

# ANÁLISE DE AGRUPAMENTOS

---

Prof. André Backes | @progdescomplicada

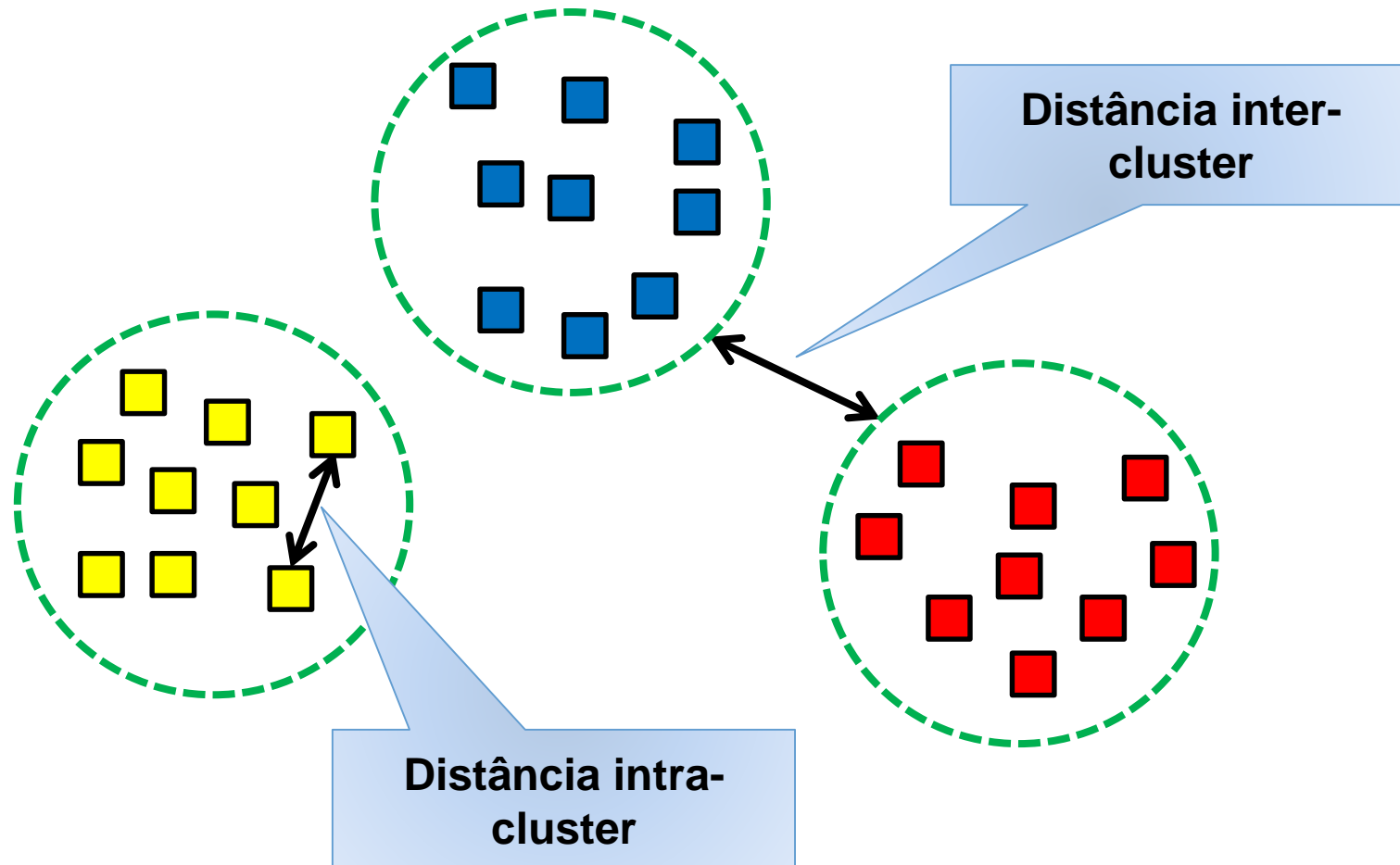
# Análise de Agrupamentos

- Definição
  - Consistem em encontrar grupos de objetos entre os objetos
    - Categorizá-los ou agrupá-los
  - Tipo de aprendizado não supervisionado
    - Encontrar grupos “naturais” de objetos para um conjunto de dados não rotulados

# Análise de Agrupamentos

- Definição
  - Os objetos de um grupo devem ser mais similares (ou relacionados) entre si do que a objetos de outro grupo
    - A similaridade pode ser a distância
    - *Distance-based Clustering*
  - Minimizar distância intra-cluster
    - Distância entre elementos de um mesmo grupo
  - Maximizar distância inter-cluster
    - Distância entre elementos de grupos distintos

# Análise de Agrupamentos



# Análise de Agrupamentos

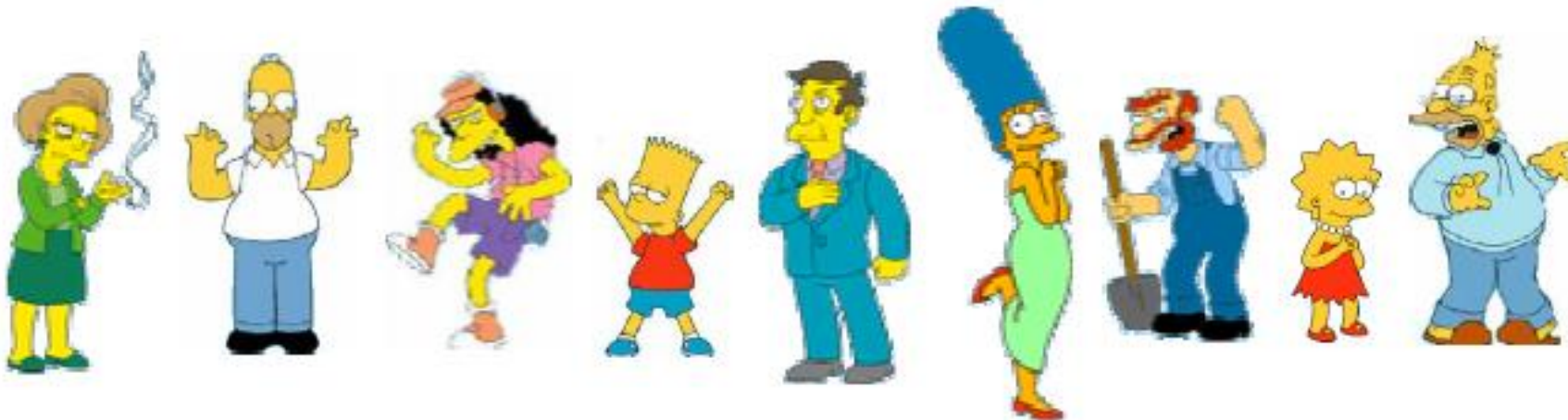
- Aplicações
  - Marketing
    - Grupos de clientes
    - Marketing direcionado
  - Biologia/Bioinformática
    - Encontrar grupos de genes com expressões semelhantes
    - Classificar grupos de plantas e animais
  - Mineração de Textos
    - Categorização de documentos
    - Classificação de páginas WWW
  - Etc.

# Classificação Vs Agrupamento

- Classificação
  - A partir de exemplos conhecidos (já classificados), aprender um método e usá-lo para predizer as classes de padrões desconhecidos (ou novos)
- Agrupamento (Clustering)
  - Dado um conjunto de dados não classificado, descobrir as classes dos elementos (grupos ou clusters) e possivelmente o número de grupos existente a partir de suas características

# O que é um cluster?

- Como organizar os dados observados em estruturas que façam sentido?
  - Afinal, o que é um agrupamento “natural” entre os seguintes objetos?



# O que é um cluster?

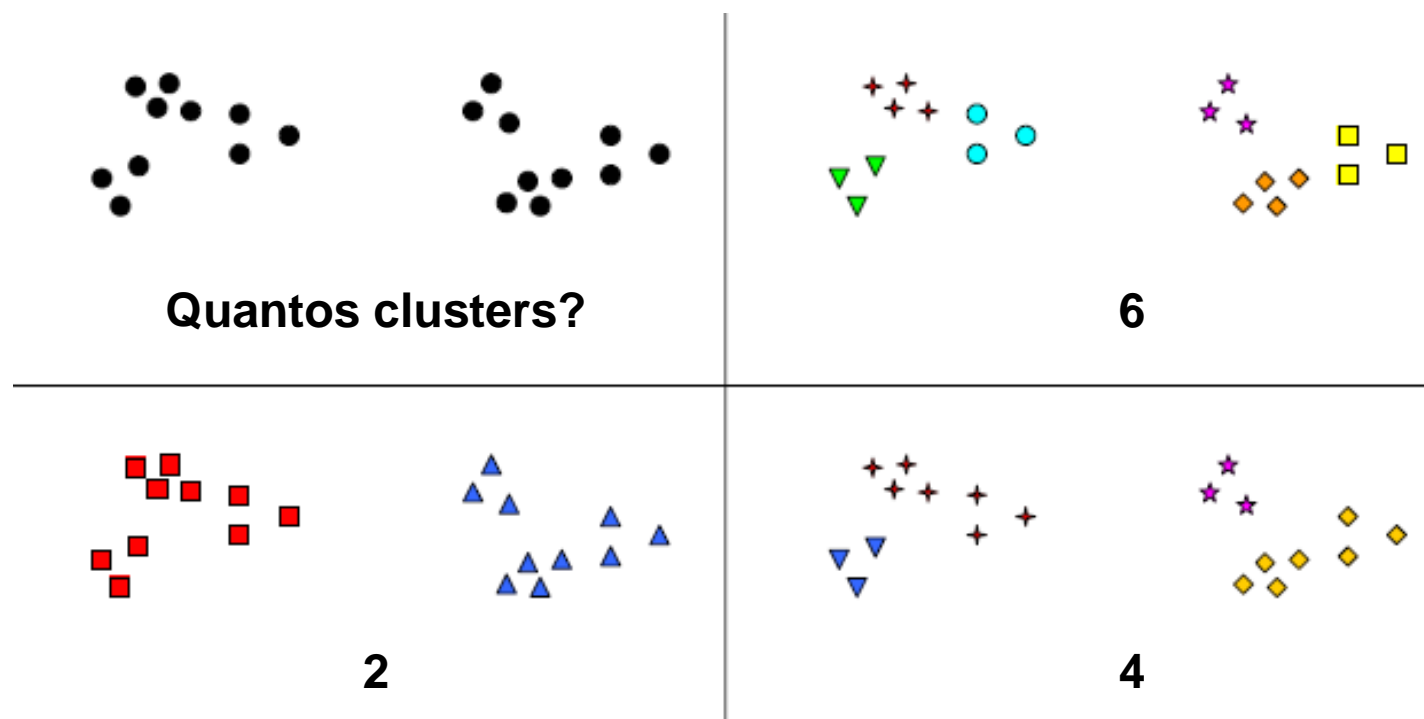
- Cluster é um conceito muito subjetivo
  - Processo de *data-driven*: agrupamento dirigido
    - Os dados observados são agrupados segundo características comuns que ocorram neles





# O que é um cluster?

- Além disso, a noção de cluster pode ser ambígua
  - Depende do número de clusters (muitas vezes definido pelo usuário)



# Definindo um cluster

- Cluster é um conceito muito subjetivo. Podemos defini-lo em termos de
  - Homogeneidade
    - Coesão interna
  - Heterogeneidade
    - Separação entre grupos
  - Necessidade de formalizar matematicamente, e para isso existem diversas medidas
    - Cada uma induz (impõe) uma estrutura aos dados
    - Geralmente baseadas em algum tipo de **(dis)similaridade**

# Como definir o que é similar ou não?

- Similar é diferente de igual!
  - Medida de semelhança



# Como definir o que é similar ou não?

- Usar uma medida de similaridade
  - Muitas vezes, esta é uma medida de distância
- Existem diversas possíveis
  - Minkowski
    - Manhattan
    - Euclideana
    - Chebyshev
  - Mahalanobis
  - Cosseno
  - Etc.

# Propriedades da medida de similaridade

- As propriedades que definem uma medida de similaridade são 3
  - $d(x,y) = d(y,x)$ , simetria
  - $d(x,y) \geq 0$
  - $d(x,x) = 0$
- Além dessas 3 propriedades, também valem
  - $d(x,y) = 0$ , se e somente se  $x = y$
  - $d(x,y) \leq d(x,z) + d(z,y)$ , também conhecida como desigualdade do triângulo

# Notações básicas

- Matriz de dados
  - Matriz contendo os dados de  **$N$**  objetos, cada qual com  **$p$**  atributos

- $$X = \begin{bmatrix} x_{11} & x_{12} \cdots & x_{1p} \\ x_{21} & x_{22} \cdots & x_{2p} \\ \cdots & \cdots \cdots & \cdots \\ x_{N1} & x_{Np} \cdots & x_{Np} \end{bmatrix}$$

- Cada objeto dessa matriz é denotado por um vetor  **$\mathbf{x}_i$** 
  - $x_i = [x_{i1} \ x_{i2} \ \cdots \ x_{ip}]$

# Notações básicas

- Matriz de (dis)similaridade

- Matriz  **$N \times N$**  contendo as distâncias entre os  **$N$**  objetos

- $$X = \begin{bmatrix} d(x_1, x_1) & d(x_1, x_2) & \dots & d(x_1, x_N) \\ d(x_2, x_1) & d(x_2, x_2) & \dots & d(x_2, x_N) \\ \dots & \dots & \dots & \dots \\ d(x_N, x_1) & d(x_N, x_2) & \dots & d(x_N, x_N) \end{bmatrix}$$

- É uma matriz simétrica em relação a sua diagonal principal
- Diagonal principal composta por 0's

# Qual abordagem de Clustering usar?

- Existem diversos métodos/algoritmos voltados para diferentes aplicações
  - Dados numéricos e/ou simbólicos
  - Dados relacionais ou não relacionais
  - Para construir partições ou hierarquias de partições
    - Partição: conjunto de clusters que compreendem os dados
  - Partições mutuamente exclusivas ou sobrepostas



# Métodos Relacionais e Não Relacionais

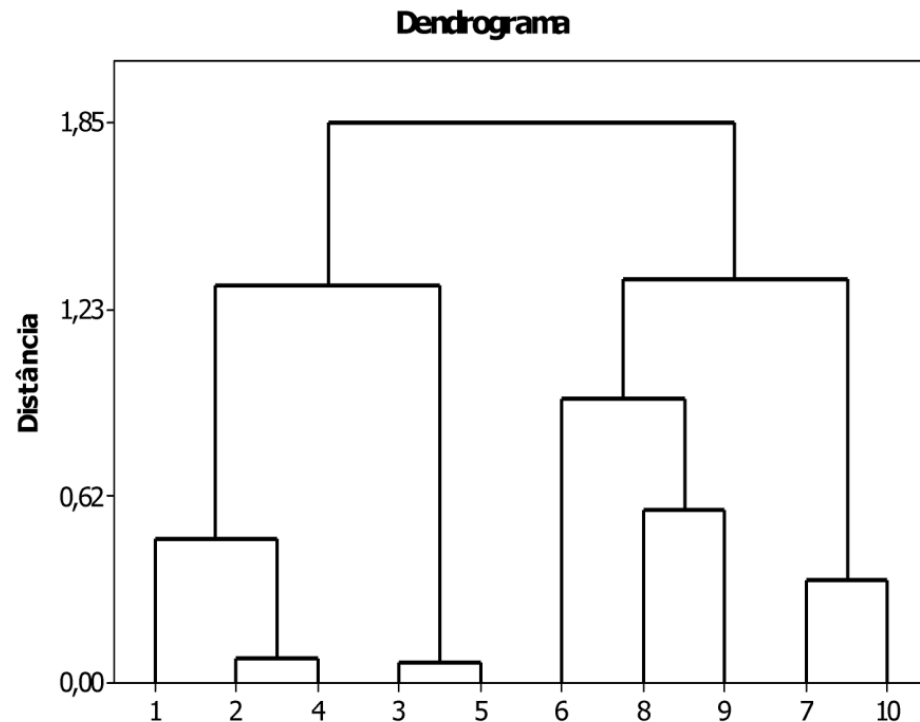
- Métodos Não Relacionais
  - Os dados não possuem nenhum tipo de relacionamento entre si
  - Utilizam apenas a matriz de dados  $X$  e uma medida de similaridade entre eles
- Métodos Relacionais
  - Se baseiam em uma relação de dependência entre os dados
    - Documentos: relação de ocorrência de palavras
    - Páginas de internet: links entre elas
    - Pode ser uma relação de dependência probabilística

# Métodos Hierárquicos e Não-Hierárquicos

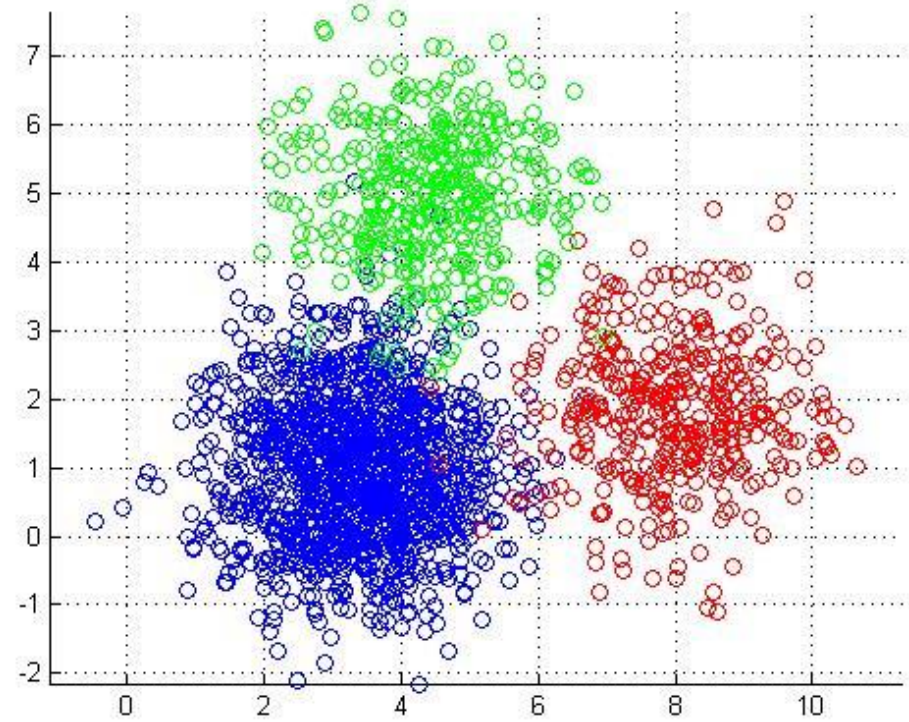
- Se referem principalmente a maneira como os dados são divididos e/ou organizados
  - Métodos Hierárquicos: constroem uma hierarquia de partições
  - Métodos Não-Hierárquicos ou Particionais: constroem uma partição dos dados

# Métodos Hierárquicos e Não-Hierárquicos

Método Hierárquico



Método Não-Hierárquico

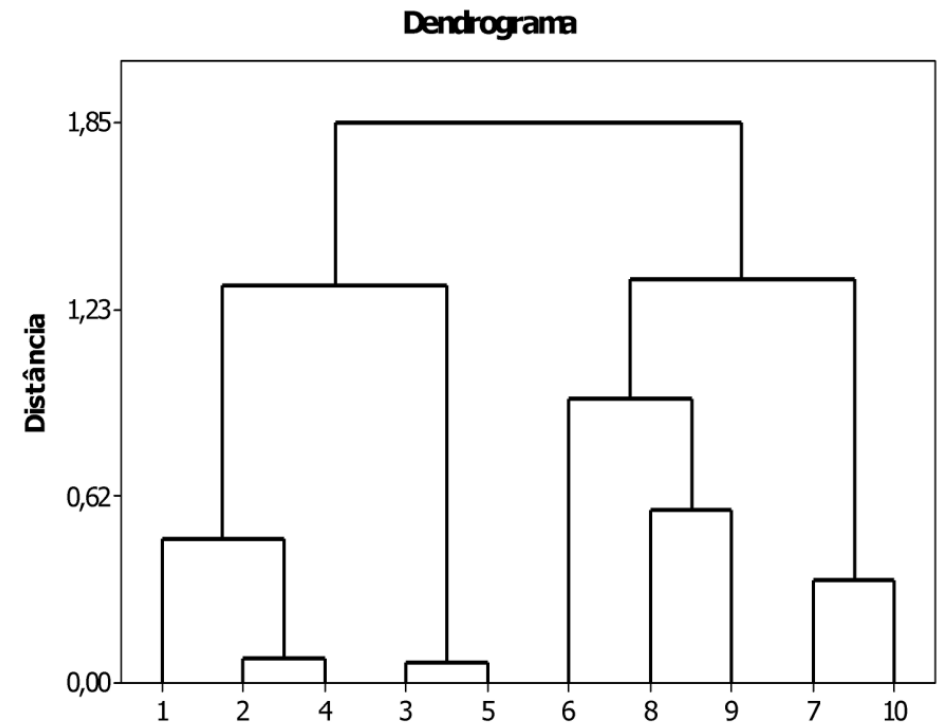


# Métodos Hierárquicos e Não-Hierárquicos

- Algoritmos Hierárquicos
  - Criam uma hierarquia de relacionamentos entre os elementos.
    - Uso de uma medida de distância
    - Muito populares na área de bioinformática
  - Bom funcionamento
    - Apesar de não terem nenhuma justificativa teórica baseada em estatística ou teoria da informação, constituindo uma técnica ad-hoc de alta efetividade.

# Métodos Hierárquicos e Não-Hierárquicos

- Algoritmos Hierárquicos
  - Dendrograma é talvez o algoritmo mais comum
    - Semelhante a uma árvore
    - Exemplo: relações evolutivas entre diferentes grupo de organismos biológicos (árvore filogenética)



# Métodos Hierárquicos e Não-Hierárquicos

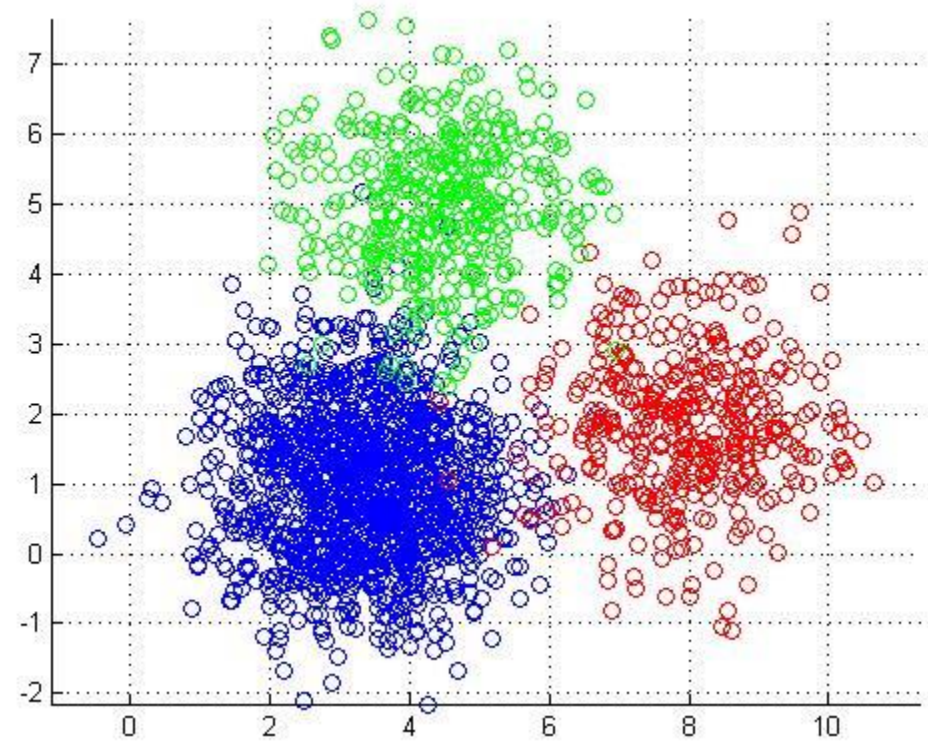
- Algoritmos Não-Hierárquicos
  - Separam os objetos em grupos baseando-se nas características que estes objetos possuem
    - Uso de uma medida de similaridade
  - Consistem de técnicas de análise de agrupamento ou *clustering*

# Métodos Hierárquicos e Não-Hierárquicos

- Algoritmos Não-Hierárquicos
  - Normalmente dependem de uma série de fatores que são determinados de forma arbitrária pelo usuário
    - Número de conjuntos
    - Número de seeds de cada conjunto.
    - Esses parâmetros podem causar impacto negativo na qualidade das partições geradas

# Métodos Hierárquicos e Não-Hierárquicos

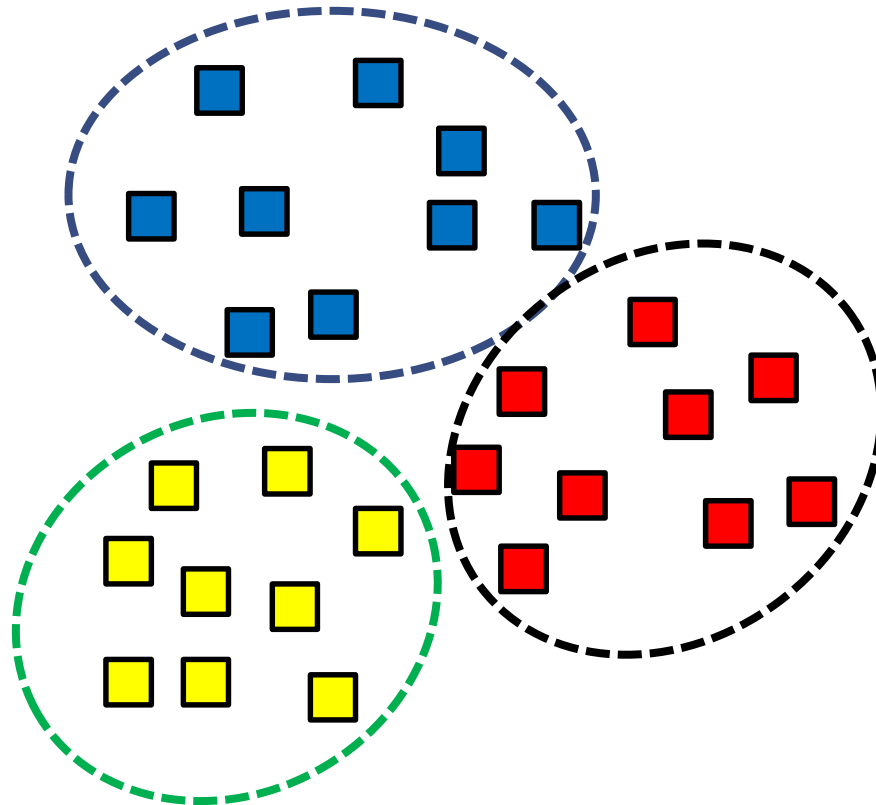
- Algoritmos Não-Hierárquicos
  - K-means é o algoritmo mais simples e mais comum
    - Busca particionar  $n$  observações em  $k$  clusters (agrupamentos)
    - Cada observação pertence ao cluster com a média mais próxima





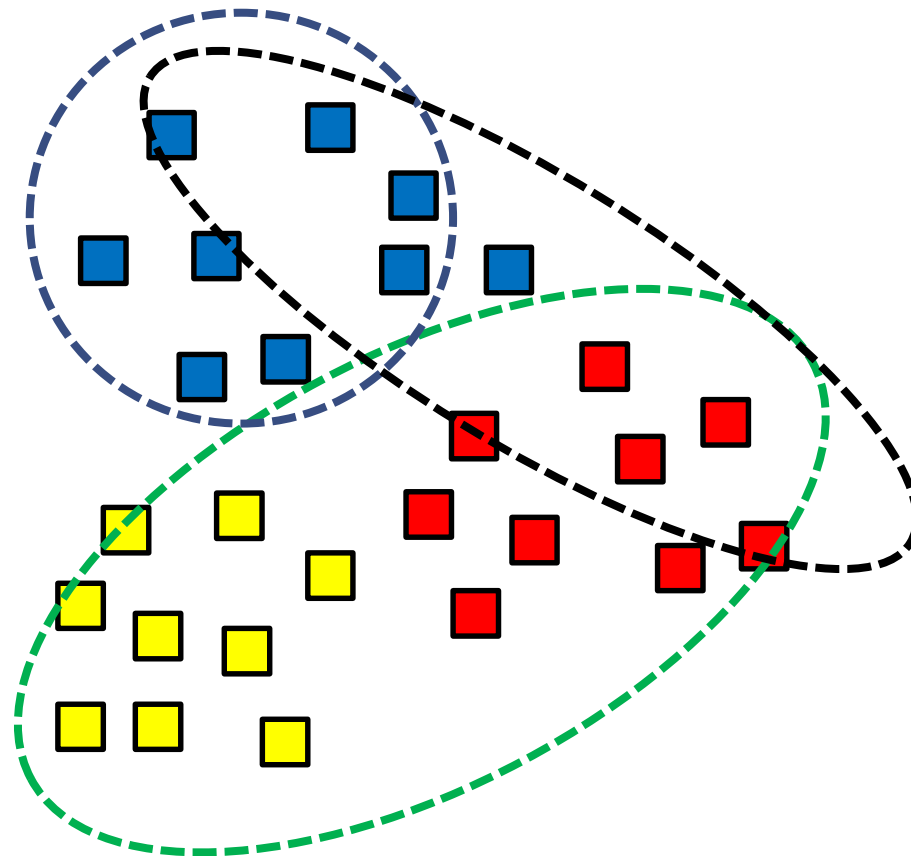
# Partições com ou sem sobreposição

- Partições sem sobreposição



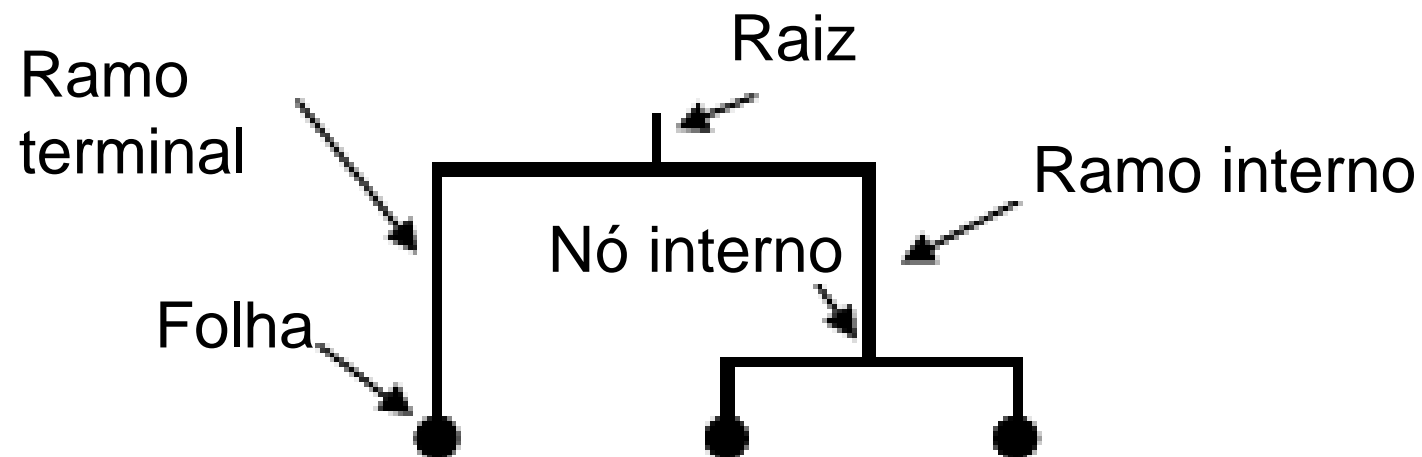
# Partições com ou sem sobreposição

- Partições com sobreposição



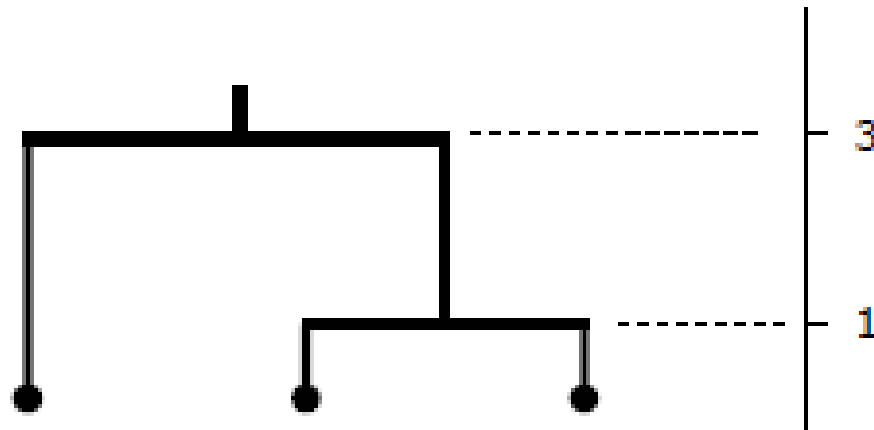
# Dendrograma

- Um dos algoritmos mais comuns para construir uma hierarquia de partições a partir das distâncias entre clusters



# Dendrograma

- A dissimilaridade entre dois clusters (possivelmente singletons, i.e., composto por apenas um elemento) é representada como a altura do nó interno mais baixo compartilhado

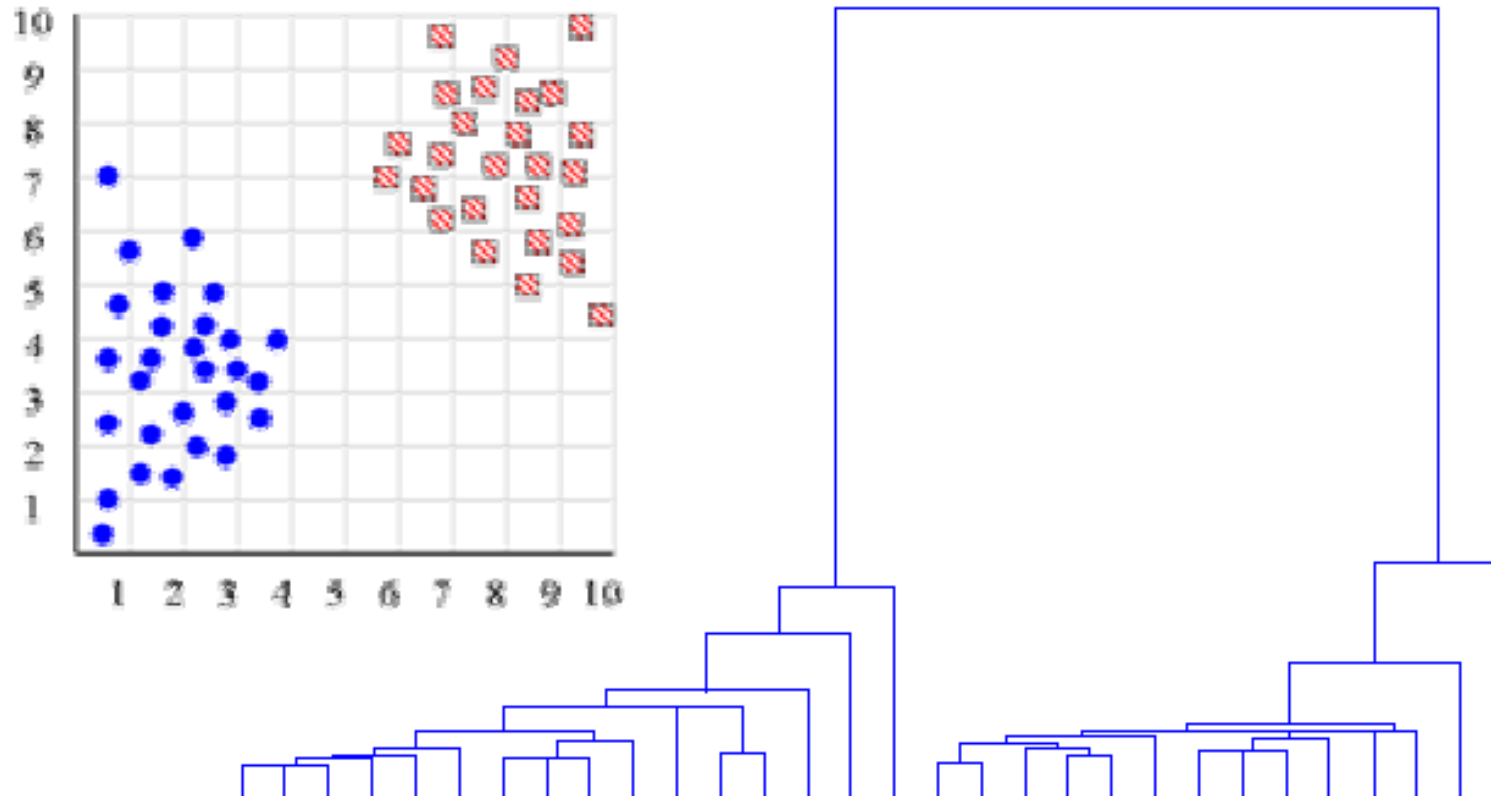


# Dendrograma

- A análise de um dendrograma permite estimar o número mais natural de clusters de um conjunto de dados
  - Sub-árvores bem separadas

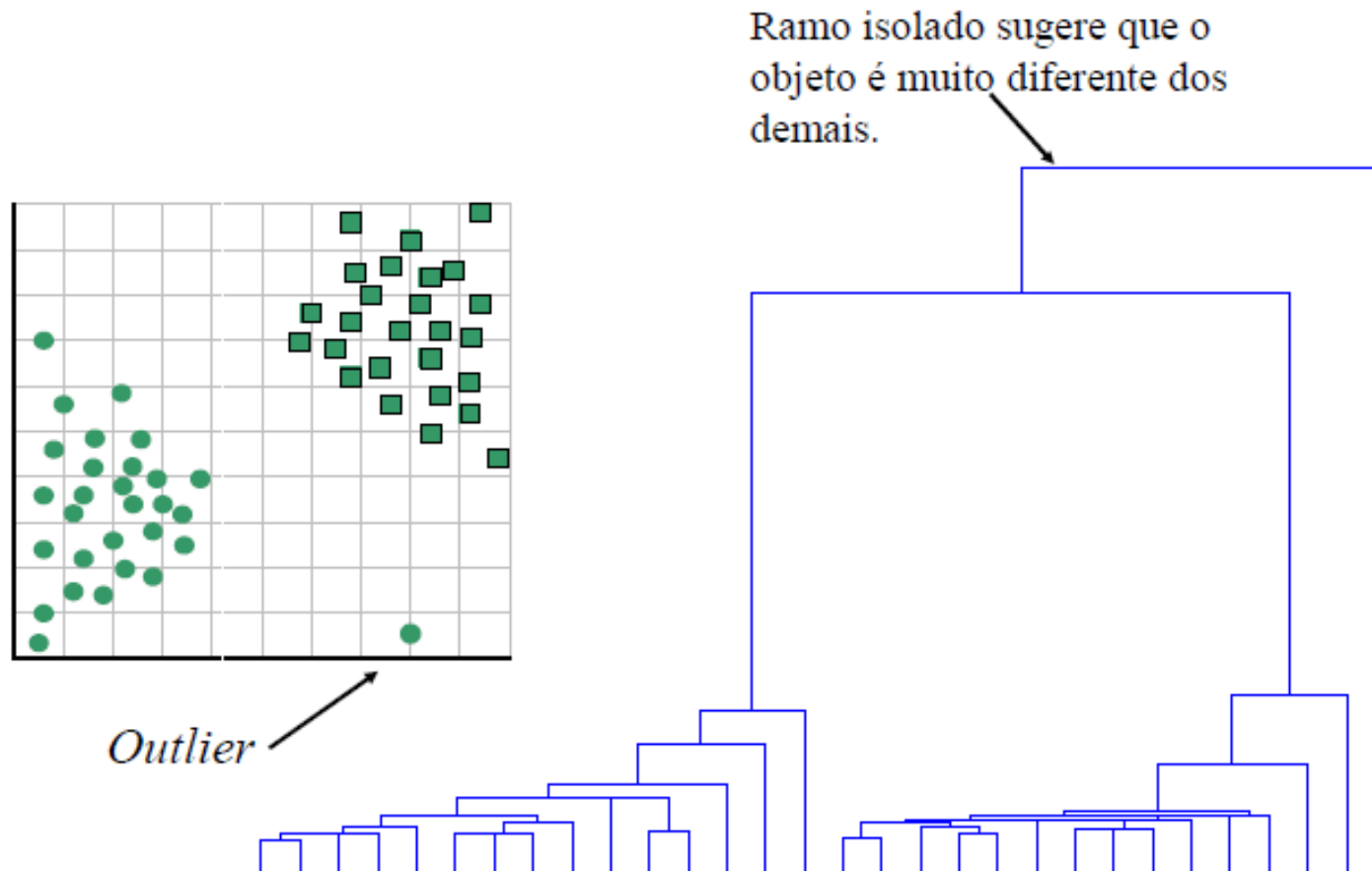
# Dendrograma

- Conjunto de dados: 2 clusters
  - Na prática, as distinções não são tão simples



# Dendrograma

- A análise de um dendrograma também permite detectar *outliers*



# Como criar uma hierarquia automaticamente?

- Abordagem Bottom-Up
  - Abordagem aglomerativa
  - Inicialmente, cada objeto é um cluster
    - Busca o melhor par de clusters para unir
    - Unir o par de clusters escolhido
    - Este processo é repetido até que todos os objetos estejam reunidos em um só cluster



# Como criar uma hierarquia automaticamente?

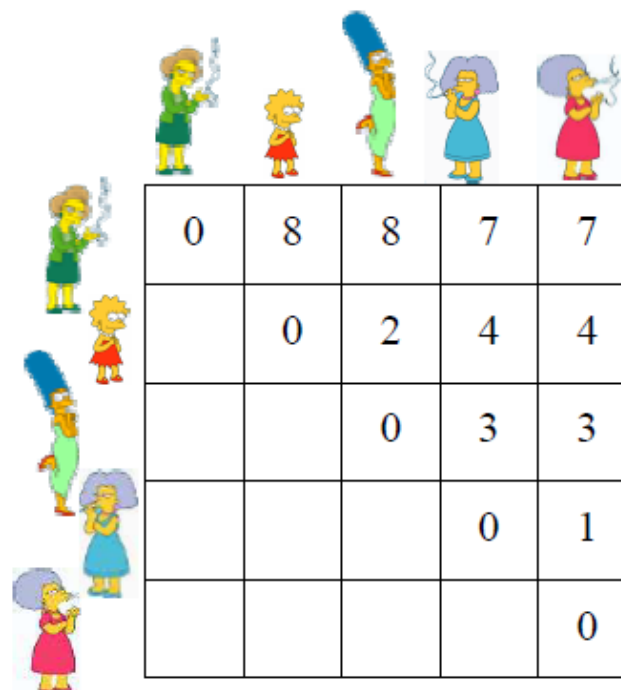
- Abordagem Top-Down
  - Abordagem divisiva
    - Inicialmente todos os objetos estão em um único cluster
    - Sub-dividir o cluster em dois novos clusters
    - Recursivamente aplicar o algoritmo em ambos os clusters, até que cada objeto forme um cluster por si só

# Como criar uma hierarquia automaticamente?

- Algoritmos hierárquicos podem operar somente sobre uma matriz de distâncias
  - Eles são (ou podem ser) relacionais


$$D(\text{Homer}, \text{Bart}) = 8$$


$$D(\text{Marge}, \text{Lisa}) = 1$$




	Homer	Bart	Marge	Lisa	Maggie
Homer	0	8	8	7	7
Bart		0	2	4	4
Marge			0	3	3
Lisa				0	1
Maggie					0










# Como criar uma hierarquia automaticamente?

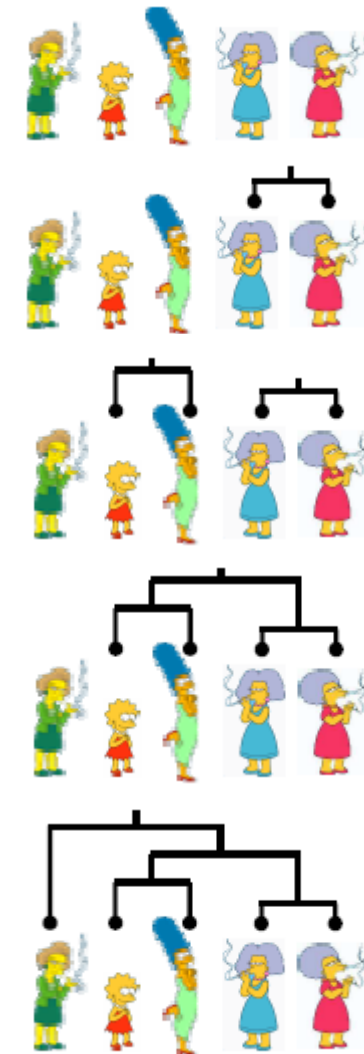
- Exemplo: abordagem Bottom-Up


$$D(\text{green}, \text{red}) = 8$$


$$D(\text{pink}, \text{blue}) = 1$$

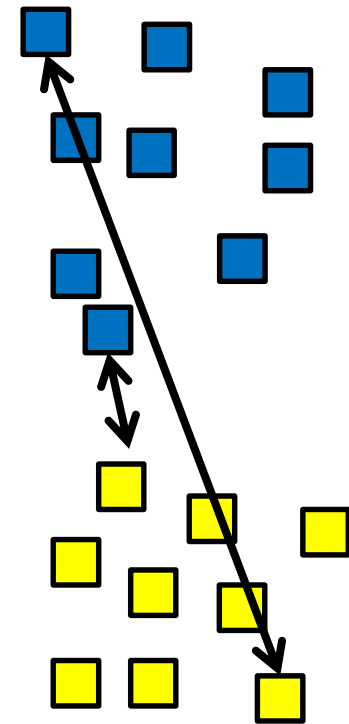


					
	0	8	8	7	7
		0	2	4	4
			0	3	3
				0	1
					0



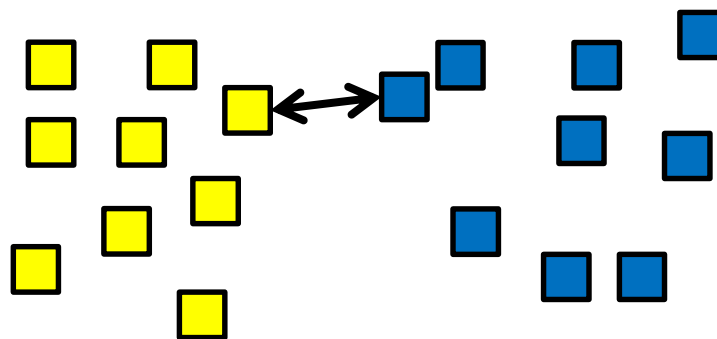
# Como medir a (dis)similaridade entre clusters?

- Eventualmente, um cluster terá mais de um elemento dentro dele. Neste caso, como medir a distância entre eles?
  - Várias possibilidades
    - Distância mínima
    - Distância máxima
    - Distância média do grupo
    - Distância entre centroides
    - Etc.



# Como medir a (dis)similaridade entre clusters?

- Método *Single Linkage*
  - Distância mínima ou Vizinho mais Próximo
    - Distância entre 2 clusters é dada pela menor distância entre dois objetos (um de cada cluster)



# Como medir a (dis)similaridade entre clusters?

- Exemplo: *Single Linkage* + Bottom-Up
  - Consideremos a seguinte matriz de distâncias iniciais ( $\mathbf{D}_1$ ) entre 5 objetos

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

- A menor distância entre objetos é  $\mathbf{d}_{12} = \mathbf{d}_{21} = 2$ 
  - Estes dois objetos serão unidos em um cluster

# Como medir a (dis)similaridade entre clusters?

- Exemplo: *Single Linkage* + Bottom-Up
  - Na sequência, devemos calcular a menor distância entre um objeto e um membro desse cluster

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

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

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

$$\mathbf{D}_2 = \begin{matrix} & 1 & 2 & 3 & 4 & 5 \\ \begin{matrix} 1 \\ 2 \end{matrix} & 0 & 5 & 9 & 8 & 12 \end{matrix}$$

- Isso resulta em uma nova matriz de distâncias ( $\mathbf{D}_2$ ), que será usada na próxima etapa do agrupamento hierárquico

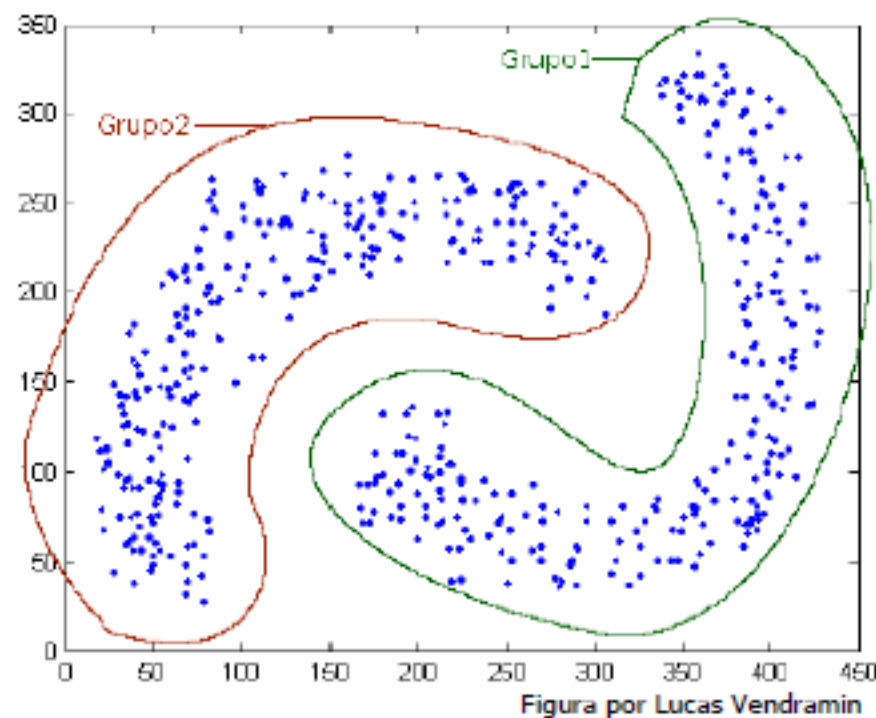
# Como medir a (dis)similaridade entre clusters?

- Método *Single Linkage*
  - A dissimilaridade entre 2 clusters pode ser computada naturalmente a partir da matriz atualizada na iteração anterior
  - Não há necessidade da matriz original dos dados



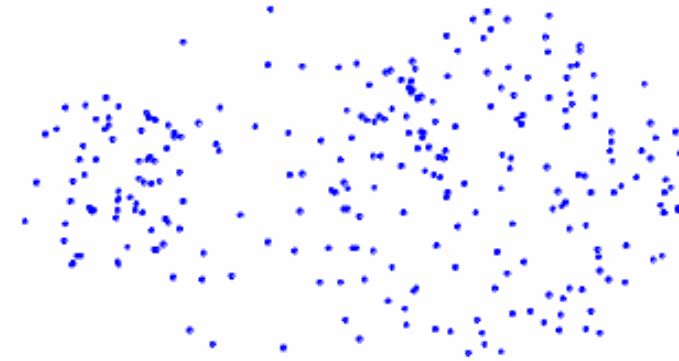
# Como medir a (dis)similaridade entre clusters?

- Método *Single Linkage*
  - Vantagens
    - Consegue manipular clusters que tenham uma forma não elíptica



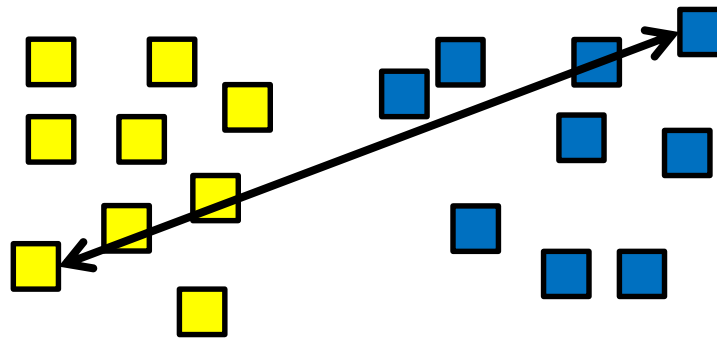
# Como medir a (dis)similaridade entre clusters?

- Método *Single Linkage*
  - Desvantagens
    - Muito sensível a ruídos e *outliers*



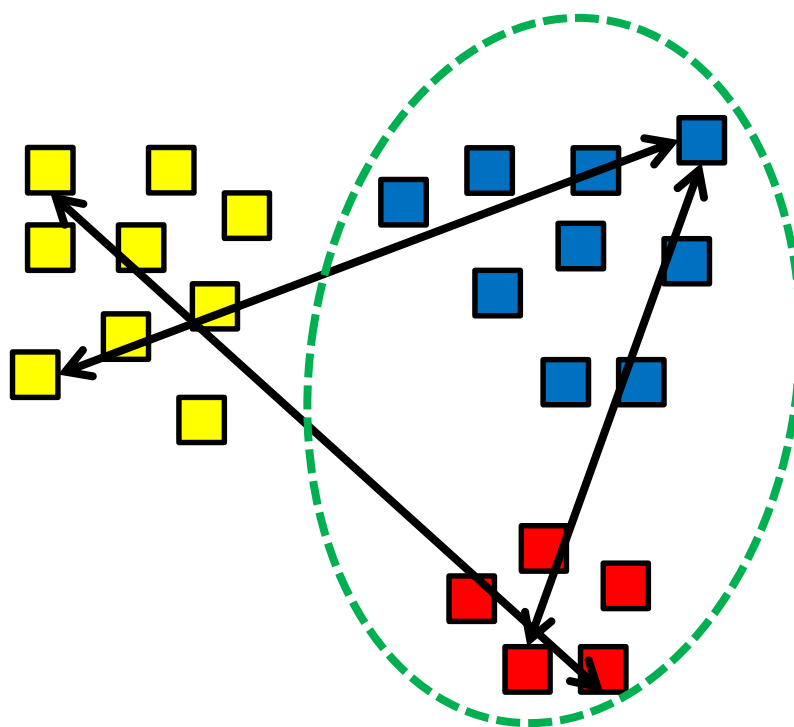
# Como medir a (dis)similaridade entre clusters?

- Método *Complete Linkage*
  - Distância máxima ou Vizinho mais Distante
    - Distância entre 2 clusters é dada pela maior distância entre dois objetos (um de cada cluster)



# Como medir a (dis)similaridade entre clusters?

- Método *Complete Linkage*
  - $D(X,Y)$ : maior distância entre dois objetos de cluster diferentes
  - Unir os dois clusters que possuem o menor valor de  $D$



# Como medir a (dis)similaridade entre clusters?

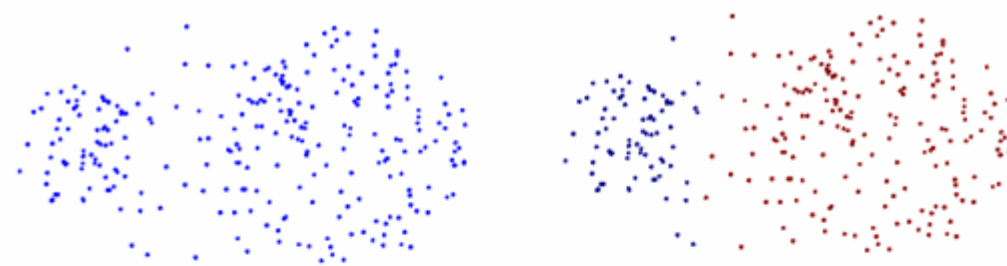
- Exemplo: *Complete Linkage* + Bottom-Up
  - A sequência do método é igual ao Single Linkage
    - Calcular a maior distância entre um objeto e um membro desse cluster
    - Obter a nova matriz de distâncias ( $\mathbf{D}_2$ ), que será usada na próxima etapa do agrupamento hierárquico
  - Como no método *Single Linkage*, a dissimilaridade entre 2 clusters pode ser computada naturalmente a partir da matriz atualizada na iteração anterior
    - Não há necessidade da matriz original dos dados

# Como medir a (dis)similaridade entre clusters?

- Método Complete *Linkage*

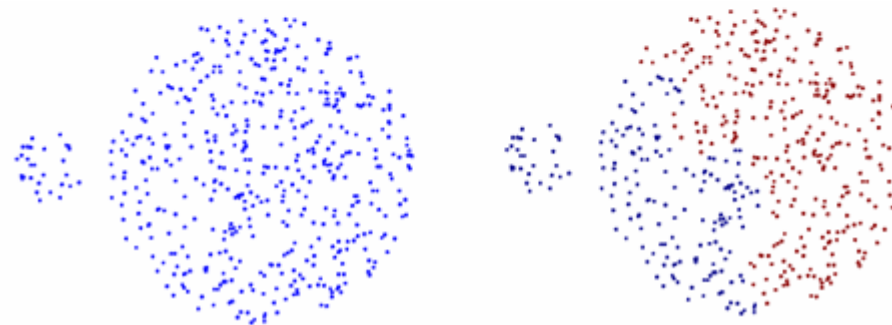
- Vantagens

- Menos sensível a ruídos e *outliers*



- Desvantagens

- Tende a quebrar clusters muito grandes



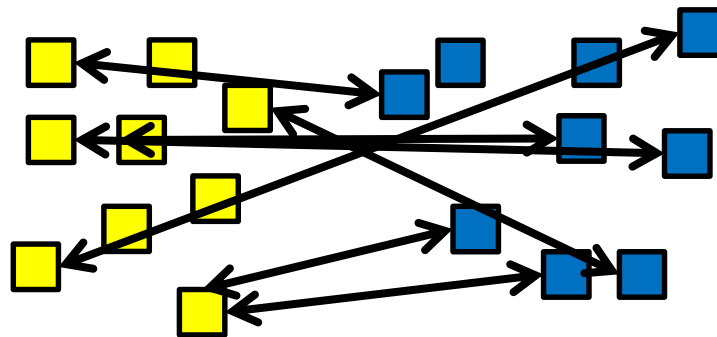
# Como medir a (dis)similaridade entre clusters?

- Método *Average Linkage*

- *Group average* ou Distância média

- Distância entre 2 clusters é dada pela média das distâncias entre cada dois objetos (um de cada cluster)
    - Média das distâncias de todos contra todos

- $$D(C_1, C_2) = \frac{\sum_{x_i \in C_1, x_j \in C_2} d(x_i, x_j)}{N_1 * N_2}$$



# Como medir a (dis)similaridade entre clusters?

- Método *Average Linkage*
  - Une características dos métodos *Single Linkage* e *Complete Linkage*
    - Menos sensível a ruídos e *outliers*
    - Propenso a clusters globulares
  - Atenção
    - O cálculo da dissimilaridade entre um novo cluster (dado pela união de outros dois) e os demais deve considerar o número de objetos em cada cluster envolvido

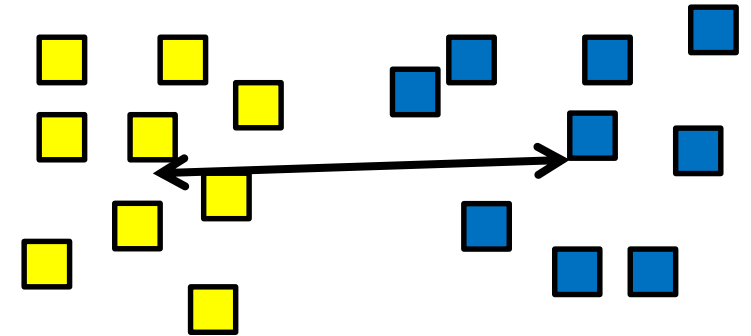


# Como medir a (dis)similaridade entre clusters?

- Comparação entre os métodos
  - *Single Linkage*
    - Detecta clusters convexos
    - Sensível a ruído ou *outliers*
  - *Complete Linkage*
    - Menos sensível a ruído ou *outliers*
    - Favorece clusters globulares
  - *Average Linkage*
    - Também favorece clusters globulares
    - Mas é muito menos sensível a ruídos e *outliers*

# Como medir a (dis)similaridade entre clusters?

- Existem ainda outras possibilidades, cada qual com as suas vantagens e desvantagens
  - Método dos Centróides
    - Usa distância entre os centróides (vetores de médias) dos clusters
  - Método dos Centróides Ponderado
  - Método de Ward



# Abordagem Top-Down

- São pouco utilizados
  - Em geral se usa a abordagem Bottom-Up
    - É mais fácil unir 2 clusters do que separá-los
  - Existem  $2^{N-1}-1$  possibilidades para se dividir  $N$  objetos em 2 clusters
    - Para  $N = 50$ , temos  $5,63 \times 10^{14}$  possibilidades
  - Diante disso, como dividir um cluster?
    - Uma possibilidade é usar a heurística de MacNaughton-Smith et al. (1964)

# Abordagem Top-Down

- Abordagem divisiva
  - Inicialmente todos os objetos estão em um único cluster
  - Sub-dividir o cluster em dois novos clusters
  - Recursivamente aplicar o algoritmo em ambos os clusters, até que cada objeto forme um cluster por si só

# Métodos Não-Hierárquicos: sem sobreposição

- Definição do problema
  - Particionar o conjunto  $X = \{x_1, x_2, \dots, x_N\}$  de objetos em uma coleção  $C = \{C_1, C_2, \dots, C_k\}$  de  $k$  sub-conjuntos mutuamente disjuntos tal que
    - $C_1 \cup C_2 \cup \dots \cup C_k = X$
    - $C_i \neq \emptyset$
    - $C_i \cap C_j = \emptyset$  para  $i \neq j$
  - Em outras palavras: *particionamento sem sobreposição*

# Partição sem sobreposição

- Definição do problema
  - Cada objeto pertence a um cluster dentre ***k*** clusters possíveis
    - O valor de ***k*** é normalmente definido pelo usuário
  - Qualidade da partição
    - Normalmente envolvem a otimização de algum índice
    - Critério numérico
- Um dos algoritmos mais utilizados é o *k-means*
  - Também chamado de *k-médias*

# K-means

- Funcionamento

- 1) Escolher aleatoriamente um número  $k$  de centroides (centros ou *seeds*) para iniciar os clusters
- 2) Cada objeto é atribuído ao cluster cujo centroides é o mais próximo
  - Usar alguma medida de distância (e.g. Euclidiana)
- 3) Mover cada centroide para a média dos objetos do cluster correspondente

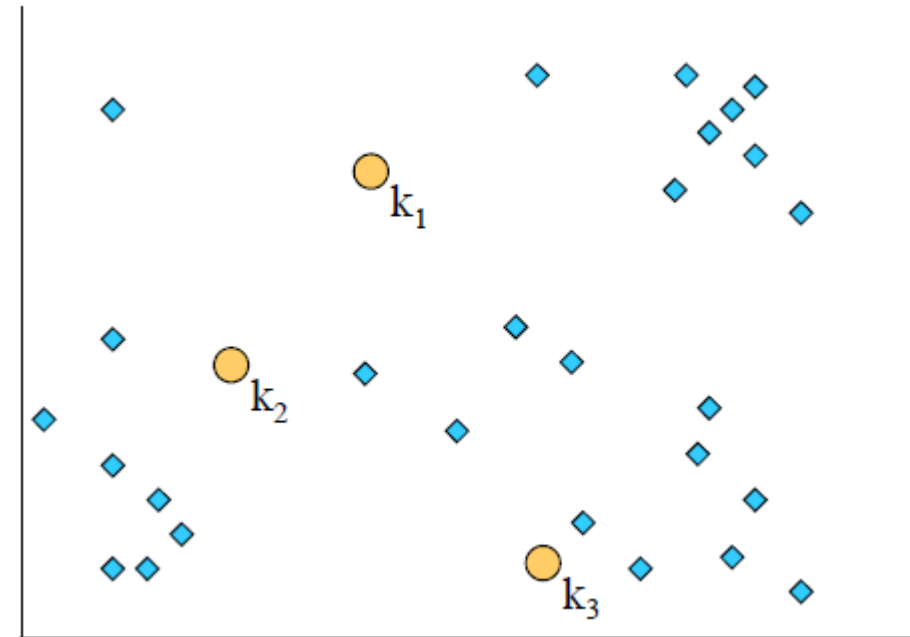
# K-means

- Funcionamento
  - 4) Repetir os passos 2 e 3 até que algum critério de convergência seja obtido
    - Número máximo de iterações
    - Limiar mínimo de mudanças nos centroides



# K-means

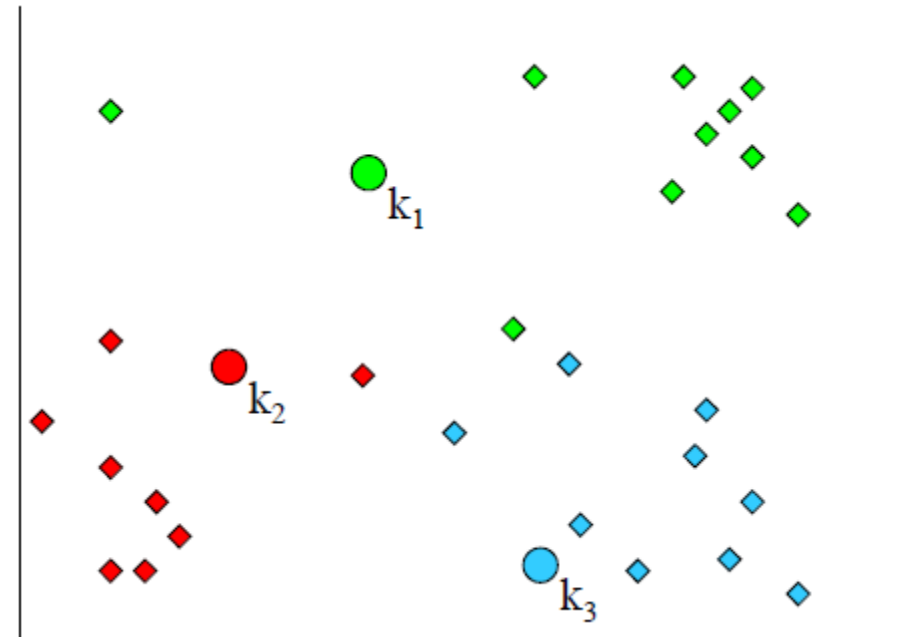
- Passo 1
  - Escolher  $k$  centros iniciais ( $k = 3$ )



Slide baseado no curso de Gregory Piatetsky-Shapiro, disponível em <http://www.kdnuggets.com>

# K-means

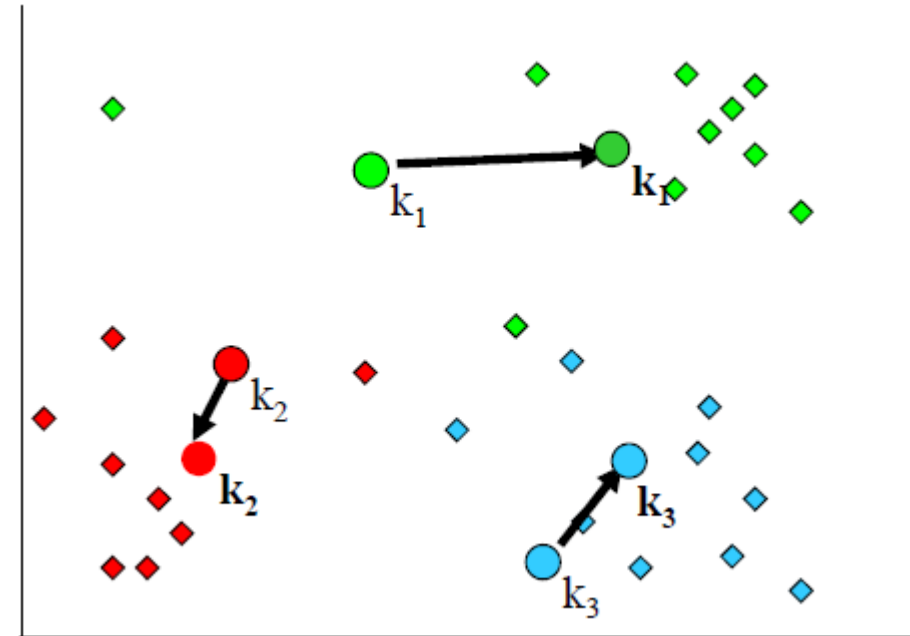
- Passo 2
  - Atribuir cada objeto ao cluster de centro mais próximo



Slide baseado no curso de Gregory Piatetsky-Shapiro, disponível em <http://www.kdnuggets.com>

# K-means

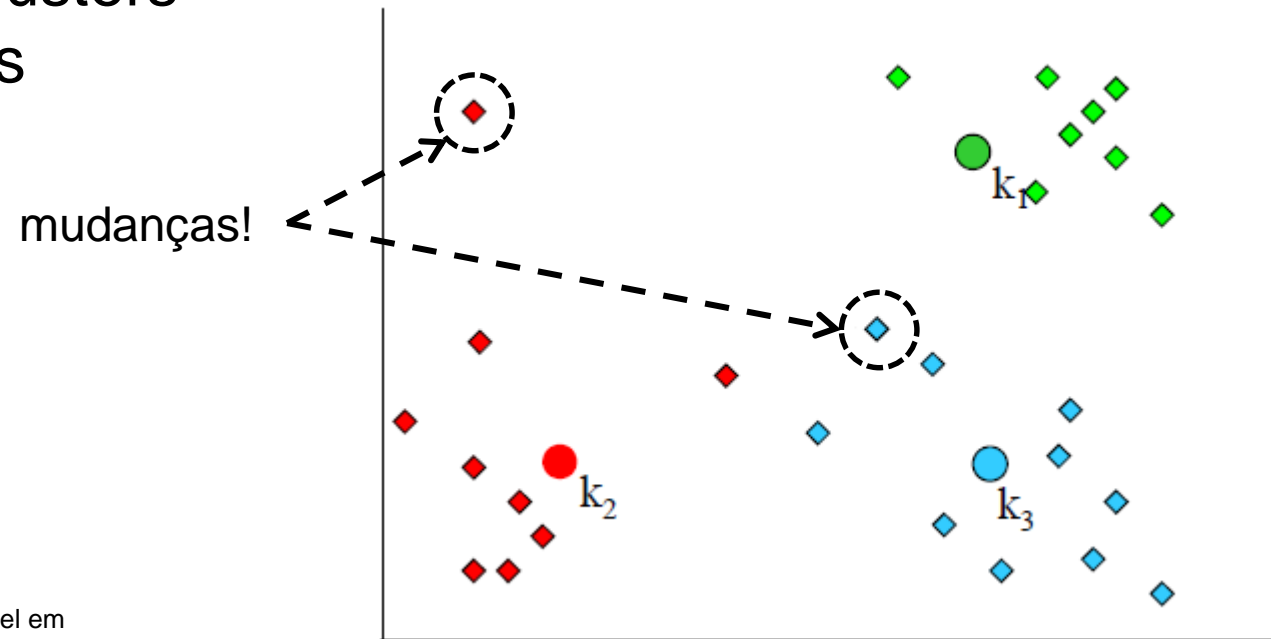
- Passo 3
  - Mover cada centro para o vetor médio do cluster (centroide)



Slide baseado no curso de Gregory Piatetsky-Shapiro, disponível em <http://www.kdnuggets.com>

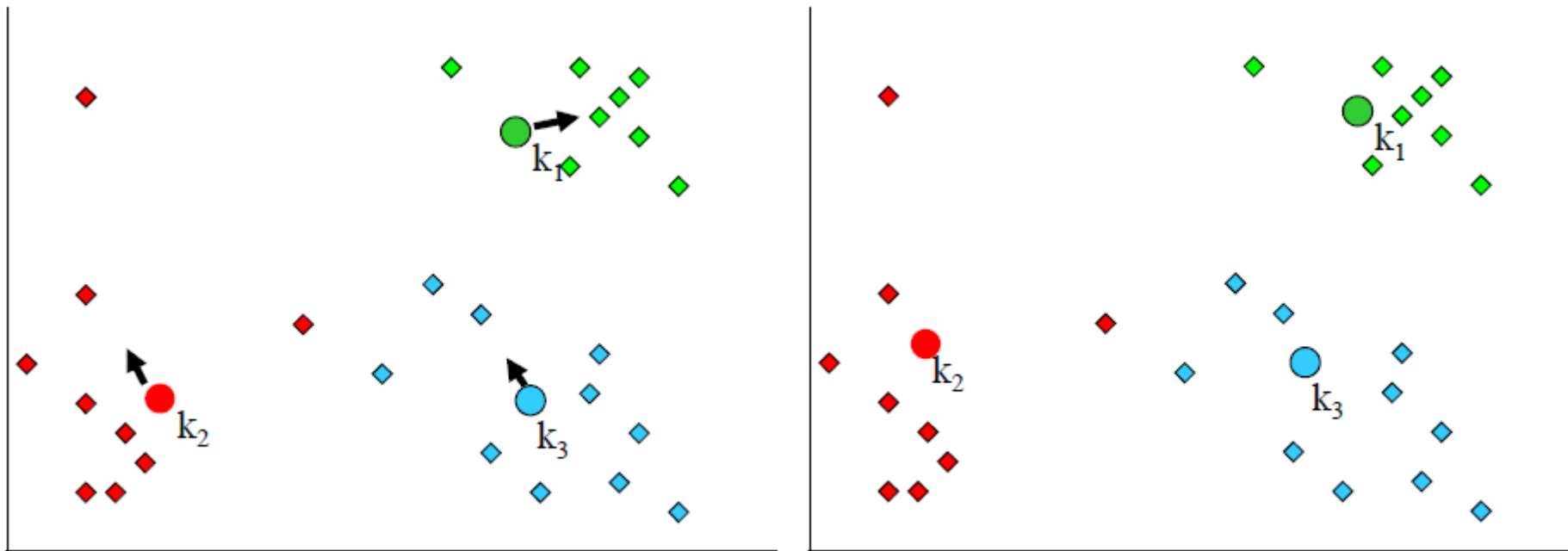
# K-means

- Passo 2
  - Re-atribuir os objetos aos clusters de centroides mais próximos



# K-means

- Passo 2
  - Re-calcular vetores médios

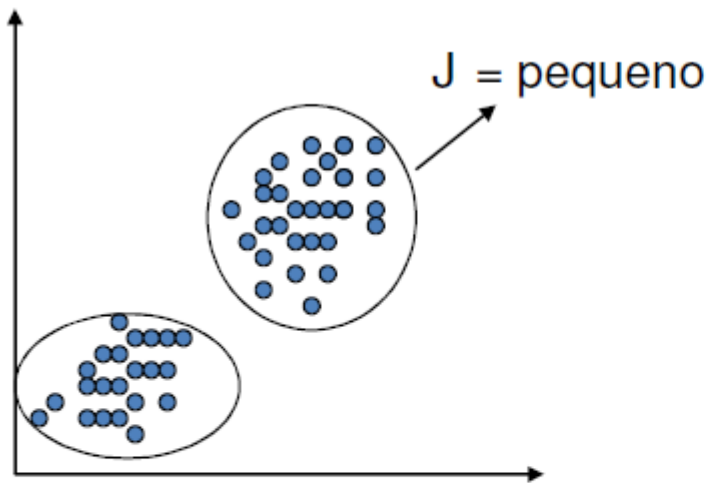


# K-means

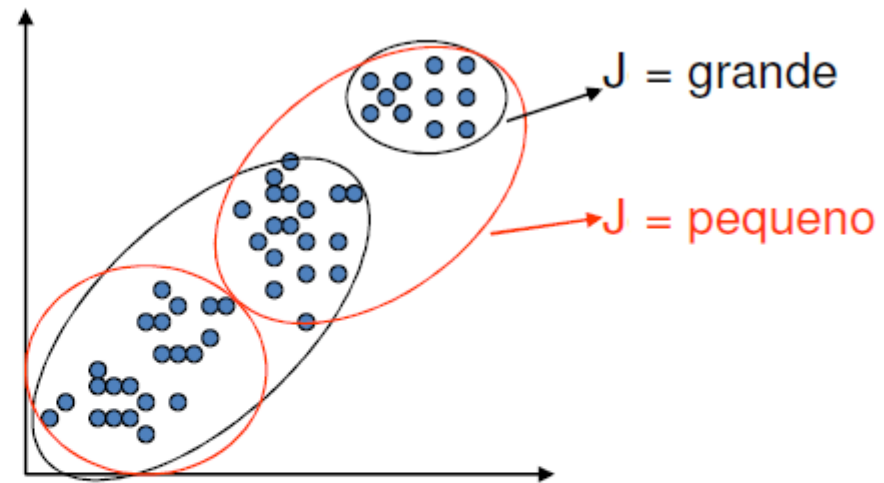
- Basicamente, o método de k-means busca minimizar a seguinte função objetivo (soma dos erros quadrados)
  - $J = \sum_{i=1}^k \sum_{x_j \in C_i} d(x_j, \bar{x}_i)^2$
- onde  $\bar{x}_i$  é o centroide do i-ésimo cluster
  - $\bar{x}_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$

# K-means

- Soma dos erros quadrados



Adequado nesses casos  
- Separação natural

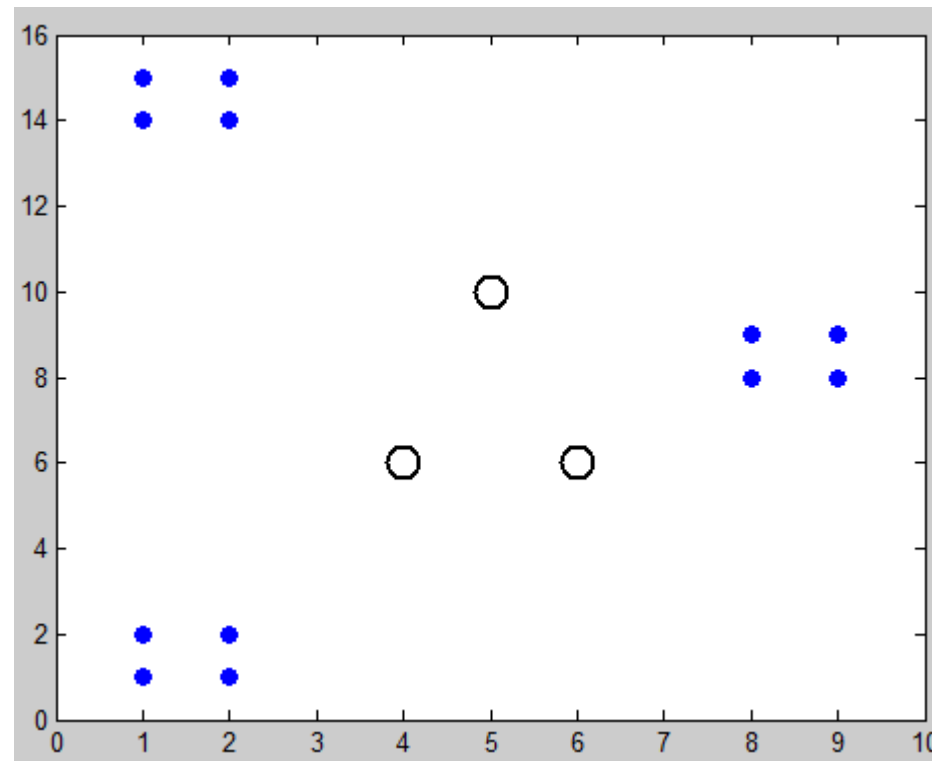


Não é muito adequado para dados  
mais dispersos.  
*Outliers* podem afetar bastante os  
vetores médios

# Kmeans: exemplo

- Executar o k-means com  **$k=3$**  nos dados abaixo a partir dos centros propostos

dados		centros	
x	y	x	y
1	2	6	6
2	1	4	6
1	1	5	10
2	2		
8	9		
9	8		
9	9		
8	8		
1	15		
2	15		
1	14		
2	14		

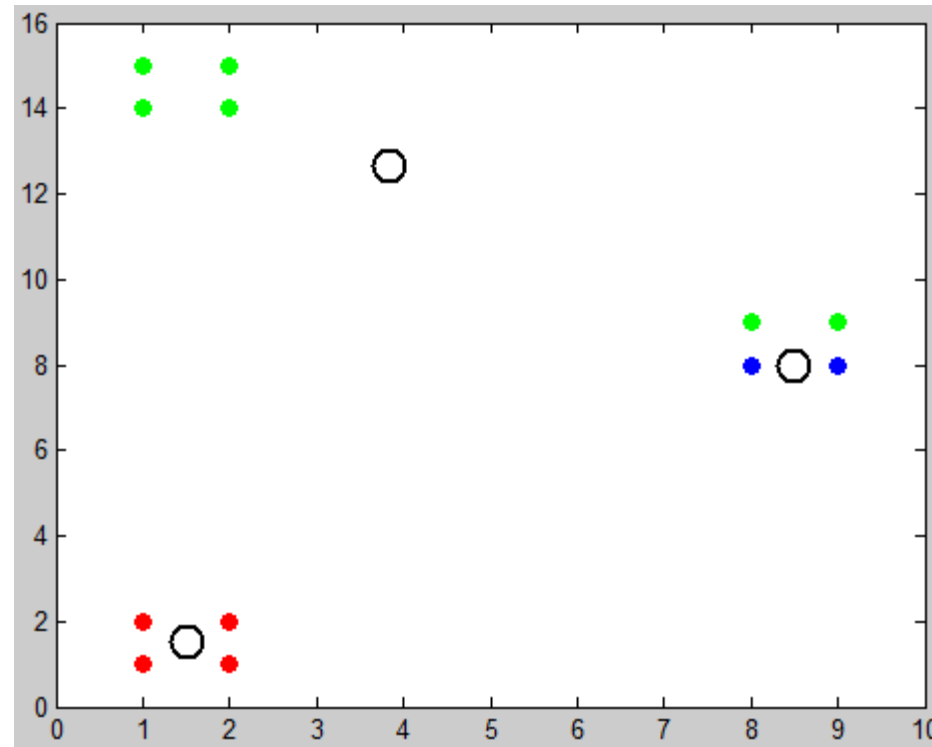




# Kmeans: exemplo

- Primeira Iteração

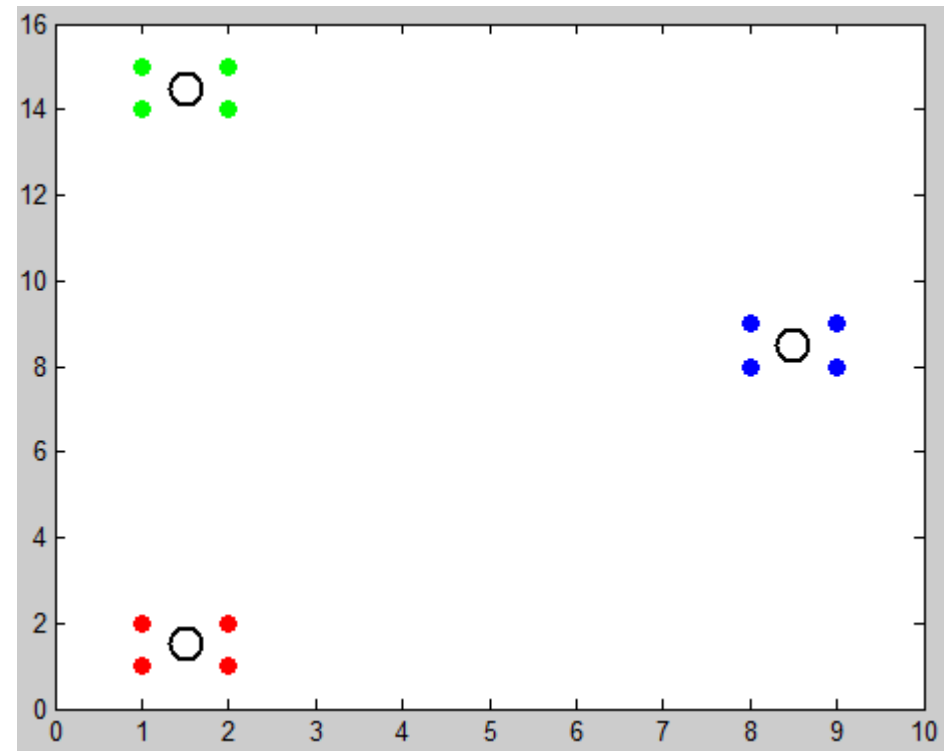
dados			centros	
x	y	C	x	y
1	2	2	8,5	8
2	1	2	1,5	1,5
1	1	2	3,8	12,6
2	2	2		
8	9	3		
9	8	1		
9	9	3		
8	8	1		
1	15	3		
2	15	3		
1	14	3		
2	14	3		



# Kmeans: exemplo

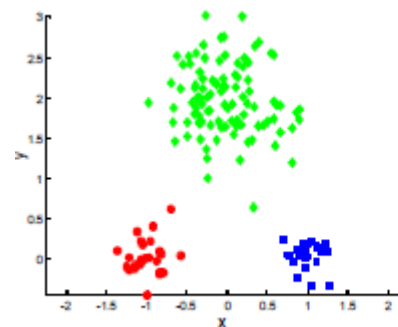
- Segunda Iteração

dados			centros	
x	y	C	x	y
1	2	2	8,5	8,5
2	1	2	1,5	1,5
1	1	2	1,5	14,5
2	2	2		
8	9	1		
9	8	1		
9	9	1		
8	8	1		
1	15	3		
2	15	3		
1	14	3		
2	14	3		

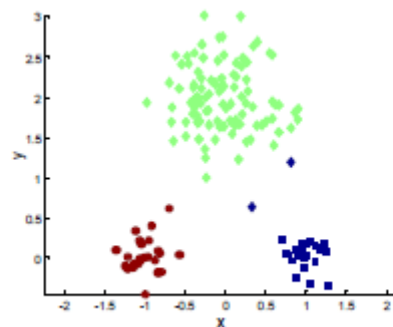


# K-means | Problemas

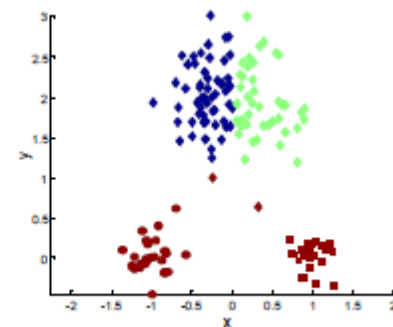
- Variação no resultado dependendo da escolha dos centroides (*seeds*) iniciais
  - Quando se têm noção dos centroides, pode-se melhorar a convergência do algoritmo.
  - Execução do algoritmo várias vezes, permite reduzir impacto da inicialização aleatória



Dados Originais

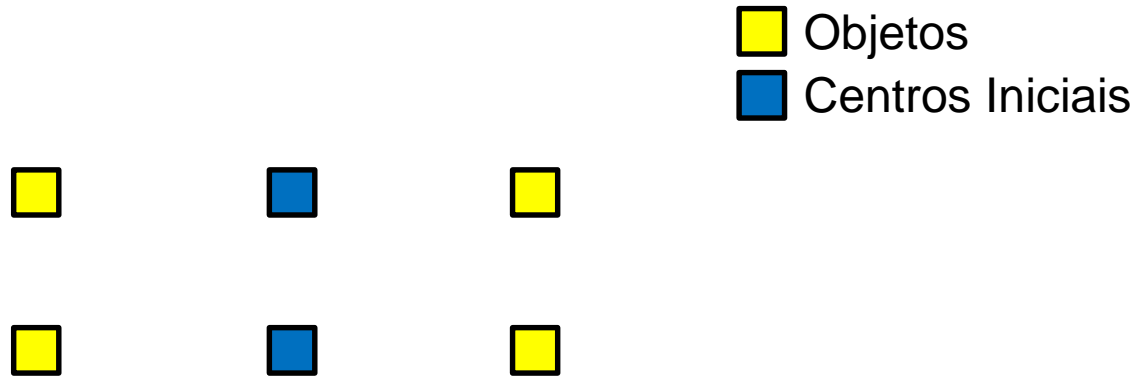


Possíveis Clusters



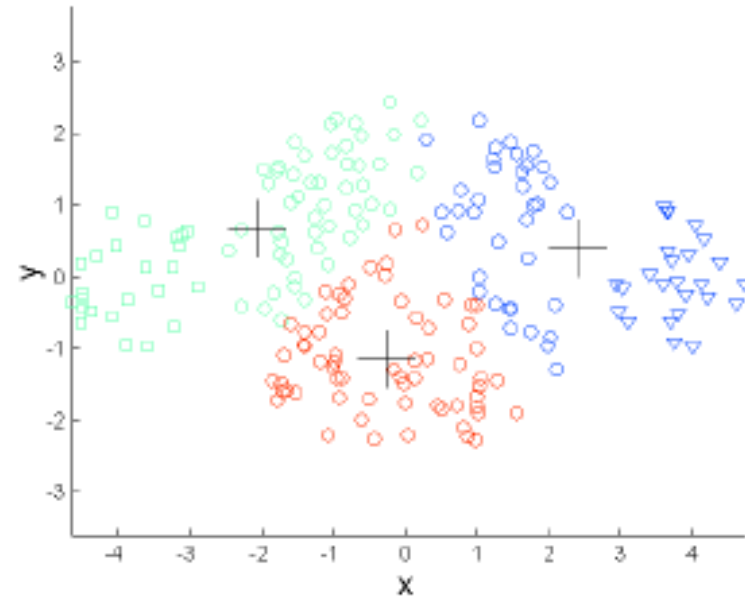
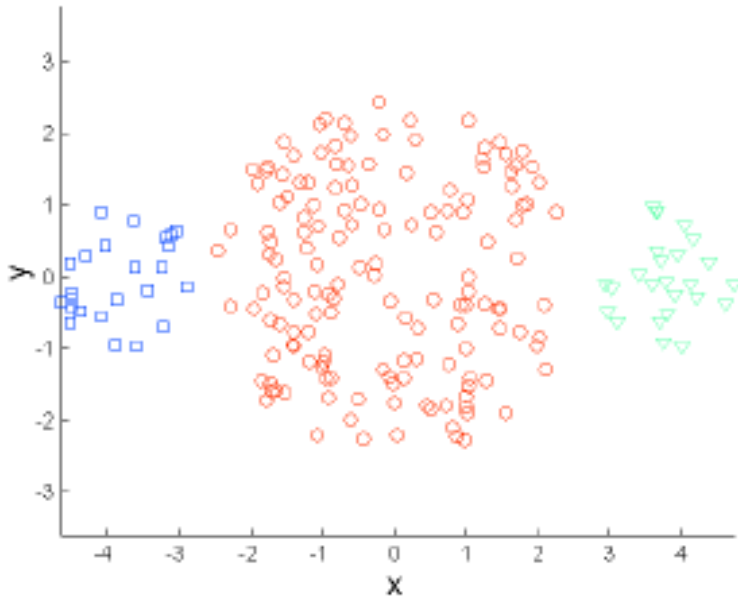
# K-means | Problemas

- O método pode “ficar preso” em ótimos locais
  - Os dois centros são equivalentes para o conjunto de objetos



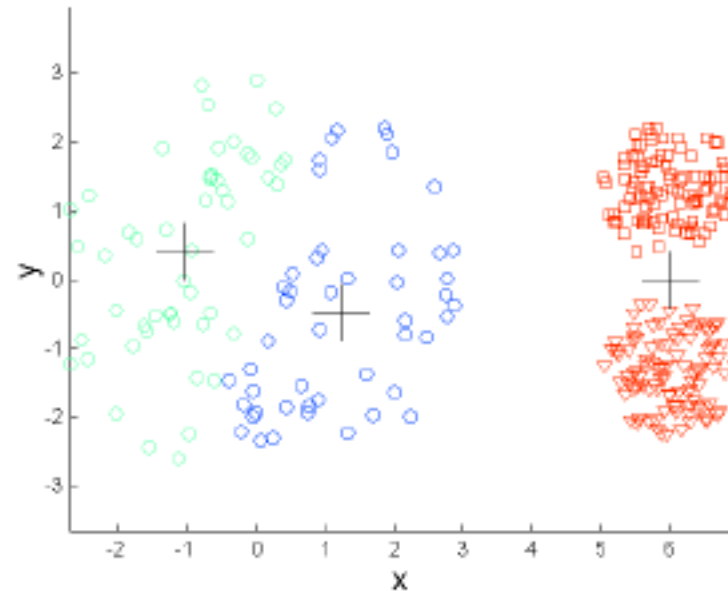
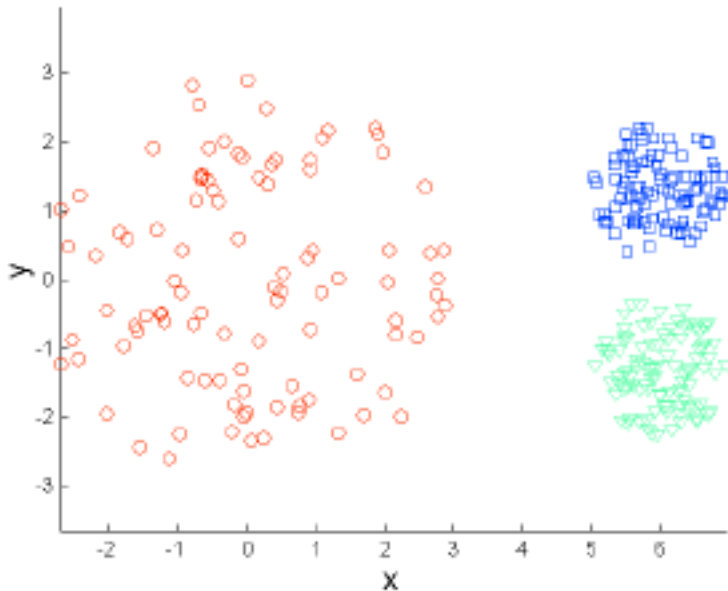
# K-means | Problemas

- É bastante susceptível a problemas quando clusters são de diferentes **tamanhos**



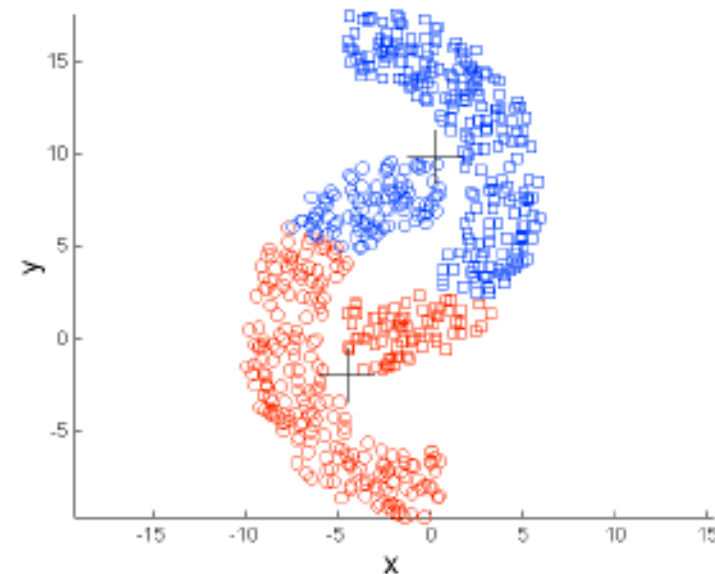
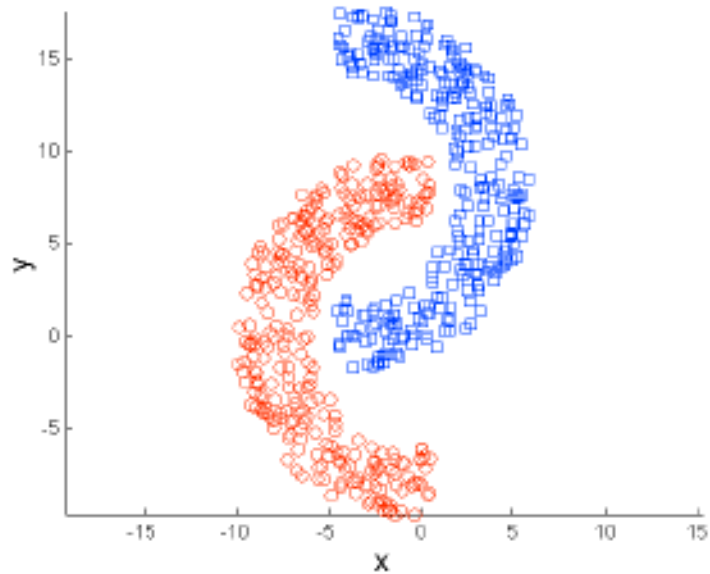
# K-means | Problemas

- É bastante susceptível a problemas quando clusters são de diferentes **densidades**



# K-means | Problemas

- É bastante susceptível a problemas quando clusters são de diferentes **formatos (em geral não globulares)**



# K-means | Problemas

- Dificuldade em definir o valor de  $k$
- Limitado a atributos numéricos
- Cada item deve pertencer a um único cluster
  - Partição rígida (sem sobreposição)



# K-means

- Apesar de seus problemas, podemos melhorar seu desempenho de diferentes formas
  - Atualização incremental
  - K-medianas
  - K-medóides
  - K-d tree
  - Etc.

# K-means | Variações

- Atualização incremental dos centroides
  - Cálculo dos novos centroides não demanda recalcular tudo novamente
    - Oportunidade de aumento no desempenho
  - Cálculo do centroide só depende
    - De seu número de objetos
    - Dos novos objetos atribuídos ao cluster
    - Dos objetos que deixaram o cluster
    - Do valor anterior do centroide

# K-means | Variações

- K-medianas: substitui as médias pelas medianas
  - Exemplo
    - Média de 1, 3, 5, 7, 9 é **5**
    - Média de 1, 3, 5, 7, 1009 é **205**
    - Mediana de 1, 3, 5, 7, 1009 é **5**
  - Vantagem: menos sensível a outliers
  - Desvantagem
    - Maior complexidade computacional devido a etapa de ordenação

# K-means | Variações

- K-medóides: substitui cada centroide por um objeto representativo do cluster
  - Medóide
    - Objeto mais próximo (em média ) aos demais objetos do cluster
  - Vantagens:
    - É menos sensível a outliers
    - Pode ser aplicado a bases com atributos categóricos (cálculo relacional)
  - Desvantagem
    - Complexidade quadrática

# K-means | Variações

- K-means para Data Streams (fluxo de dados)
  - Utiliza o conceito de vizinhos mais próximos (K-NN)
  - Objetos são dinamicamente incorporados ao cluster mais próximo
  - Atualização do centroide do cluster pode ser incremental
  - Heurísticas podem ser usadas para criação ou remoção de clusters

# K-means | Variações

- Múltiplas Execuções
  - Várias execuções do k-means
  - Uso de diferentes valores de  $k$  e de posições iniciais dos centroides
    - Ordenado: uma execução para cada valor de  $k$  em  $[k_{min}, k_{max}]$
    - Aleatório: para cada execução  $k$  é sorteado em  $[k_{min}, k_{max}]$

# K-means | Variações

- Múltiplas Execuções
  - Usa um critério de qualidade (critério de validade de agrupamento)
    - Permite escolher a melhor partição
- Vantagens
  - Permite estimar o melhor valor de  $k$
  - Menos sensível a mínimos locais
- Desvantagem
  - Pode apresentar um custo computacional elevado

# Qualidade do cluster

- Como avaliar relativamente a qualidade de diferentes partições
  - Necessidade de um tipo de índice
- Critério Relativo de Validade de Agrupamento
  - Existem dezenas de critérios na literatura
    - Alguns são melhores para algumas classes de problemas
    - Não há garantias de que um certo critério funcione para todos os problemas em geral

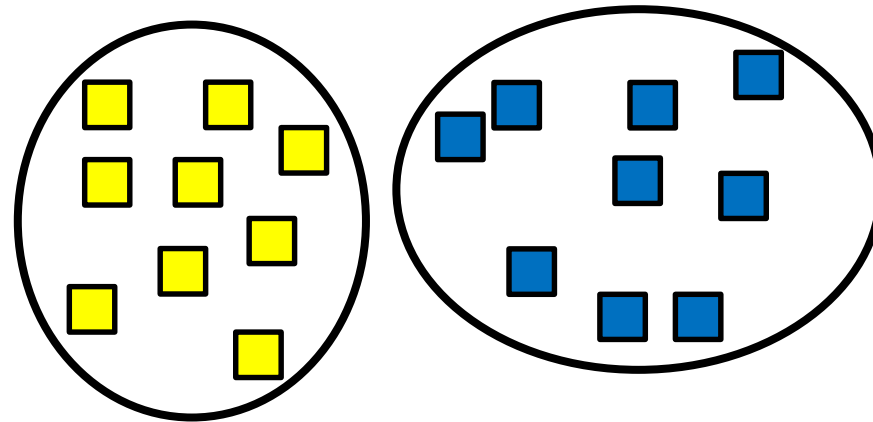


# Qualidade do cluster

- Alguns critérios existentes na literatura
  - Largura de Silhueta
  - Variance Ratio Criterion (VRC)
    - Também denominado Calinski-Harabaz
  - Davies-Bouldin
  - Índice de Dunn
    - E variantes

# Qualidade do cluster

- Largura de Silhueta
  - Cada cluster é representado por uma silhueta
    - Isso nos mostra que objetos se posicionam bem dentro do cluster e quais meramente ficam em uma posição intermediária



# Qualidade do cluster

- Largura de Silhueta

- Para cada objeto  $i$  obtêm-se o valor  $s(i)$

- $s(i) = \frac{b_i - a_i}{\max(a_i, b_i)}$

- Onde

- $a_i$  é a dissimilaridade média do objeto  $i$  em relação a todos os outros objetos do seu cluster
    - $b_i$  é a dissimilaridade média do objeto  $i$  em relação a todos os outros objetos do cluster vizinho mais próximo

# Qualidade do cluster

- Largura Média de Silhueta (SWC)
  - É a média de  $s(i)$  para todos os objetos  $i$  nos dados
    - $SWC = \frac{1}{N} \sum_{i=1}^N s(i)$
    - Coeficiente de Silhueta varia de -1 a 1
  - Valores negativos não são desejáveis
    - Significa que a distância média dos objetos para seu cluster é maior que distância média para outros clusters

# Qualidade do cluster

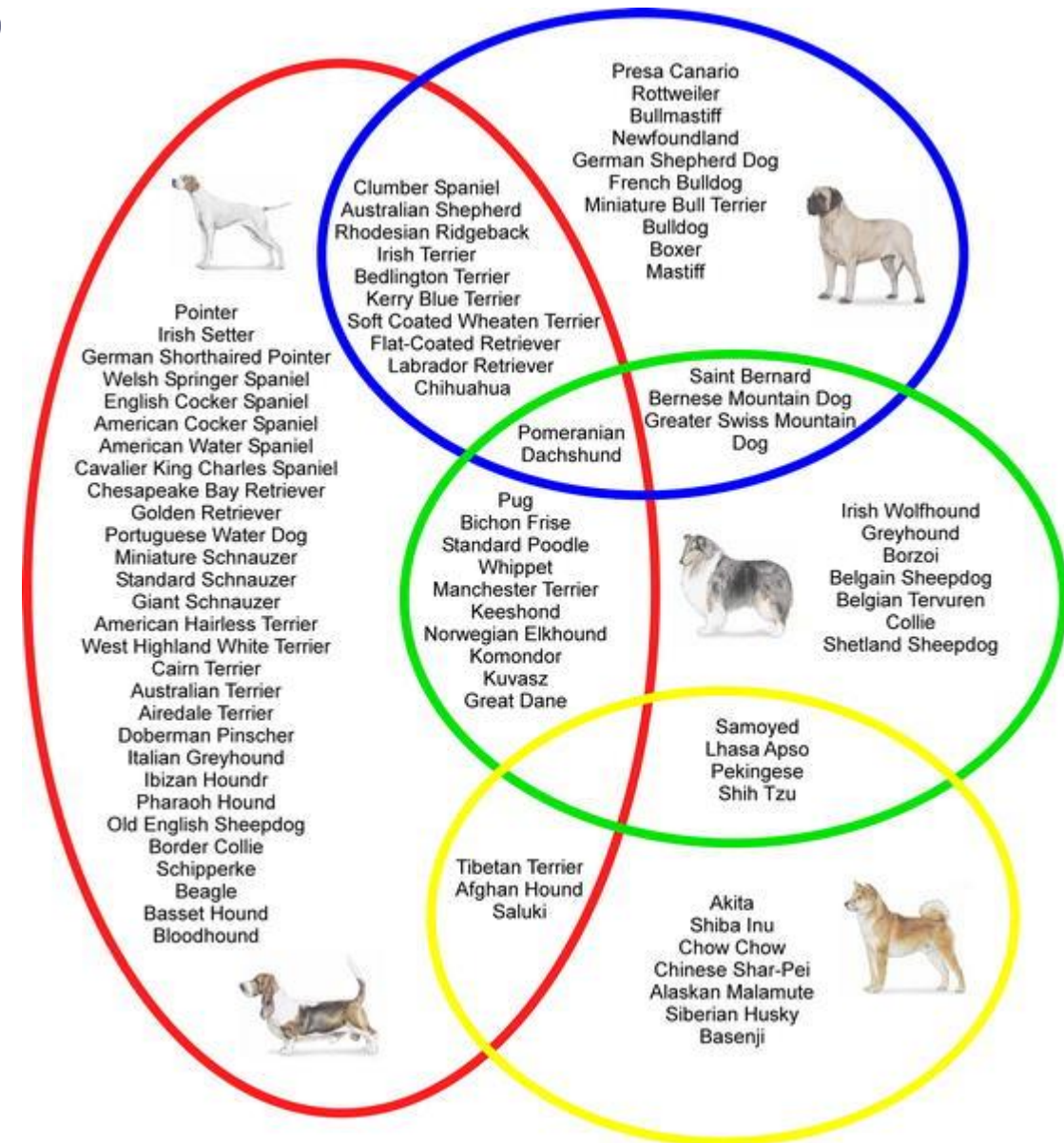
- Largura Média de Silhueta (SWC)
  - Valores Ideais
    - valores positivos
    - $a_i$  bem próximo de zero
    - Coeficiente de silhueta bem próximo de 1
  - Pode ser utilizado para selecionar o "melhor" número de clusters
    - Selecionar o valor de  $k$  dando a maior média de  $s(i)$

# Partição com Sobreposição

- Método k-means
  - Partição sem sobreposição dos dados
  - Também conhecido como *partição rígida*
- Muitos problemas envolvem grupos mal delineados
  - Não podem ser separados adequadamente dessa maneira
  - Os dados podem compreender categorias que se sobrepõem umas às outras em diferentes níveis

# Partição com Sobreposição

- Exemplo
  - A estrutura da população de 85 raças de cães



# Partição com Sobreposição

- Métodos de agrupamento com sobreposição são concebidos para lidar essas situações
  - Em inglês, *overlapping clustering algorithms*
- Ao todo, 3 tipos de partições são possíveis
  - Soft
    - Objetos podem pertencer de forma integral a mais de um grupo
  - Fuzzy
    - Objetos pertencem a todos os grupos com diferentes graus de pertinência
  - Probabilísticas
    - Objetos possuem probabilidades de pertinência associadas a cada cluster



# Partição com Sobreposição

- Agrupamento Fuzzy: Fuzzy c-Means (FCM)
  - Trata-se de uma extensão de k-means para o domínio fuzzy
    - Garantia de convergência apenas para soluções locais
  - Também é susceptível a mínimos locais da função objetivo  $J$ 
    - Depende da inicialização dos protótipos
    - Pode-se utilizar o esquema de múltiplas execuções
  - Existem dezenas de variantes

# Partição com Sobreposição

- Agrupamento Fuzzy: Fuzzy c-Means (FCM)

- $\min_{f_{ij}} J = \sum_{i=1}^k \sum_{j=1}^N f_{ij}^m d(x_j, \bar{x}_i)^2$
- $0 \leq f_{ij} \leq 1$
- $\sum_{i=1}^k f_{ij} = 1, \forall j \in \{1, 2, \dots, N\}$
- $0 < \sum_{j=1}^N f_{ij} < N, \forall i \in \{1, 2, \dots, k\}$

- Onde

- $f_{ij}$ : Pertinência do objeto  $j$  ao grupo  $i$
- $m > 1$  (*usualmente  $m = 2$* )

# Partição com Sobreposição

- Agrupamento Fuzzy: Fuzzy c-Means (FCM)
  - Existem versões fuzzy para os critérios de validade de agrupamento discutidos anteriormente
    - Silhueta Fuzzy
    - Jaccard Fuzzy
    - Etc.

# Agradecimentos

- Agradeço ao professor
  - Prof. Ricardo J. G. B. Campello – ICMC/USP
- pelo material disponibilizado