da densidade do ponto, ele pode ser classificado como
  - **Core point** - É um ponto cujo número de pontos na sua vizinhança - região delimitada pela distância **Eps** - ultrapassa um determinado limite denominado **MinsPts**
  - **Border point** - É um ponto que nao se qualifica como core point, mas cai na vizinhança de algum core point
  - **Noise point** - É um ponto que não se qualifica nem como core point, nem como border point



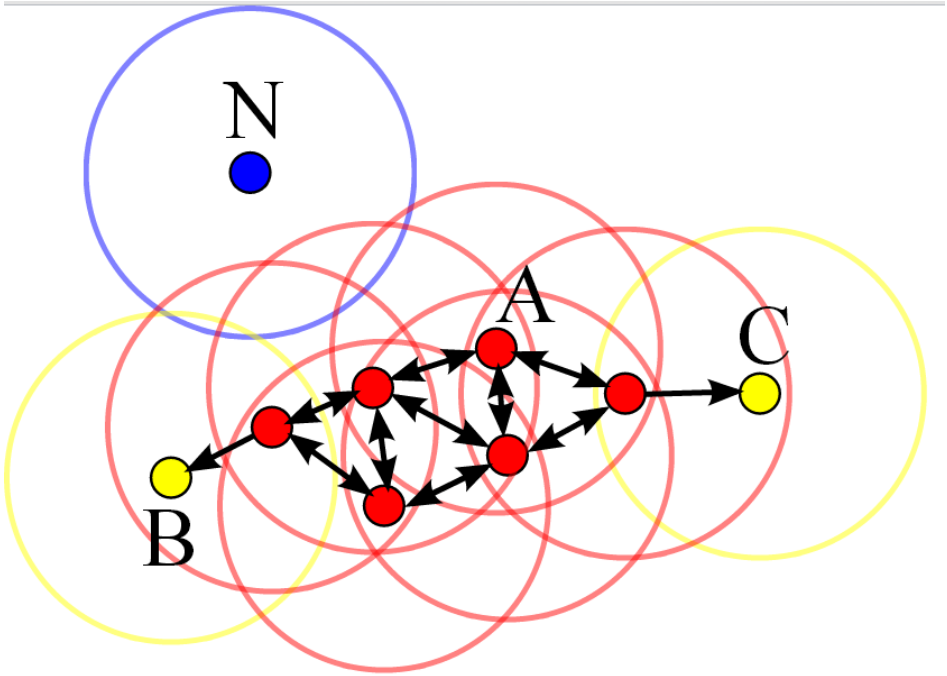
DBSCAN

Algoritmo DBSCAN

1. Rotule todos os pontos como “core”, “border” ou “noise”
2. Elimine todos os pontos “noise”
3. Coloque uma aresta entre todos os pontos core que estão dentro de um raio dado por *Eps* entre si
4. Separe cada grupo de componentes conectados em clusters separados
5. Atribua cada ponto “border” para o cluster dos seus pontos “core” associados

DBSCAN

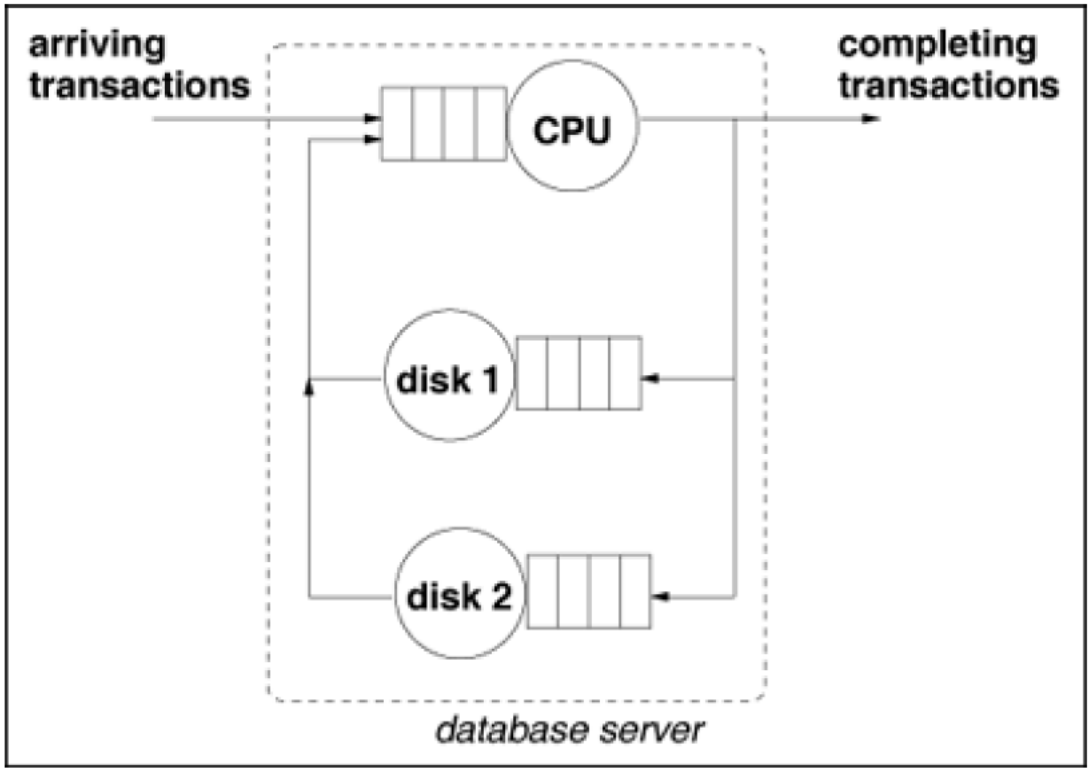
DBSCAN ilustrado:



Sumario

Atividade prática

- Considere um sistema de banco de dados com 1 CPU e dois discos



Atividade prática

- Através das ferramentas DBMS Performance Monitor e OS Performance Monitor nós caracterizamos os parâmetros do workload
- A ferramenta DBMS fornece um log de atividade de cada recurso para cada transação, conforme indicado na tabela abaixo:

Transaction Id	CPU Time (msec)	Disk 1 I/O Count	Disk 2 I/O Count
.	.	.	.
.	.	.	.
005	25.4	12	21
006	32.8	18	15
007	107.6	36	10
.	.	.	.
.	.	.	.

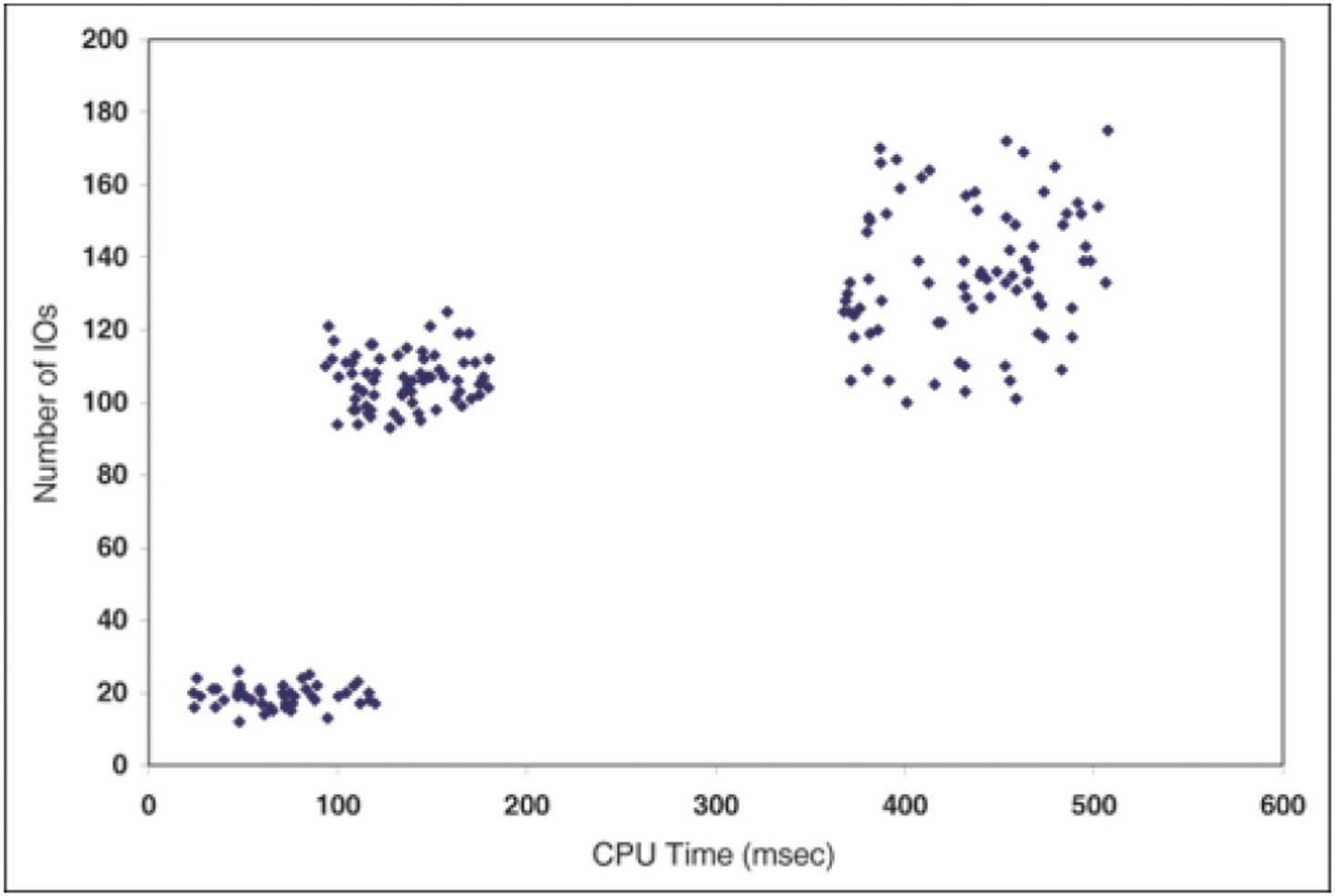
Atividade prática

- Através de uma análise exploratória de dados inicial, obtemos as seguintes estatísticas:

	CPU Time (msec)	*No. I/Os Disk 1*	*No. I/Os Disk 2*
Mean	238.2	51.38	44.85
Standard Deviation	165.9	27.0	26.4
Sample Variance	27510.4	728.7	698.1
Coeff. of Variation	0.696	0.525	0.677
Minimum	23.6	5	7
First Quartile (Q1)	104.4	33	26
Median (Q2)	151.6	63	39
Third Quartile (Q3)	418.1	72	68
Maximum	507.5	85	92
Range	483.9	80	85
Largest	507.5	85	92
Smallest	23.60	5	7
Sum	47640.8	10275	8969

Atividade prática

- Plotando um X-Y scatter plot mostrando o número de IOs (em ambos os discos) versus tempo de CPU:



Atividade prática

- Atividade:
  - Aplicar um método de clustering nos dados
  - Calcular as mesmas estatísticas (em especial o CV) para cada cluster separadamente

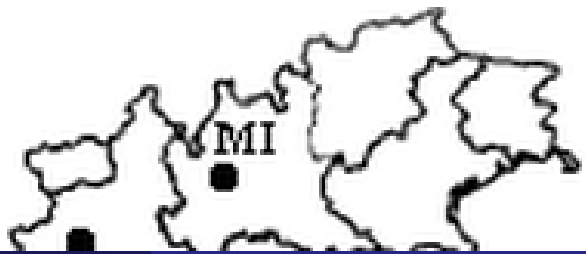
Prática

- Distância entre cidades da Itália.
- Com os dados abaixo, aferir utilizando cluster aglomerativo as cidades que estão mais próximas entre si.

Prática

- Matriz de distâncias.

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0



Referências

- http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html
- http://maya.cs.depaul.edu/classes/ect584/weka/k-means.html

Referências

Zaiane, O.R et al

Zaiane, O.R, On data clustering analysis: scalability, constraints and validation. Edmonton Alberta, University of Alberta, 2003

Doni, M.V.

Doni, M.V., Análise de cluster: Métodos Hierárquicos e de particionamento. TCC, Universidade Presbiteriana Mackenzie, São Paulo, 2004

Anderberg, M.R.C.

Anderberg, M.R.C., Cluster analysis for applications. New York: Academic Press, 1973.

Kaufman,L; Rousseeuw, P. J.

Anderberg, M.R.C., Finding groups in data: an introduction to cluster analysis. New York: Wiley, 1990