Aprendizado de Máquina

Agrupamento de Dados

Eduardo R. Hruschka

Agenda

- Motivação e conceitos
- Definições preliminares
- k-means
- Estimando o número de clusters a partir dos dados
- Bisecting k-means
- k-medoids
- EM para misturas de Gaussianas
- Avaliação de agrupamentos

Motivação e potenciais aplicações

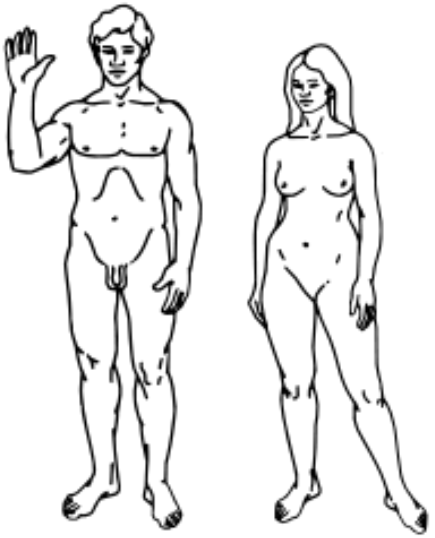
Humanos se interessam por categorizações:

- Música: erudita, popular, religiosa etc.

- Filmes: Animação, Comédia, Drama etc.

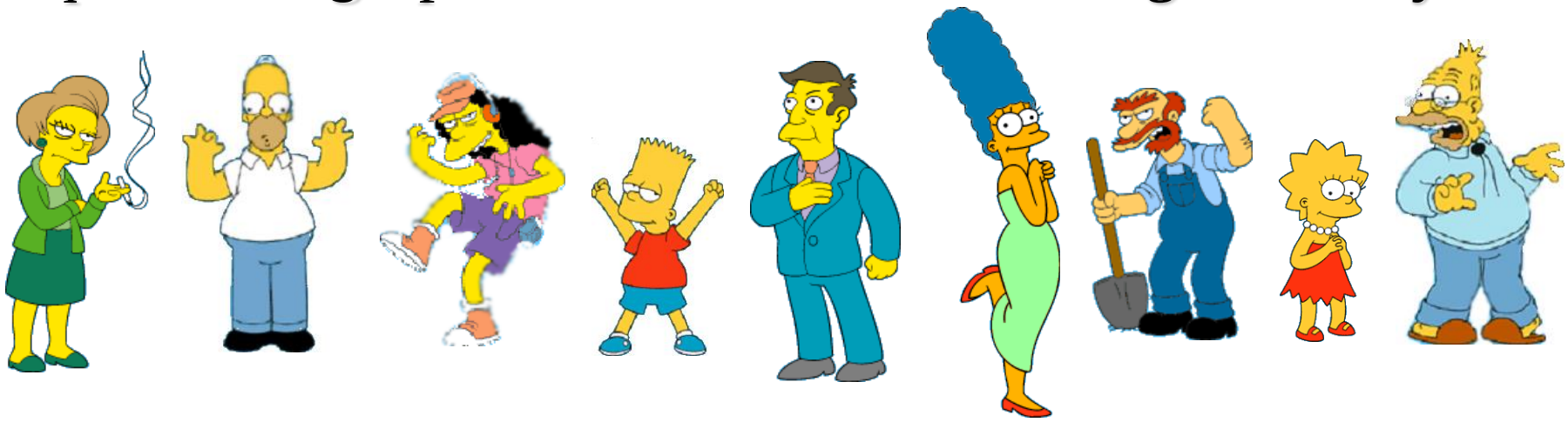
Diversas ciências se baseiam na *organização* de objetos de acordo com suas similaridades.

- Biologia:
  - Reino: Animalia
  - Ramo: Chordata
  - Classe: Mammalia
  - Ordem: Primatas
  - Família: *Hominidae*
  - Gênero: *Homo* (homem moderno)
  - Espécie: *Homo sapiens*

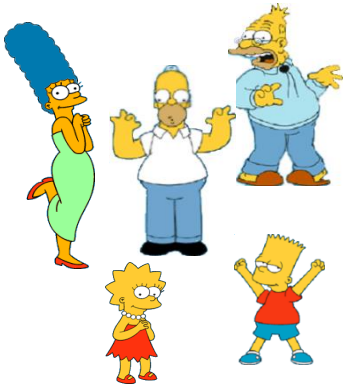


- Entretanto, existem muitas situações nas quais não sabemos de antemão uma maneira apropriada de **agrupar** uma coleção de objetos de acordo com suas "similaridades";
  - massas de dados, possivelmente descritas por várias características (atributos) diferentes.
- Frequentemente não sabemos sequer se existe algum **agrupamento natural** dos objetos segundo um conjunto de características que descrevem esses objetos;
- Vejamos um exemplo ilustrativo...

O que é um *agrupamento natural* entre os seguintes objetos?

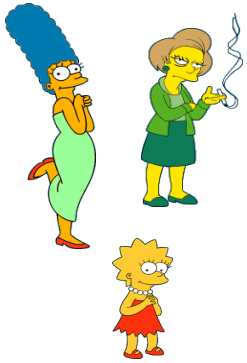


Grupo é um conceito subjetivo:



Família

Empregados da Escola



Mulheres

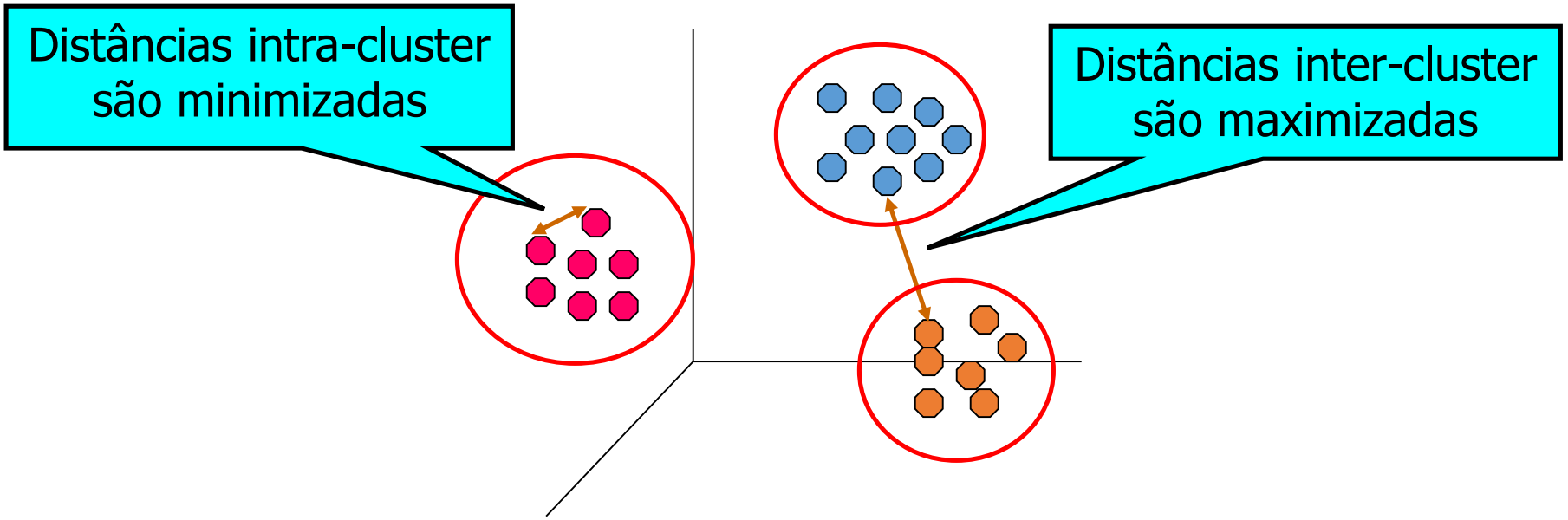
Homens

Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Manaus.

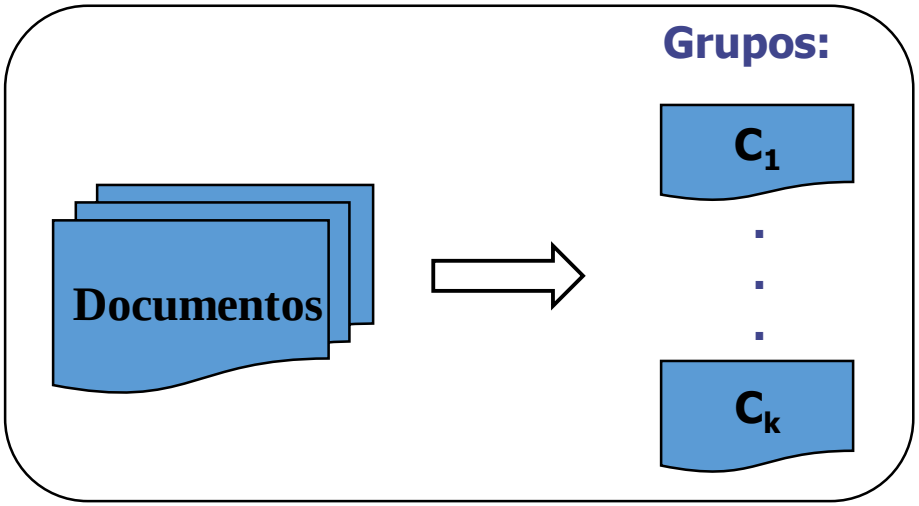
Uma definição para *agrupamento de dados*

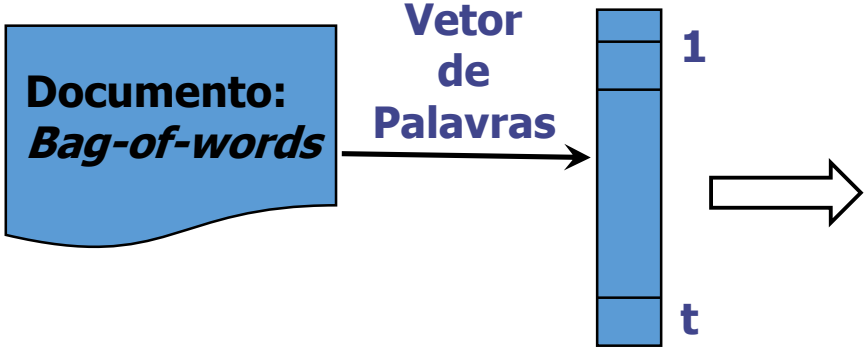
"Finding groups of objects such that the objects in a group are similar (or related) to one another and different from (or unrelated to) the objects in other groups." (Tan et al., 2006)

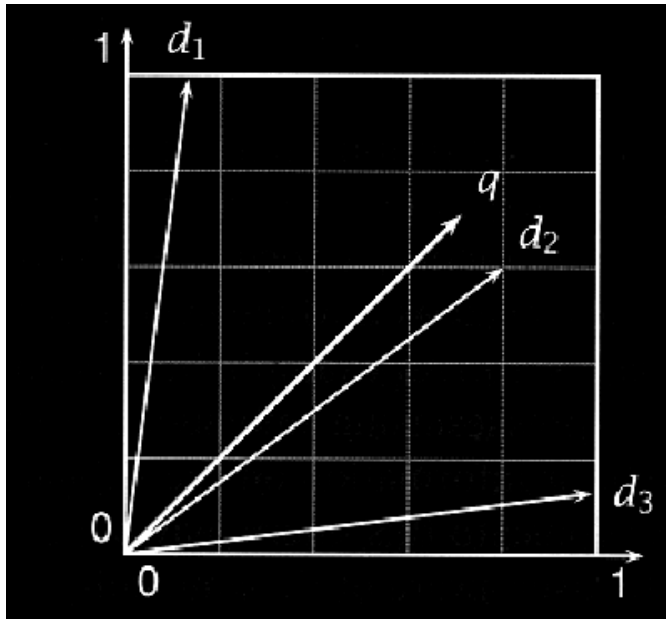
- Uma visão matemática/geométrica:



Agrupamento para mineração de textos







Frequência com que se usa *clustering*?

Web of Science: **+12.000 artigos** usando o termo *cluster analysis* no (título, palavras chaves, resumo) oriundos de mais de **3.000 *journals*** diferentes.

(Xu & Wunsch, *Clustering*, IEEE Press, 2009)

Algorithms for data analysis / data mining

Which methods/algorithms did you use for data analysis in 2011? [311 voters]	
Decision Trees/Rules (186)	59.8 %
Regression (180)	57.9 %
Clustering (163)	52.4 %
Statistics (descriptive) (149)	47.9 %
Visualization (119)	38.3 %
Time series/Sequence analysis (92)	29.6 %
Support Vector (SVM) (89)	28.6 %
Association rules (89)	28.6 %
Ensemble methods (88)	28.3 %
Text Mining (86)	27.7 %
Neural Nets (84)	27.0 %
Boosting (73)	23.5 %
Bayesian (68)	21.9 %
Bagging (63)	20.3 %
Factor Analysis (58)	18.7 %
Anomaly/Deviation detection (51)	16.4 %
Social Network Analysis (44)	14.2 %
Survival Analysis (29)	9.32 %
Genetic algorithms (29)	9.32 %
Uplift modeling (15)	4.82 %

www.kdnuggets.com/polls/2011/algorithms-analytics-data-mining.html

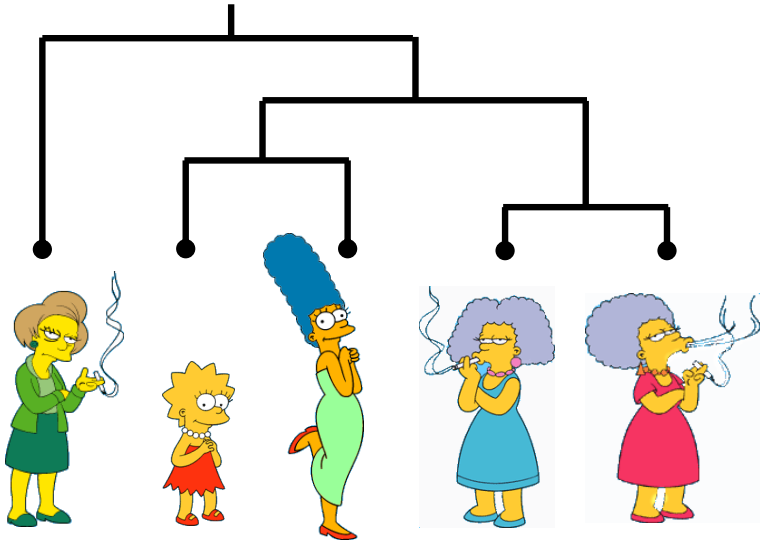
Lembre que algoritmos induzem os *clusters*

- Os *clusters* a serem induzidos dependem de uma série de fatores, além dos dados propriamente ditos:
  - medidas de dis(similaridade), índices de avaliação, parâmetros definidos pelo usuário etc.
  - fortemente dependente do domínio / problema
- Na perspectiva de **Aprendizado de Máquina** (AM) há uma relação com o conceito de bias indutivo:
  - *projetista define o que o computador pode aprender*
  - *existem **centenas** de algoritmos...*

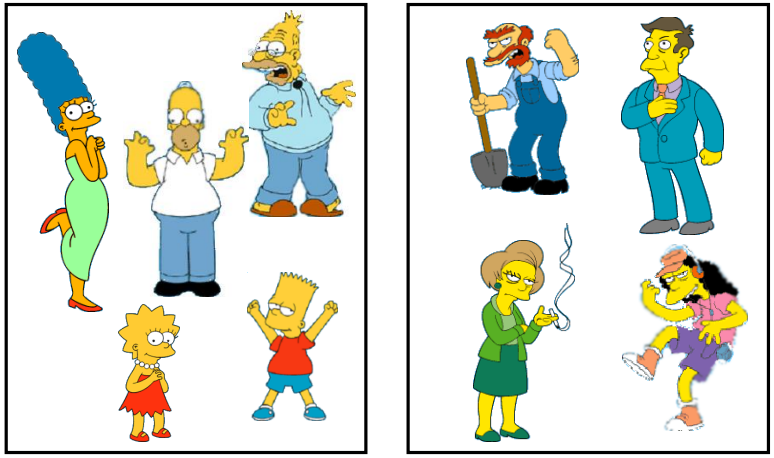
Métodos particionais e hierárquicos

- **Particionais:** constroem uma partição dos dados
- **Hierárquicos:** constroem uma hierarquia de partições

Hierárquicos

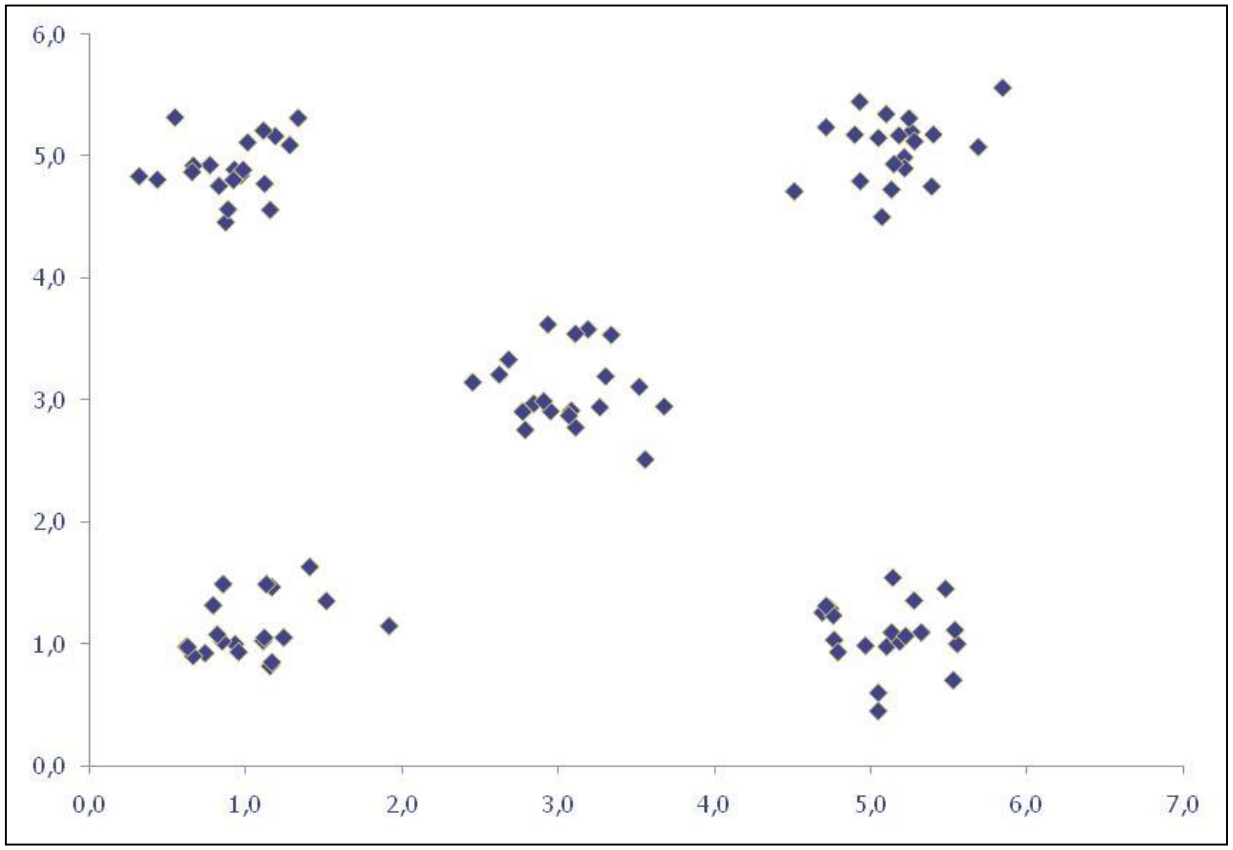


Particionais



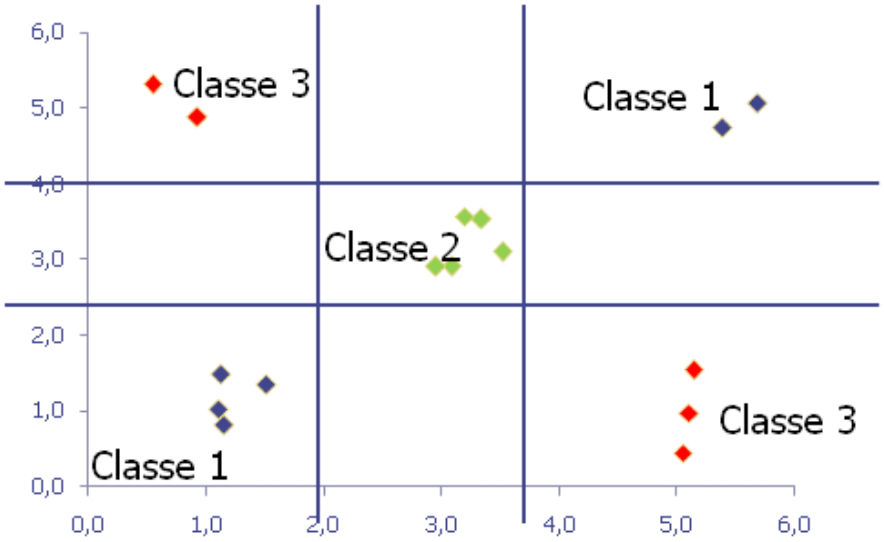
Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Manaus.

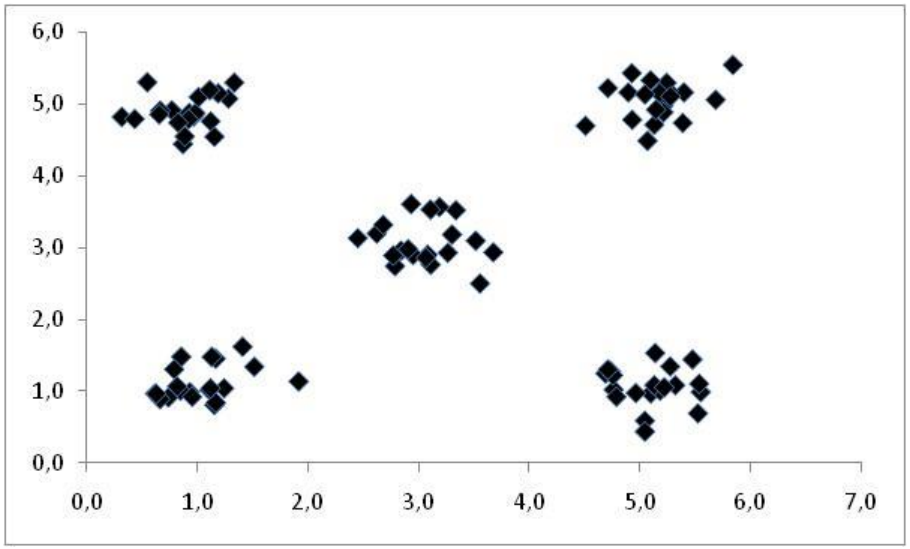
Agrupamento x Classificação



Agrupamento:
Indução de grupos a partir da base de dados...

- Grupos obtidos serão então cuidadosamente estudados

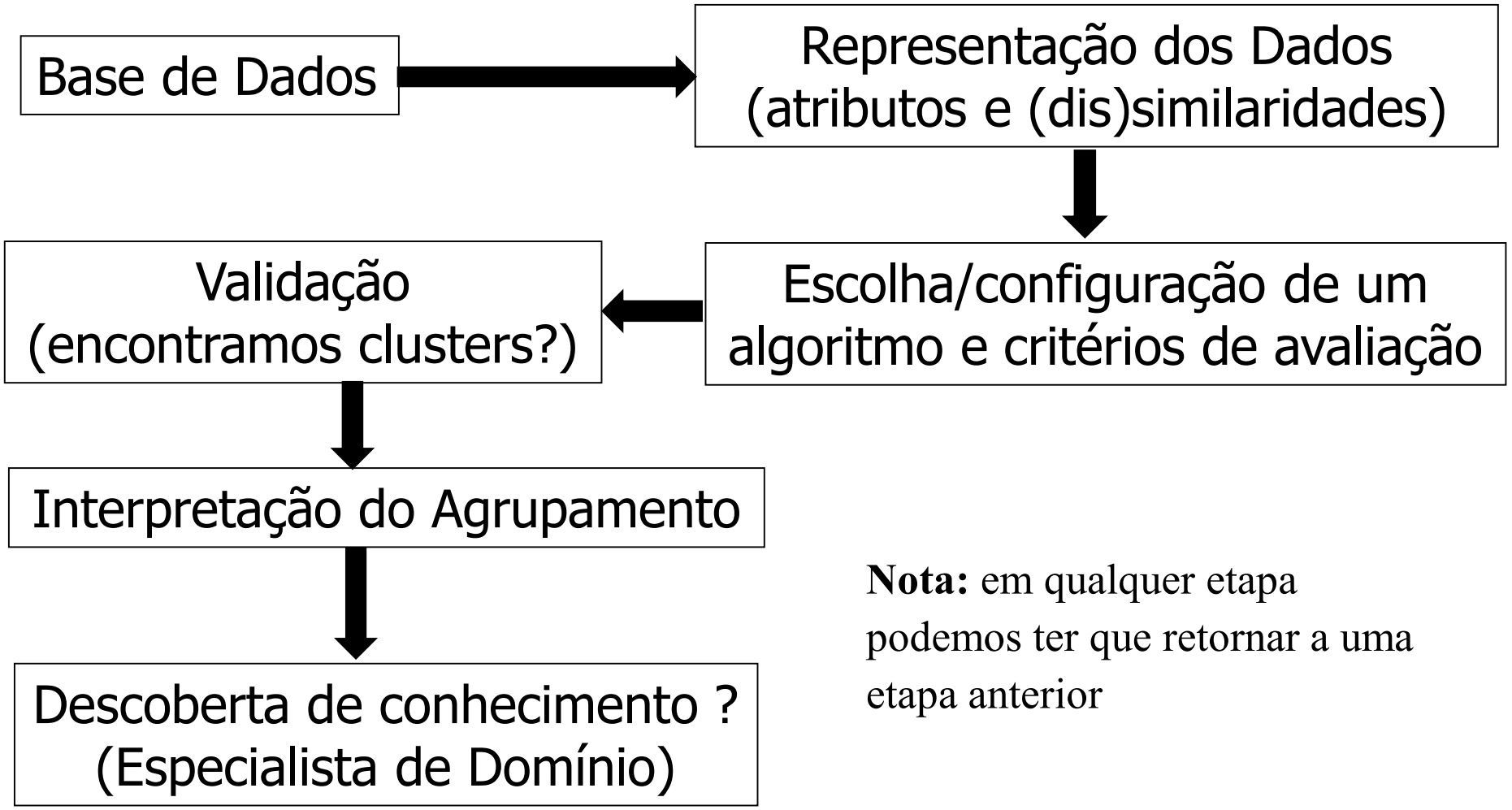




Base de treinamento com dados rotulados: classificador (modelo)

Rotular dados de teste em função do modelo obtido

Ciclo de modelagem em agrupamento



Nota: em qualquer etapa podemos ter que retornar a uma etapa anterior

Agenda

- Motivação e conceitos
- Definições preliminares
- k-means
- Estimando o número de clusters a partir dos dados
- Bisecting k-means
- k-medoids
- EM para misturas de Gaussianas
- Avaliação de agrupamentos

Definição. Considerando um conjunto de N objetos a serem agrupados $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, uma **partição** (rígida) é uma coleção de k grupos não sobrepostos $\mathbf{P} = \{\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_k\}$ tal que:

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

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

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

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

Definição. Uma **Matriz de Partição** é uma matriz com k linhas (no. de grupos) e N colunas (no. de objetos) na qual cada elemento μ_{ij} indica o *grau de pertinência* do j-ésimo objeto ($\mathbf{x}_j$) ao i-ésimo grupo ($\mathbf{C}_i$):

$$\mathbf{U}(\mathbf{X}) = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2N} \\ \vdots & & \ddots & \vdots \\ \mu_{k1} & \mu_{k2} & \cdots & \mu_{kN} \end{bmatrix}$$

Se essa matriz for **binária**, ou seja, $\mu_{ij} \in \{0,1\}$ e, ainda, se a restrição $\sum_i(\mu_{ij}) = 1 \ \forall j$ for respeitada, então denomina-se de matriz de partição rígida ou sem sobreposição.

Exemplo de matriz de partição: considerando uma partição **P** = {($\mathbf{x}_1$), ($\mathbf{x}_3$,$\mathbf{x}_4$,$\mathbf{x}_6$), ($\mathbf{x}_2$,$\mathbf{x}_5$)} temos:

$$\mathbf{U}(\mathbf{X}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

- Algoritmos *particionais* sem sobreposição buscam (explícita ou implicitamente) por uma matriz de partição rígida de um conjunto de objetos **X**.

- Há outras classes de algoritmos?

Particionamento combinatório

Problema: Presumindo que k seja conhecido, o no. de possíveis formas de agrupar N objetos em k *clusters* é dado por (Liu, 1968):

$$NM(N,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i \binom{k}{i}(k-i)^N$$

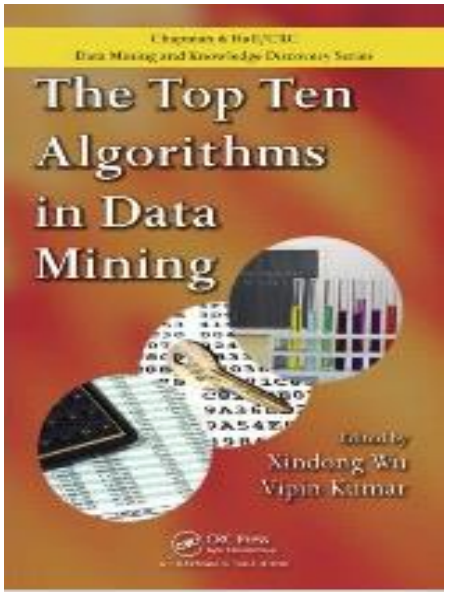
- Por exemplo, NM(100, 5) ≈ 56.6×10^{67}. Em um computador com capacidade de avaliar 10^9 partições/s, levaria ≈ 1.8×10^{50} séculos para processar todas as avaliações.
- Como k em geral é desconhecido, problema é ainda maior.
- Em problemas NP-Hard, precisamos de formulações alternativas.

Agenda

- Motivação e conceitos
- Definições preliminares
- k-means
- Estimando o número de clusters a partir dos dados
- Bisecting k-means
- k-medoids
- EM para misturas de Gaussianas
- Avaliação de agrupamentos

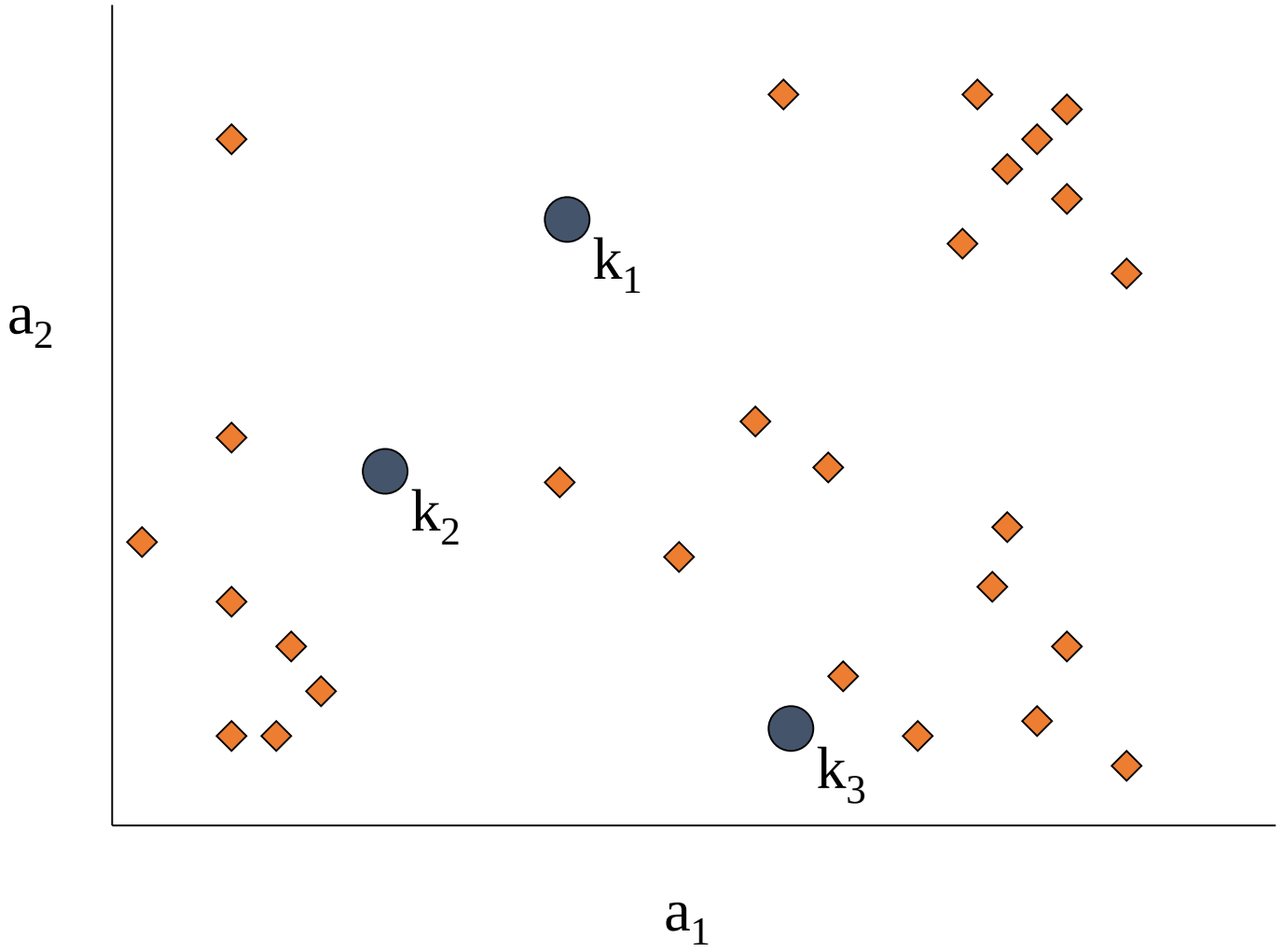
Algoritmo k-means

- Muito bem estudado (MacQueen, 1967; Kulis & Jordan, 2012)
- Conceitualmente simples e fácil de implementar
- Um dos algoritmos mais utilizados na prática:
  - Wu, X. and Kumar, V. (Editors), *The Top Ten Algorithms in Data Mining*, CRC Press, 2009.
  - X. Wu et al., "Top 10 Algorithms in Data Mining", *Knowledge and Information Systems*, vol. 14, pp. 1-37, 2008.



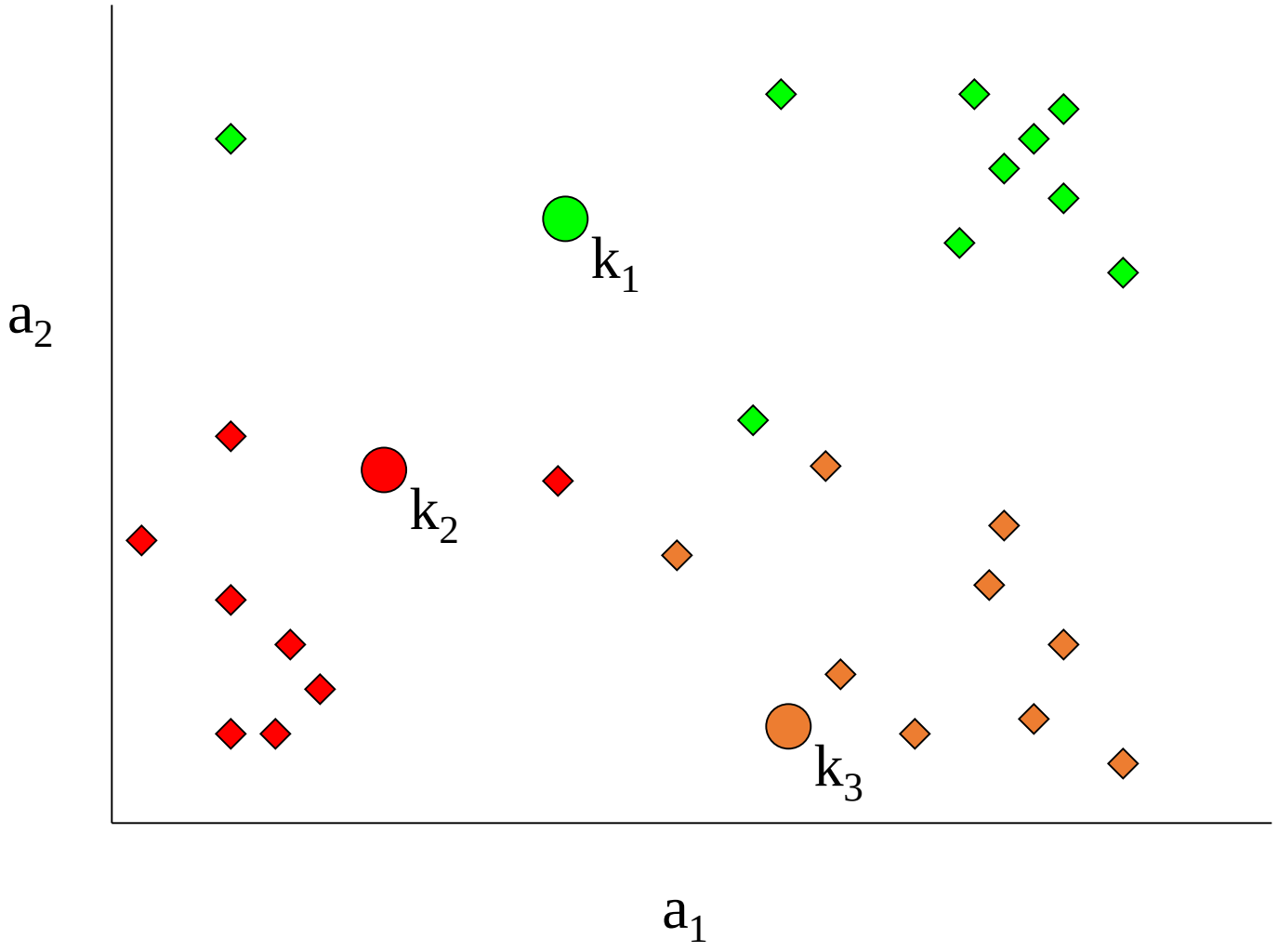
1) Escolher aleatoriamente *k* protótipos (centros) para os clusters (grupos)
2) Atribuir cada objeto para o cluster de centro mais *próximo* (segundo alguma medida de distância, e.g. Euclidiana)
3) Mover cada centro para a média (centróide) dos objetos do cluster correspondente
4) Repetir os passos 2 e 3 até que algum critério de convergência seja obtido:
   - número máximo de iterações
   - limiar máximo de mudanças nos centróides

Escolher 3 centros iniciais:

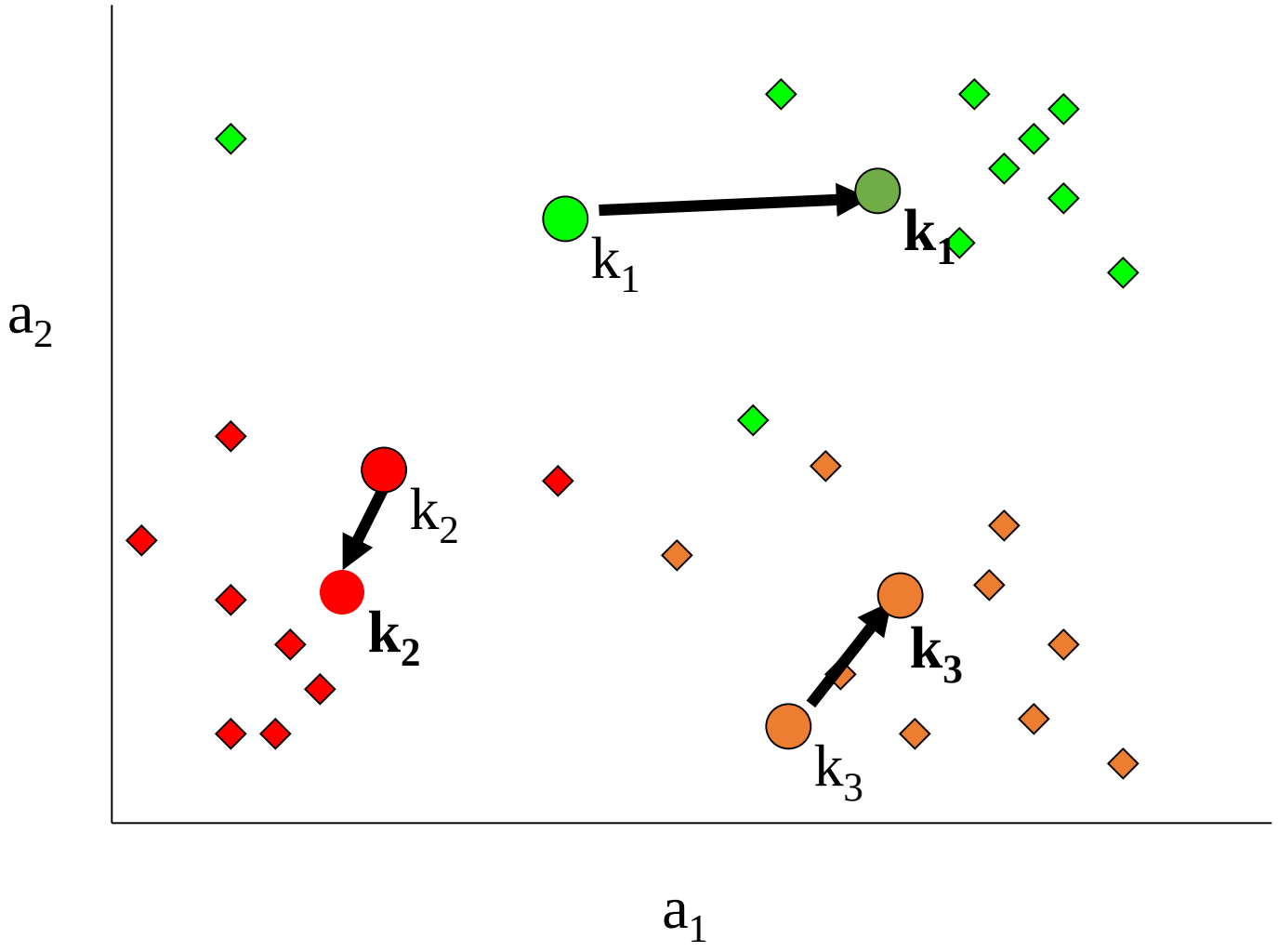


Slides desse exemplo são baseados no curso de Gregory Piatetsky-Shapiro, disponível em http://www.kdnuggets.com

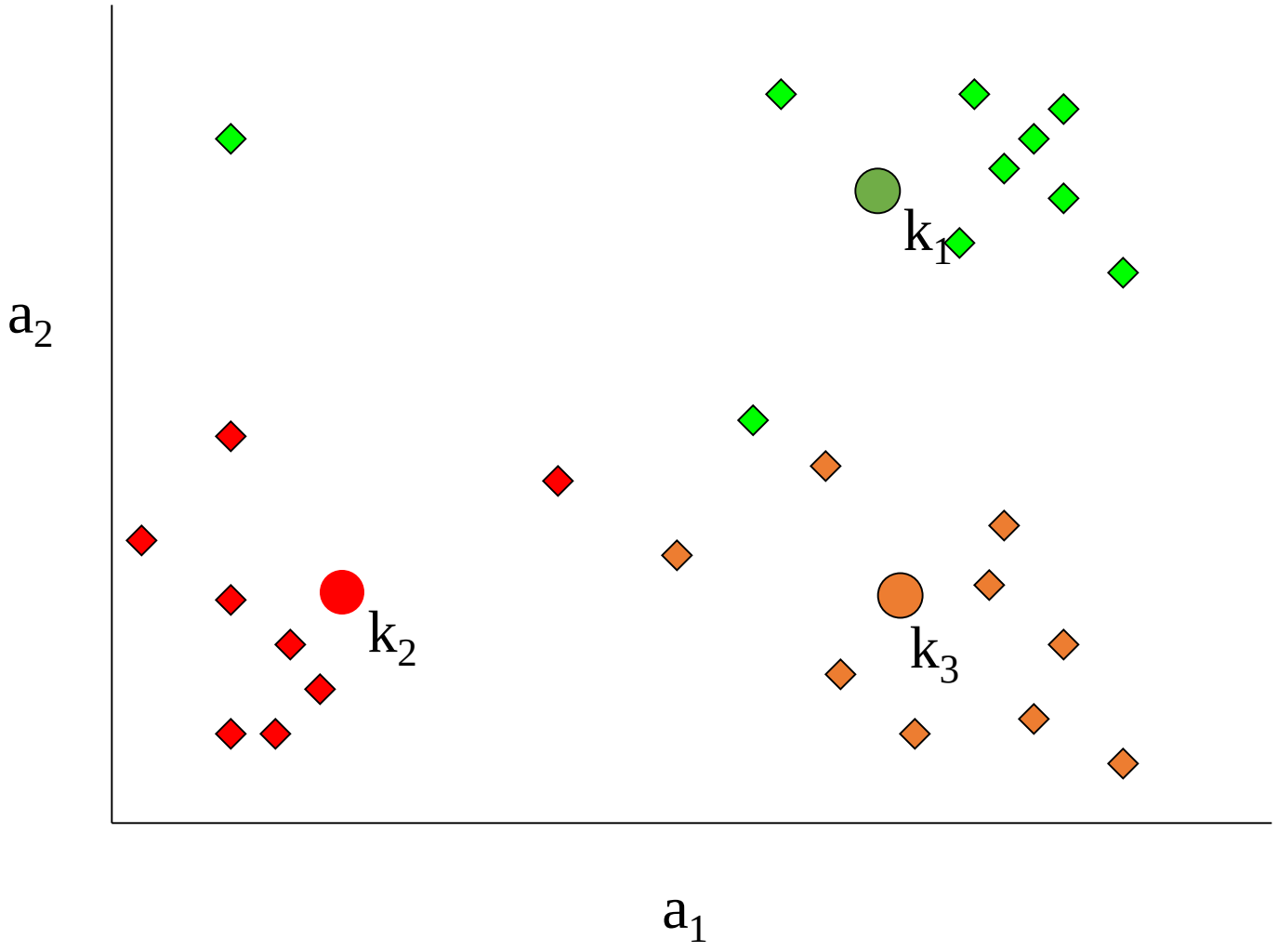
Atribuir cada objeto ao cluster de centro mais próximo:

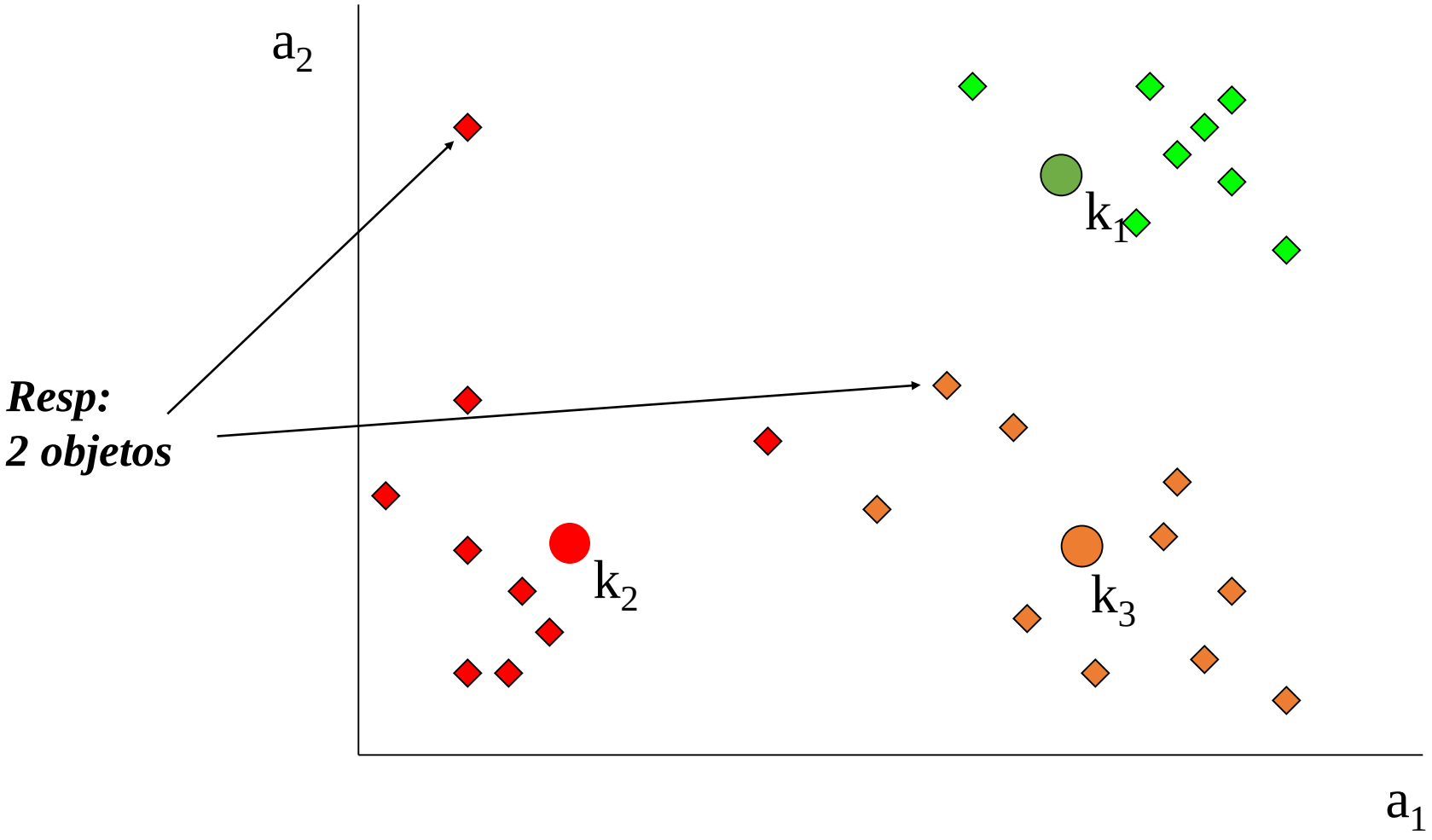


Mover cada centro para o vetor médio do cluster (centróide):

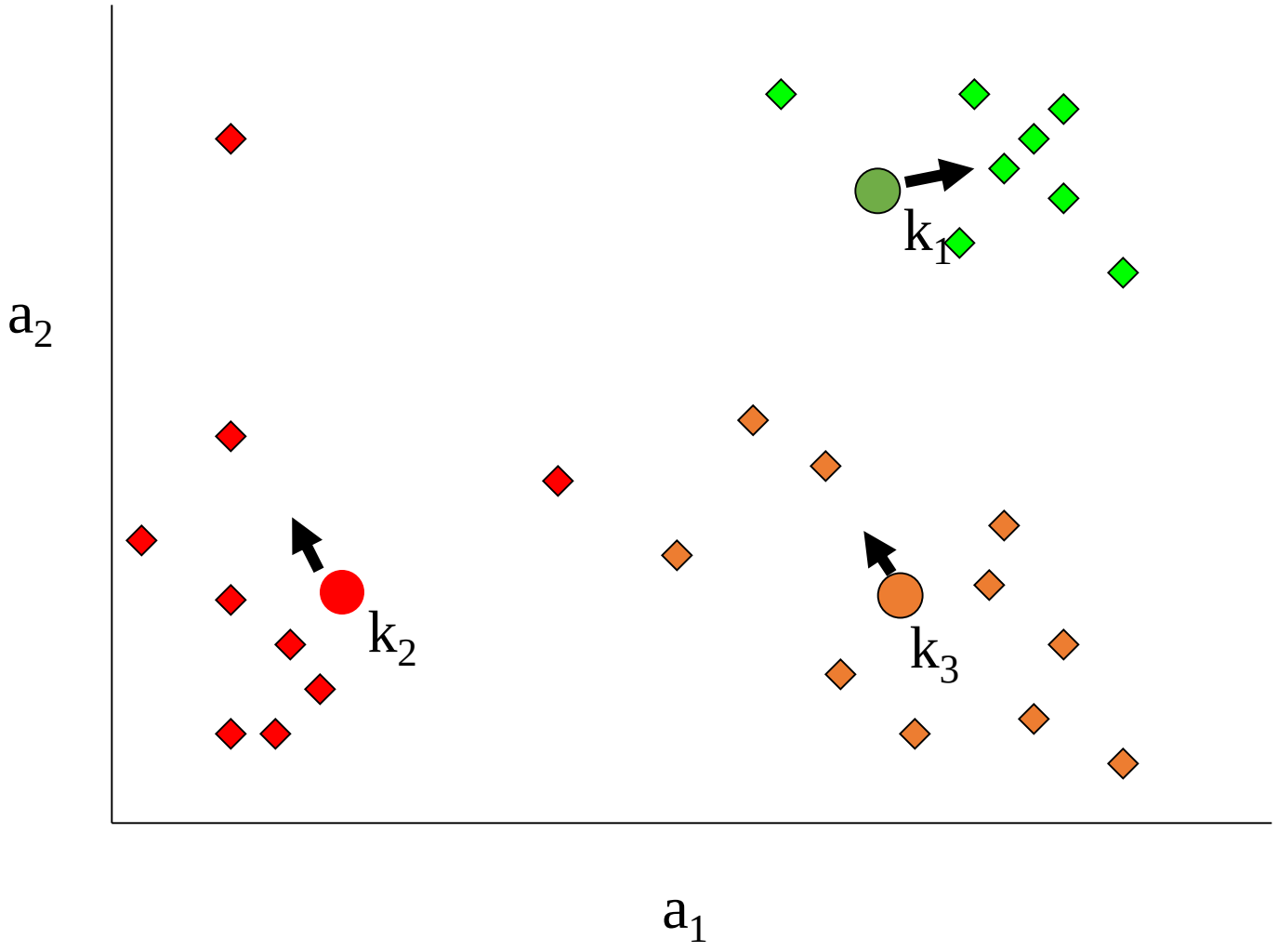


Reatribuir objetos aos clusters mais próximos...
Quais mudarão de cluster?

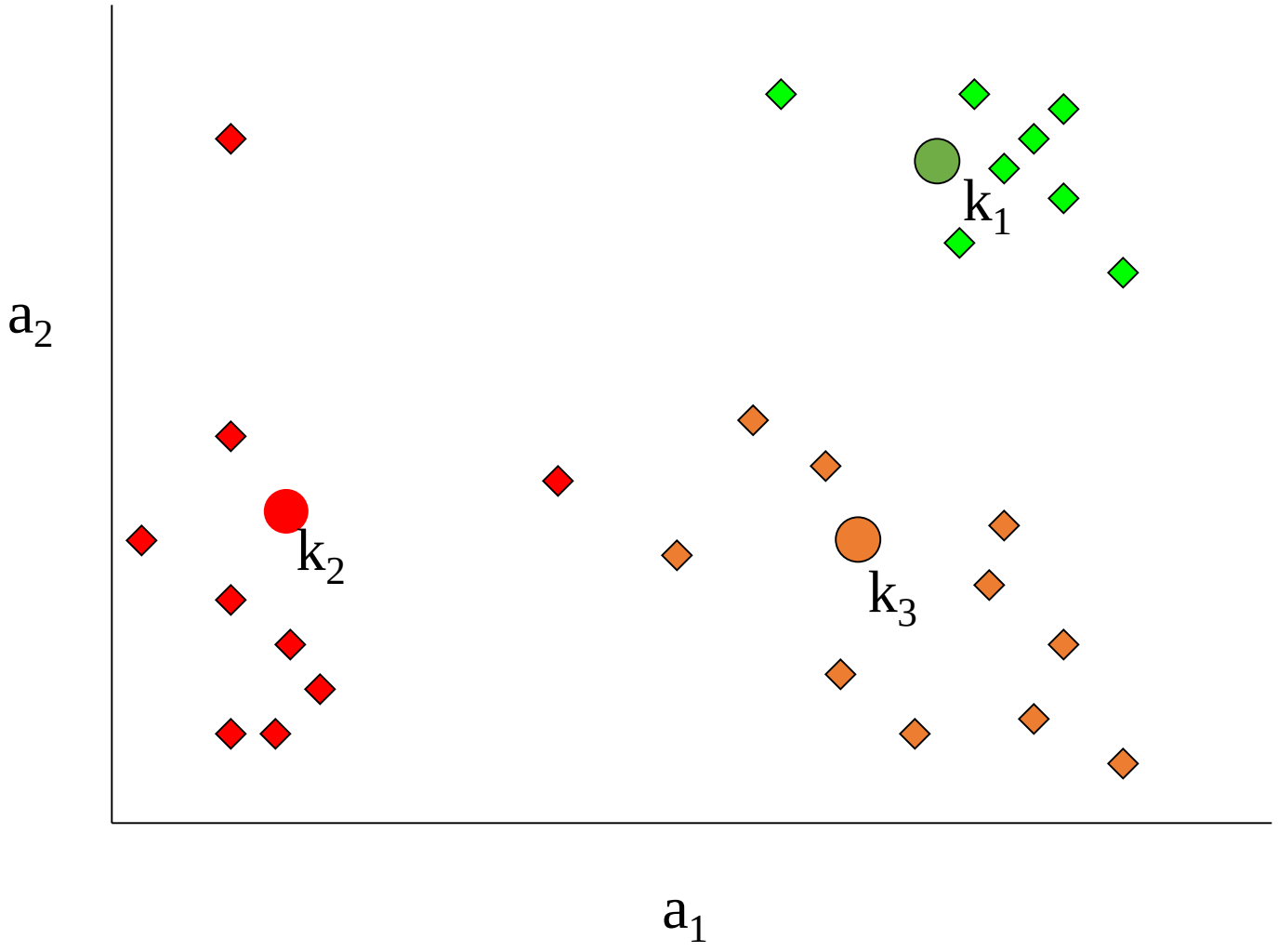



a2
k1
Resp:
2 objetos
k2
k3
a1

Recalcular vetores médios:

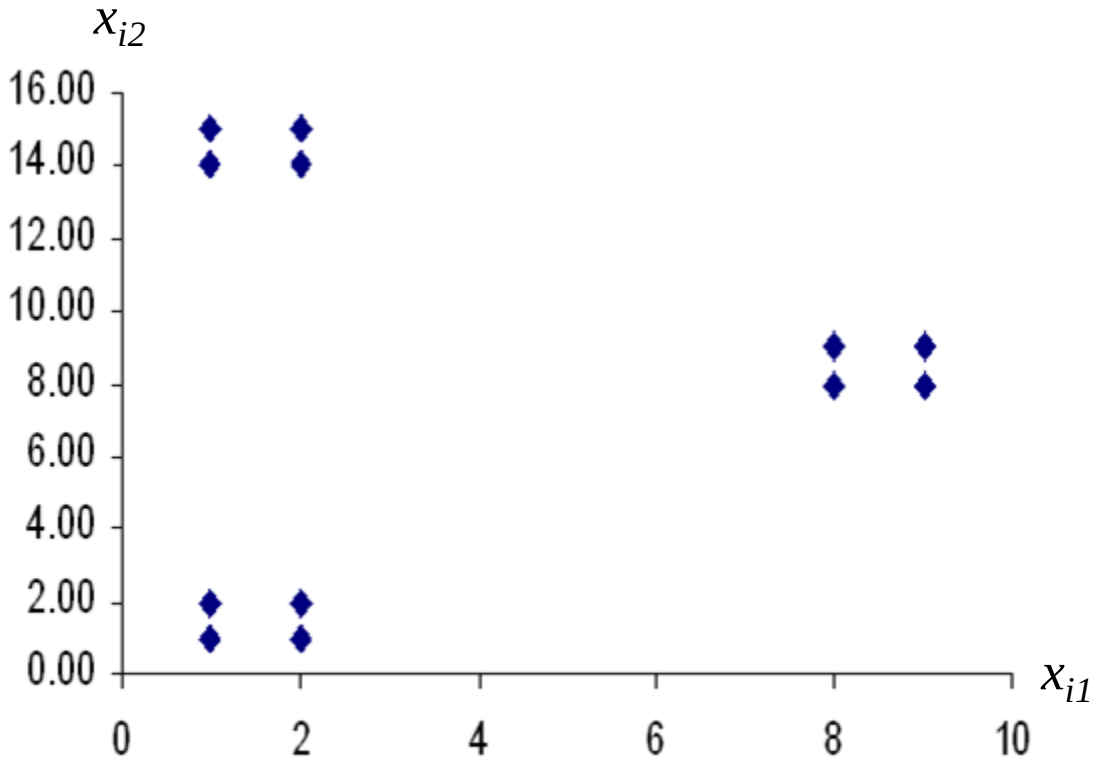


Mover centros dos clusters:



Exercício - Homework

Objeto $\mathbf{x}_i$	x_{i1}	x_{i2}
1	1	2
2	2	1
3	1	1
4	2	2
5	**8**	**9**
6	**9**	**8**
7	**9**	**9**
8	**8**	**8**
9	1	15
10	2	15
11	1	14
12	2	14



Executar k-means com $k = 3$ nos dados acima a partir dos protótipos [6 6], [4 6] e [5 10]. Quais foram as partições e os centróides obtidos?

k-means sob a perspectiva de otimização

Algoritmo minimiza a seguinte função objetivo:

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

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

- Minimizar *J* equivale a minimizar as variâncias intra-cluster.
- Para facilitar o entendimento, vamos reescrever o problema de otimização...

- Consideremos:

  - conjunto de objetos $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$

  - conjunto de k centróides quaisquer $\{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, ..., \bar{\mathbf{x}}_k\}$

- Podemos reescrever o critério SSE de forma equivalente como:

$$J = \sum_{j=1}^{N}\sum_{c=1}^{k} \mu_{cj} \left\| \mathbf{x}_j - \bar{\mathbf{x}}_c \right\|^2 \; ; \; \sum_{c=1}^{k} \mu_{cj} = 1 \;\; \forall j \;\; ; \;\; \mu_{cj} \in \{0,1\}$$

- Desejamos minimizar J com respeito a $\{\bar{\mathbf{x}}_c\}$ e $\{\mu_{cj}\}$

- Pode-se fazer isso via um procedimento iterativo (2 passos):

  a) Fixar $\{\bar{\mathbf{x}}_c\}$ e minimizar J com respeito a $\{\mu_{cj}\}$ (**E**)

  b) Minimizar J com respeito a $\{\bar{\mathbf{x}}_c\}$, fixando-se $\{\mu_{cj}\}$ (**M**)

$$J = \sum_{j=1}^{N}\sum_{c=1}^{k} \mu_{cj} \left\| \mathbf{x}_j - \bar{\mathbf{x}}_c \right\|^2 \; ; \; \sum_{c=1}^{k} \mu_{cj} = 1 \;\; \forall j \;\; ; \;\; \mu_{cj} \in \{0,1\}$$

a) Fixar $\{\bar{\mathbf{x}}_c\}$ e minimizar J com respeito a $\{\mu_{cj}\}$ (**Passo E**)

- Termos envolvendo diferentes j são independentes
- Logo, pode-se otimizá-los separadamente
- μ_{cj} =1 para c que fornece o menor valor do erro quadrático

* **Atribuir μ_{cj} =1 para o grupo mais próximo**.

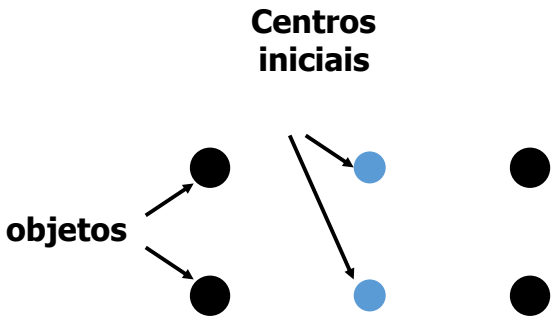
b) Minimizar J com respeito a $\{\bar{\mathbf{x}}_c\}$, fixando-se $\{\mu_{cj}\}$ (**Passo M**)

- Derivar J com respeito a cada $\bar{\mathbf{x}}_c$ e igualar a zero:

$$\nabla_{\bar{\mathbf{x}}_c} J = \sum_{j=1}^{N} \mu_{cj} \nabla_{\bar{\mathbf{x}}_c} \left[\left(\mathbf{x}_j - \bar{\mathbf{x}}_c \right)^T \left(\mathbf{x}_j - \bar{\mathbf{x}}_c \right) \right] = 2 \sum_{j=1}^{N} \mu_{cj} \left(\bar{\mathbf{x}}_c - \mathbf{x}_j \right) = \mathbf{0} \;\; \rightarrow \;\; \boxed{\bar{\mathbf{x}}_c = \frac{\sum_{j=1}^{N} \mu_{cj} \mathbf{x}_j}{\sum_{j=1}^{N} \mu_{cj}}}$$

Sensibilidade em relação à inicialização

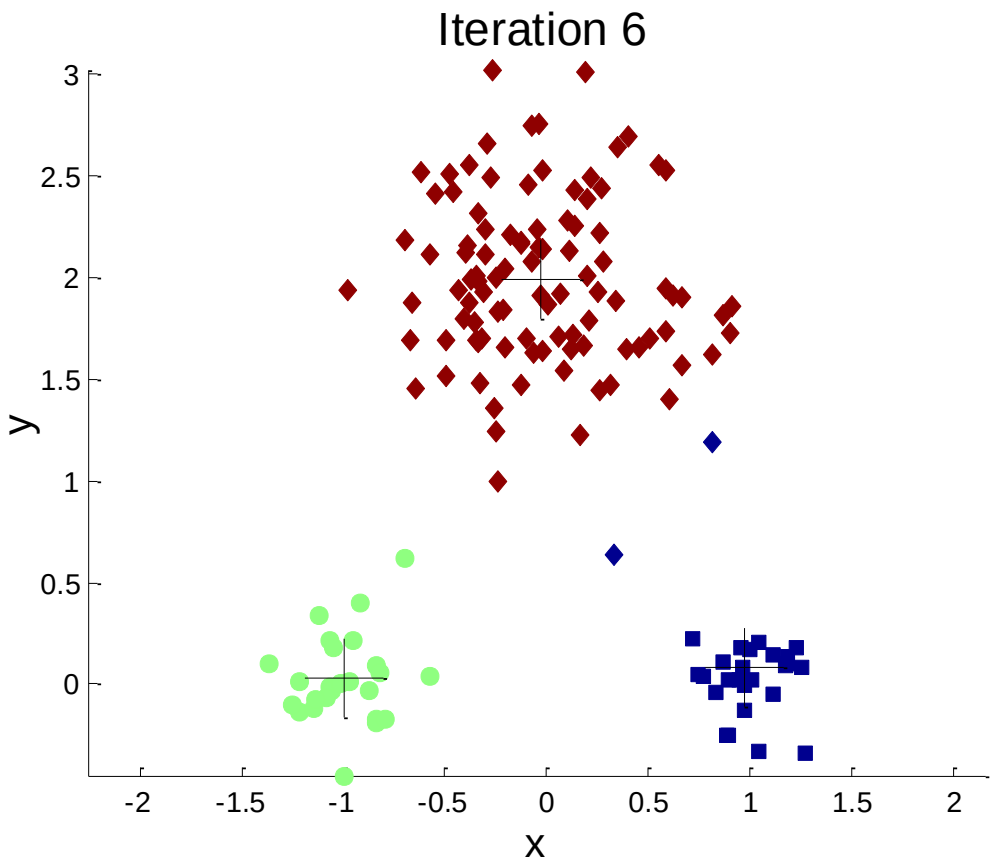
- Resultado pode variar significativamente dependendo da escolha das sementes (protótipos) iniciais
- *k*-means pode "ficar preso" em ótimos locais:



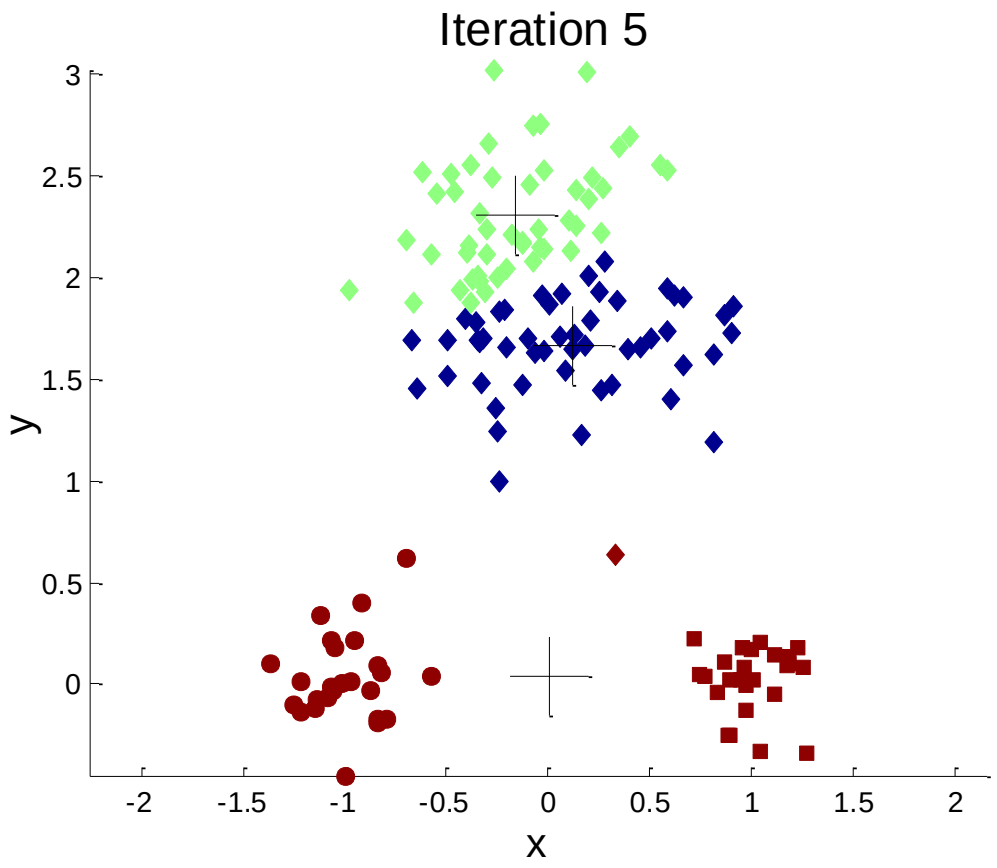
- Como evitar ... ?

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

Exemplos – Inicialização 1



Exemplos – Inicialização 2

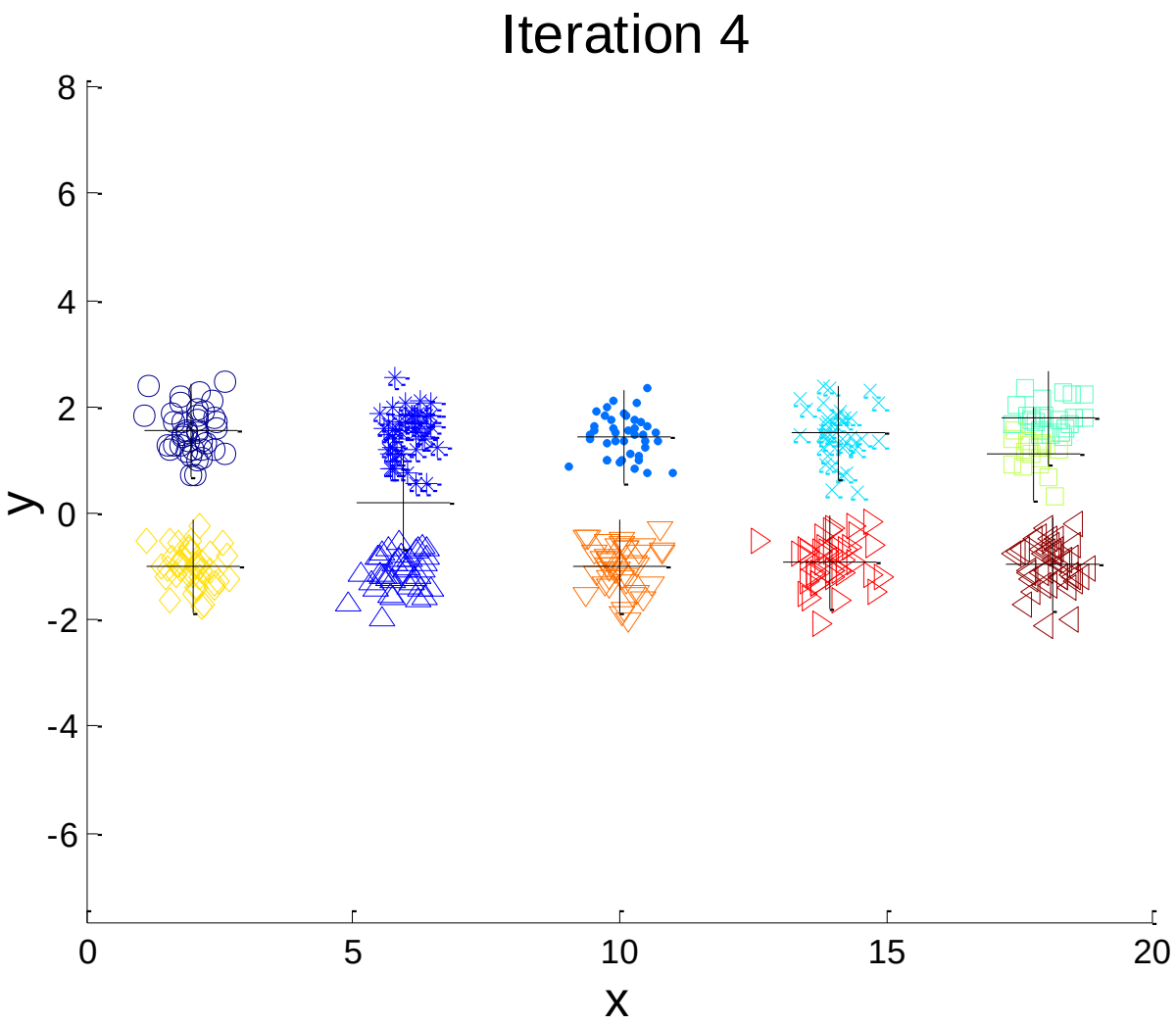


Inicialização (problema)

- **Premissa**: Uma boa seleção de *k* protótipos iniciais em uma base de dados com *k* grupos naturais é tal que cada protótipo é um objeto de um grupo diferente.
- No entanto, a chance de se selecionar um protótipo de cada grupo é pequena, especialmente para *k* grande.
- Consideremos grupos balanceados, com uma mesma quantidade g = N / *k* de objetos cada. A probabilidade de selecionar 1 protótipo de cada grupo diferente é:

$$P = \frac{\text{no. de maneiras de selecionar 1 objeto de cada grupo (N / k objetos)}}{\text{no. de maneiras de selecionar k dentre N objetos}} = \frac{k!}{k^k}$$

Para $k = 10$ temos P = 0.00036.


Iteration 4
8
6
4
2
0
-2
-4
-6
y
0
5
10
15
20
x

Como lidar com o problema?

- Múltiplas Execuções (inicializações aleatórias):
  - Funciona bem em muitos problemas;
  - Pode demandar muitas execuções (especialmente com *k* alto).

- Agrupamento Hierárquico: agrupa-se uma amostra dos dados para tomar os centros da partição com *k* grupos.

- Seleção "informada" em uma amostra dos dados:
  - Tomar o 1º protótipo como um objeto aleatório ou como o centro dos dados (*grand mean*);
  - Sucessivamente escolhe-se o próximo protótipo como o objeto mais distante dos protótipos correntes.

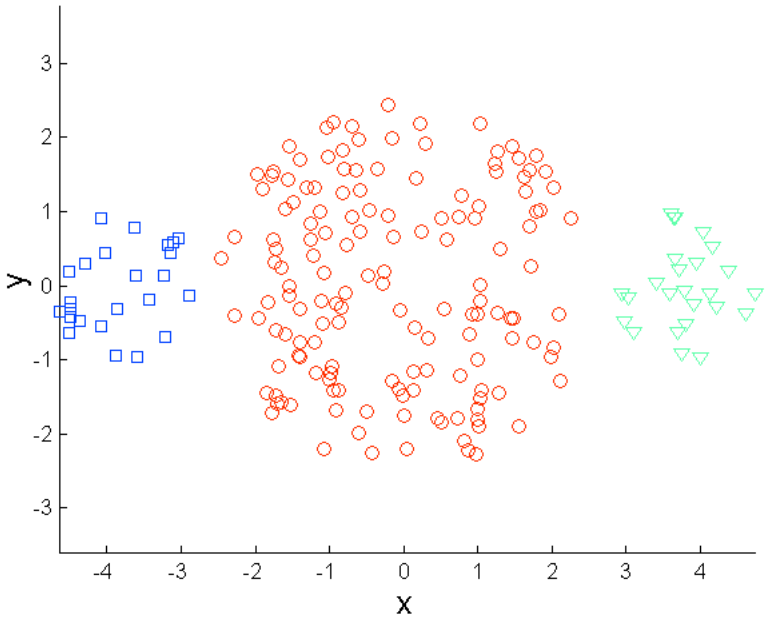
- Busca Guiada: X-means, *k*-means evolutivo, ...

Problemas estruturais

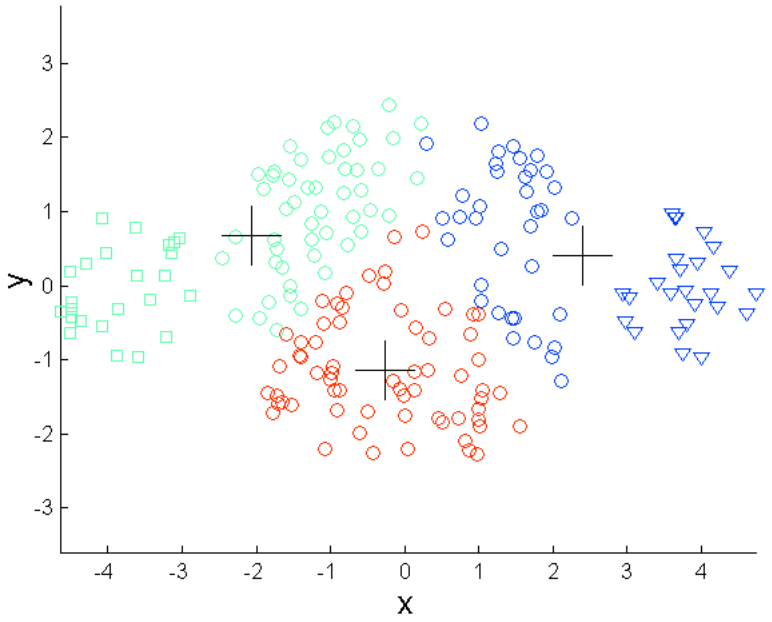
Algoritmo *k*-means funciona bem se:

- Clusters são (hiper)esféricos e bem separados
- Clusters de volumes aproximadamente iguais
- Cluster com quantidades de pontos semelhantes

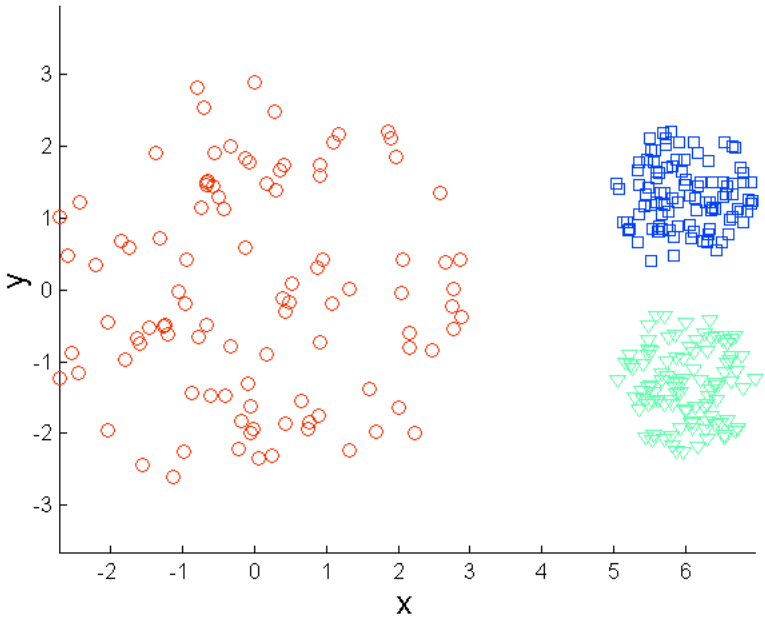
➢ Vejamos alguns exemplos ilustrativos de problemas...



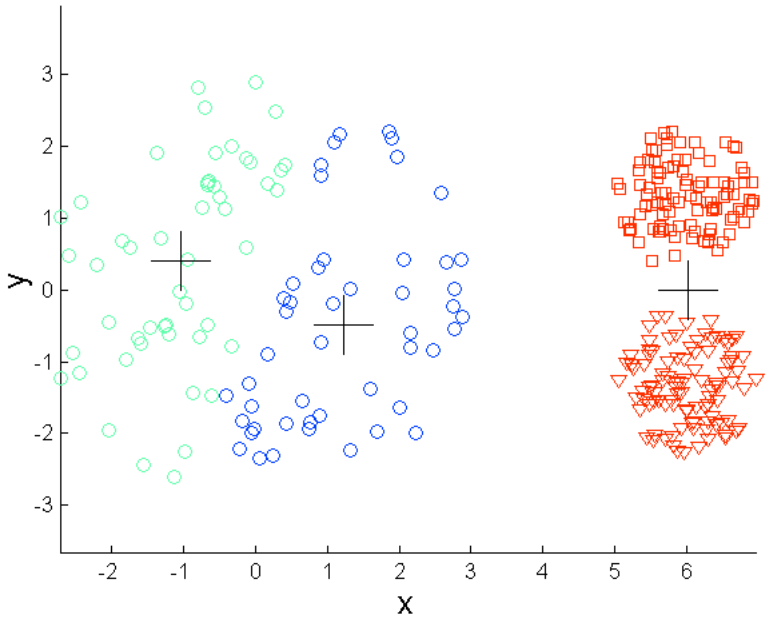
Estrutura correta



k-means (3 Clusters)

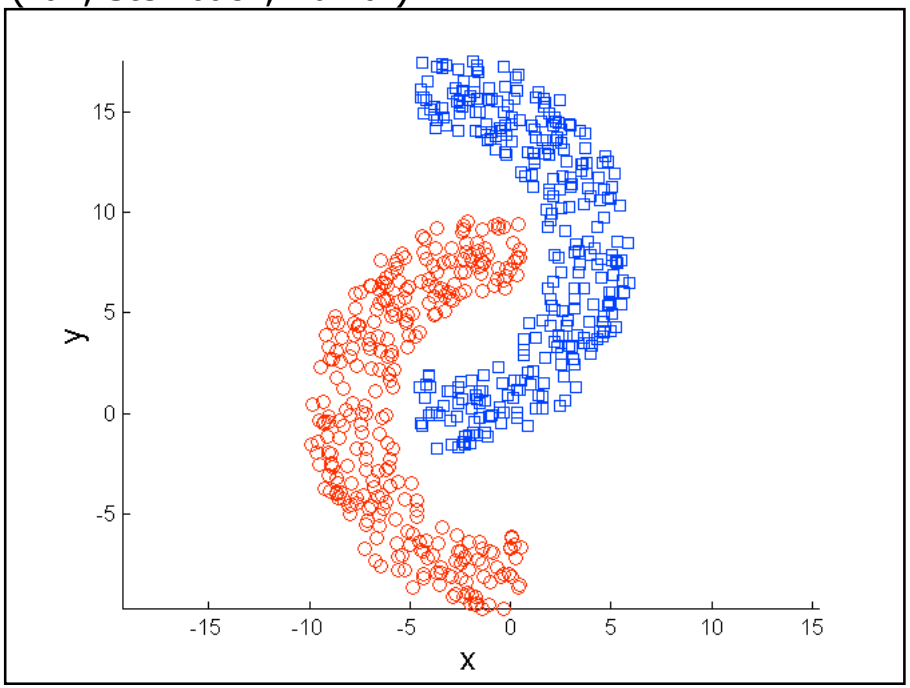


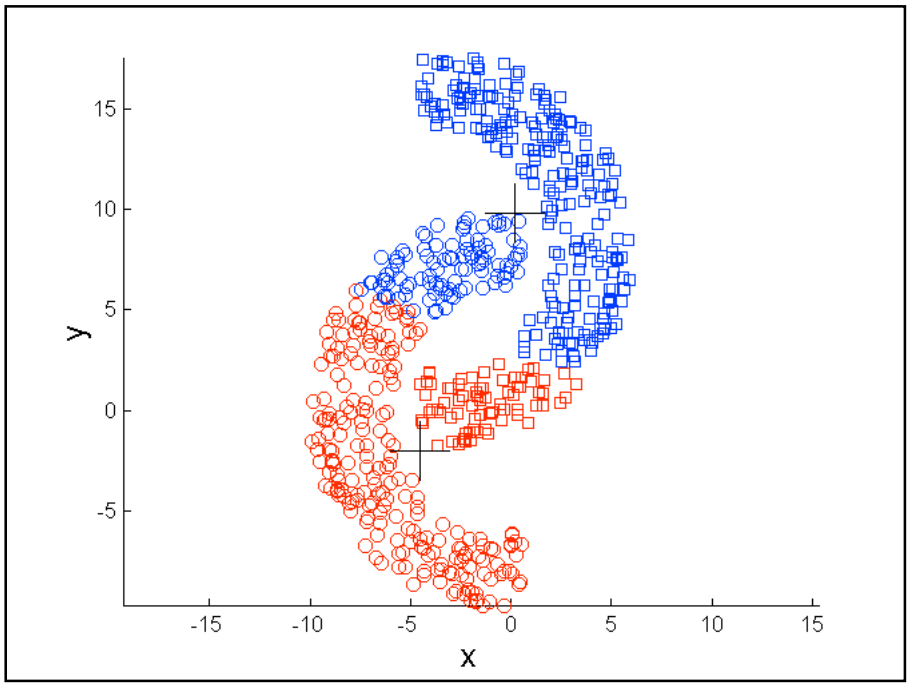
Estrutura correta



K-means (3 Clusters)

(Tan, Steinbach, Kumar)





Nota: na prática, esse problema em geral não é crítico, i.e., há pouco interesse na maioria das aplicações de mundo real.

Custo computacional

- **Complexidade (assintótica) de tempo:**

$$O(i \cdot K \cdot N \cdot n)$$

- O que isso significa?

O que dizer sobre a constante de tempo?

→ Computar Distância Euclidiana via aproximações sucessivas (Newton-Raphson) custa caro.

Se também tenho problema de espaço em memória...

→ Solução aproximada (*sampling*)

→ Paralelizar (mesmo computador) ou distribuir (e.g., map-reduce) o processamento.

Implementações eficientes

- Desempenho computacional pode ser melhorado:
  - **Estruturas de Dados**, e.g. kd-trees
  - **Algoritmos**:
    - **Atualização recursiva dos centróides**

      Cálculo dos centróides só depende dos valores anteriores, dos nos. de objetos dos grupos e dos objetos que mudaram de grupo

      Exercício: a partir da equação do cálculo do centróide, escrever a equação de atualização recursiva descrita acima.
    - **Uso da desigualdade triangular**
    - **Paralelização** (vide discussão a seguir)

Algoritmo *k*-means paralelo e/ou distribuído

Dados distribuídos em múltiplos *data sites* ou processadores

- **Algoritmo**:
  - Mesmos protótipos iniciais são distribuídos a cada sítio de dados
  - Cada sítio executa (em paralelo) uma iteração de *k*-means
  - Protótipos locais e nos. de objetos dos grupos são comunicados
  - Protótipos globais são calculados e retransmitidos aos sítios
  - Repete-se o processo

Dhillon and Modha, A Data Clustering Algorithm on Distributed Memory Multiprocessors, LNAI 1759, 2000.

Resumo das (des)vantagens do *k*-means

Vantagens

- Simples e intuitivo
- Complexidade **linear** em todas as variáveis críticas
- Eficaz em muitos cenários de aplicação
- Resultados de interpretação simples

Desvantagens

- k = ?
- Sensível à inicialização dos protótipos (mínimos locais de J)
- Limita-se a encontrar clusters volumétricos / globulares
- Cada item deve pertencer a um único cluster (**partição rígida**)
- Limitado a atributos numéricos
- Sensível a *outliers*

Agenda

- Motivação e conceitos
- Definições preliminares
- k-means
- Estimando o número de clusters a partir dos dados
- Bisecting k-means
- k-medoids
- EM para misturas de Gaussianas
- Avaliação de agrupamentos

Executar *k*-means múltiplas vezes

Rodar *k*-means repetidas vezes a partir de diferentes valores de *k* e de posições iniciais dos protótipos:

Ordenado: n_p inicializações para cada $k \in [k_{min}, k_{max}]$

Aleatório: n_T inicializações com k sorteado em $[k_{min}, k_{max}]$

Tomam a melhor partição resultante de acordo com algum critério de qualidade (**critério de validade de agrupamento**)

- **Vantagens:** estimam k e são menos sensíveis a mínimos locais
- **Desvantagem:** custo computacional pode ser elevado

Poderíamos usar *J* para estimar k* ?

- Sim se todas as partições têm o mesmo *k* (fixo).

- E se *k** for desconhecido e, portanto, variável ?

Para responder, considere, por exemplo, que as partições são geradas a partir de múltiplas execuções do algoritmo:

- Com protótipos iniciais aleatórios

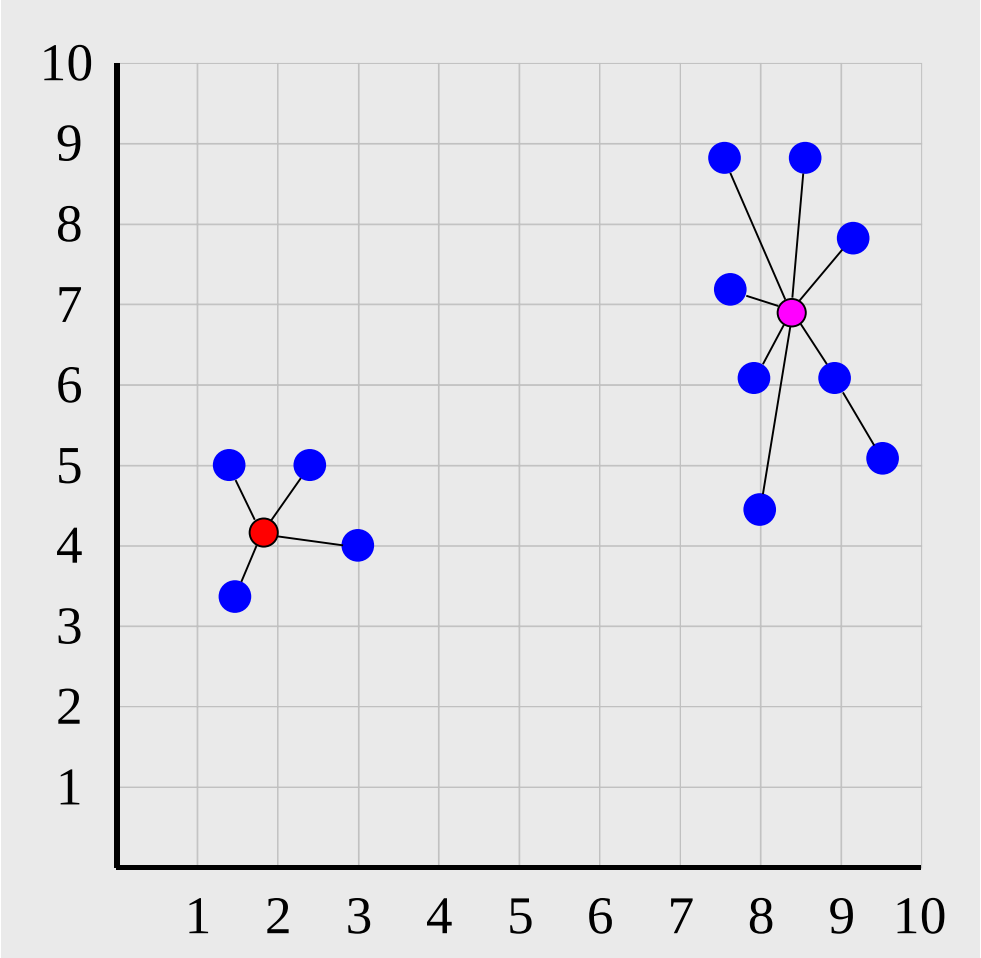
- Com no. variável de grupos $k \in [k_{min}, k_{max}]$

- Vejamos um exemplo ilustrativo...

Erro Quadrático:

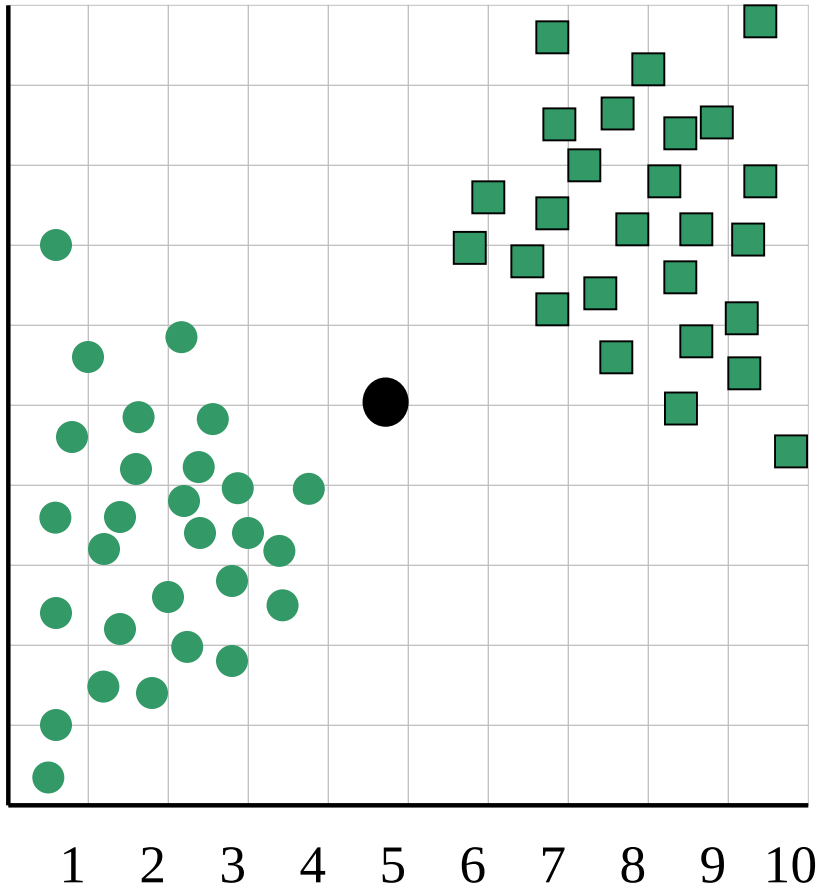
$$J = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in \mathbf{C}_i} d\left(\mathbf{x}_j, \bar{\mathbf{x}}_i\right)^2$$

Função Objetivo

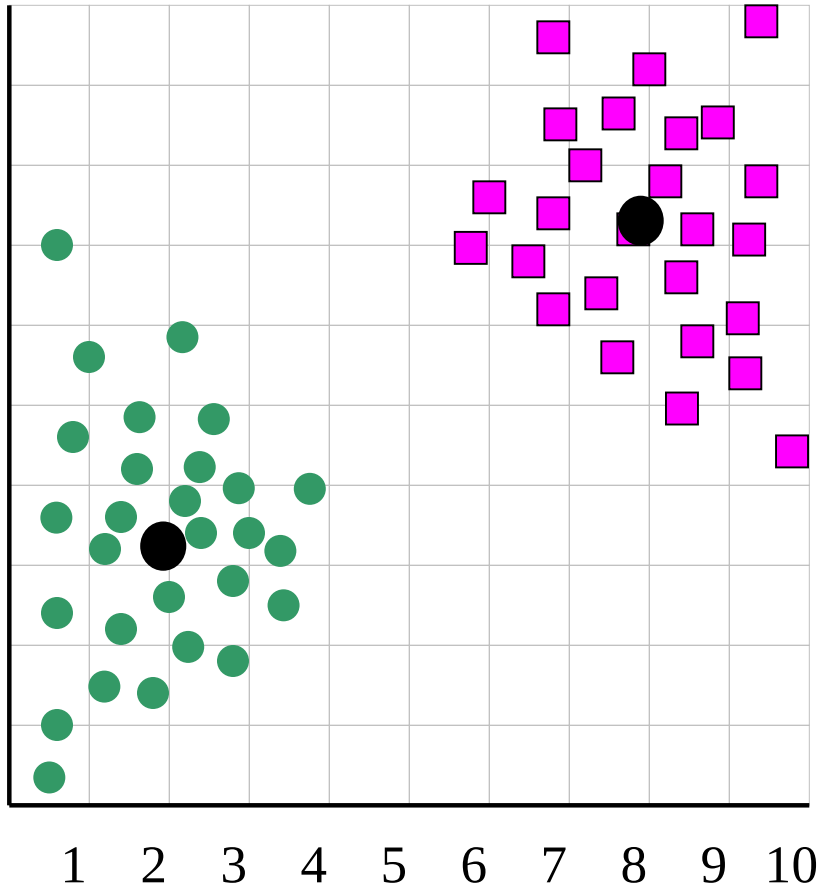


Exemplo de Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Manaus.

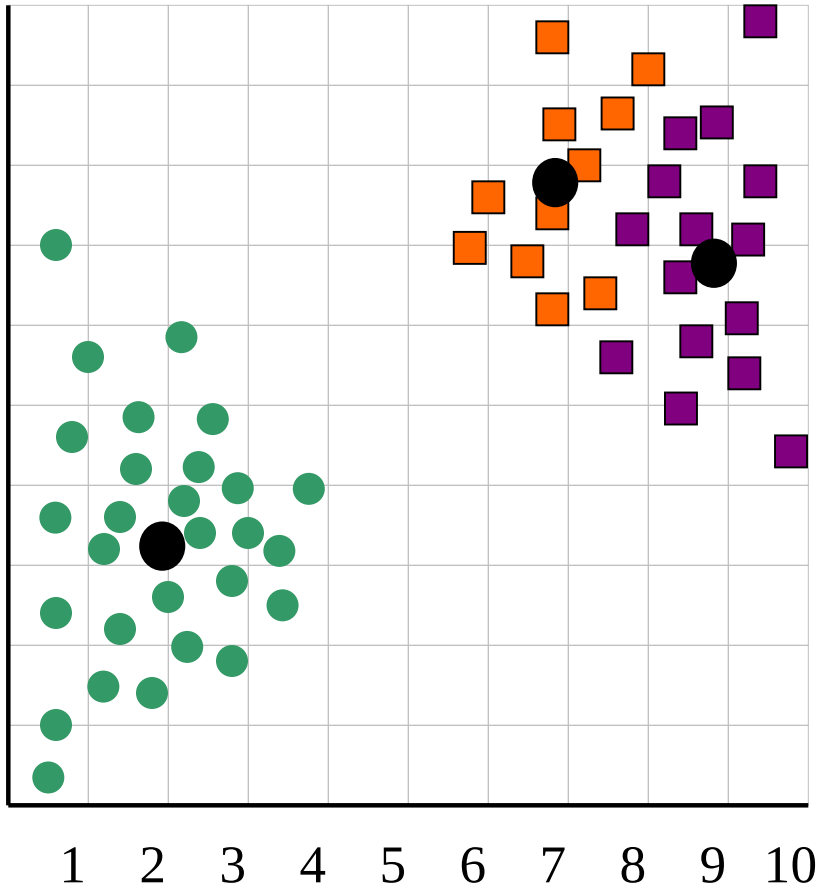
Para $k = 1$, o valor da função objetivo é 873



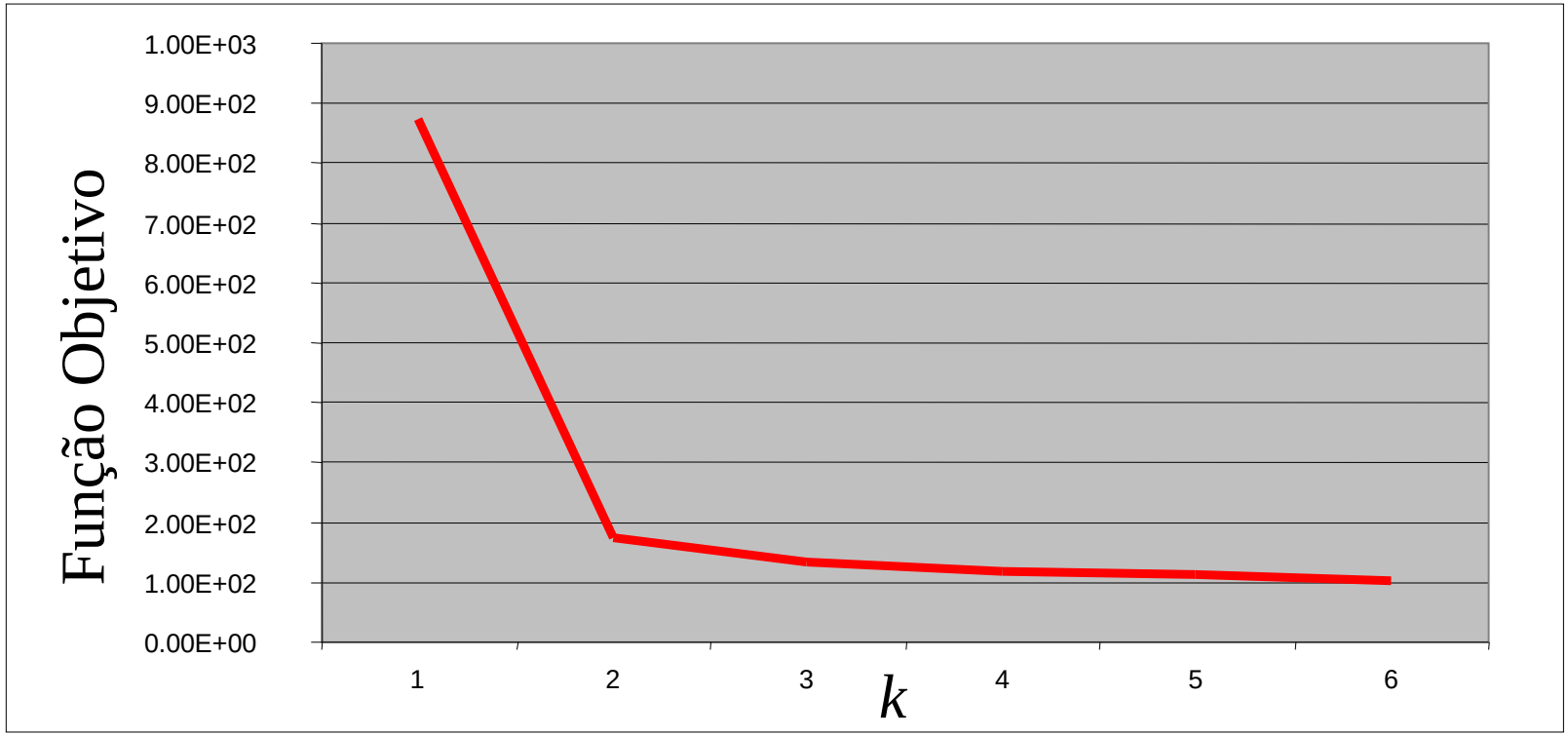
Para $k = 2$, o valor da função objetivo é 173



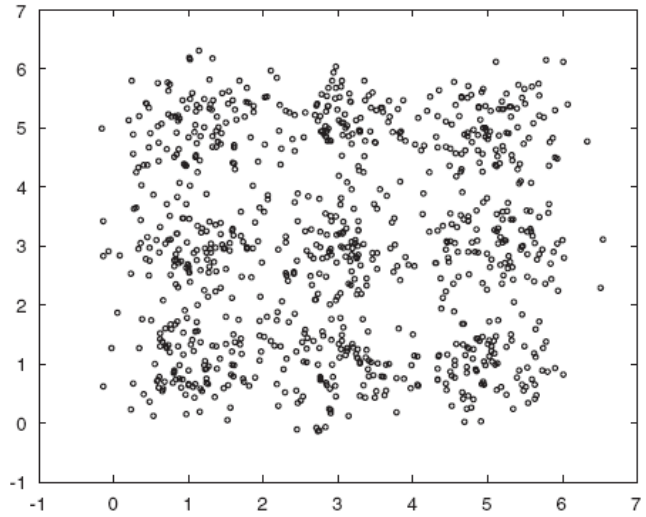
Para $k = 3$, o valor da função objetivo é 134

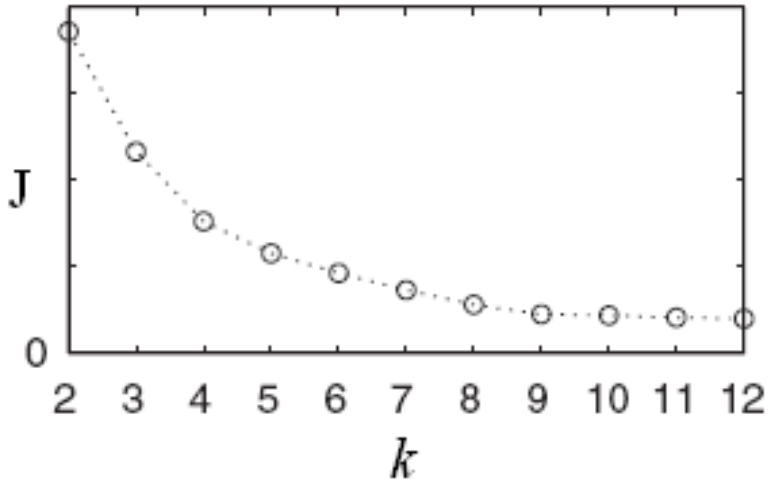


Podemos então repetir este procedimento e plotar os valores da função objetivo J para k =1,…,6, ... e tentar identificar um “joelho” :



- Infelizmente os resultados não são sempre tão claros quanto no exemplo anterior:



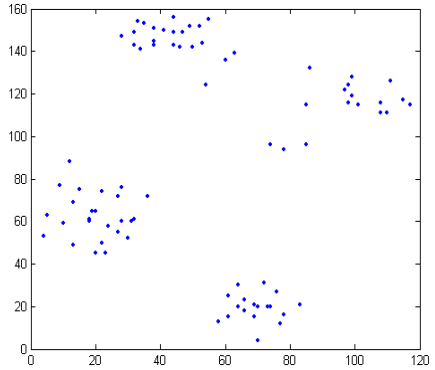


- Outras alternativas para lidar com o problema de se estimar o número de clusters?

➢ Índices de validade relativos...

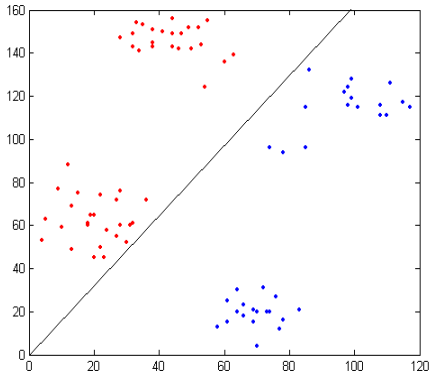
Critérios de validade relativos

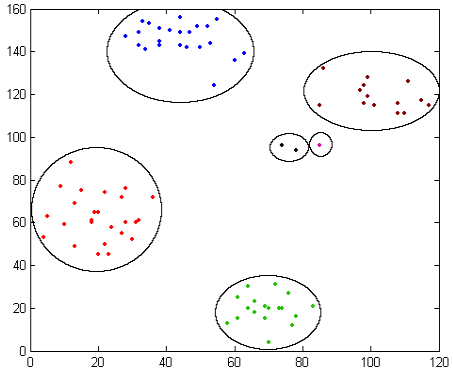
A aplicação de um ou mais algoritmos usualmente retorna múltiplas soluções que precisam ser comparadas:



Base de Dados

Algoritmo(s) de Agrupamento





Soluções

Precisamos de critérios objetivos para compará-las:

- Produzir uma ordenação de um conjunto de partições de acordo com suas avaliações
- Índices numéricos de validade relativos. Vejamos um deles...

Critério da silhueta

SWC = Silhueta média sobre todos os objetos: $SWC = \frac{1}{N}\sum_{i=1}^{N} s(i)$

Silhueta (i-ésimo objeto): $s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$ (s(i) := 0 para singletons)

***a(i)*: dissimilaridade média do i-ésimo objeto ao seu cluster**

***b(i)*: dissimilaridade média do i-ésimo objeto ao cluster vizinho mais próximo**

Silhueta Original: $a(i)$ e $b(i)$ são calculados como a distância média (Euclidiana, Mahalanobis etc.) do i-ésimo objeto a todos os demais objetos do cluster em questão - $O(N^2)$.

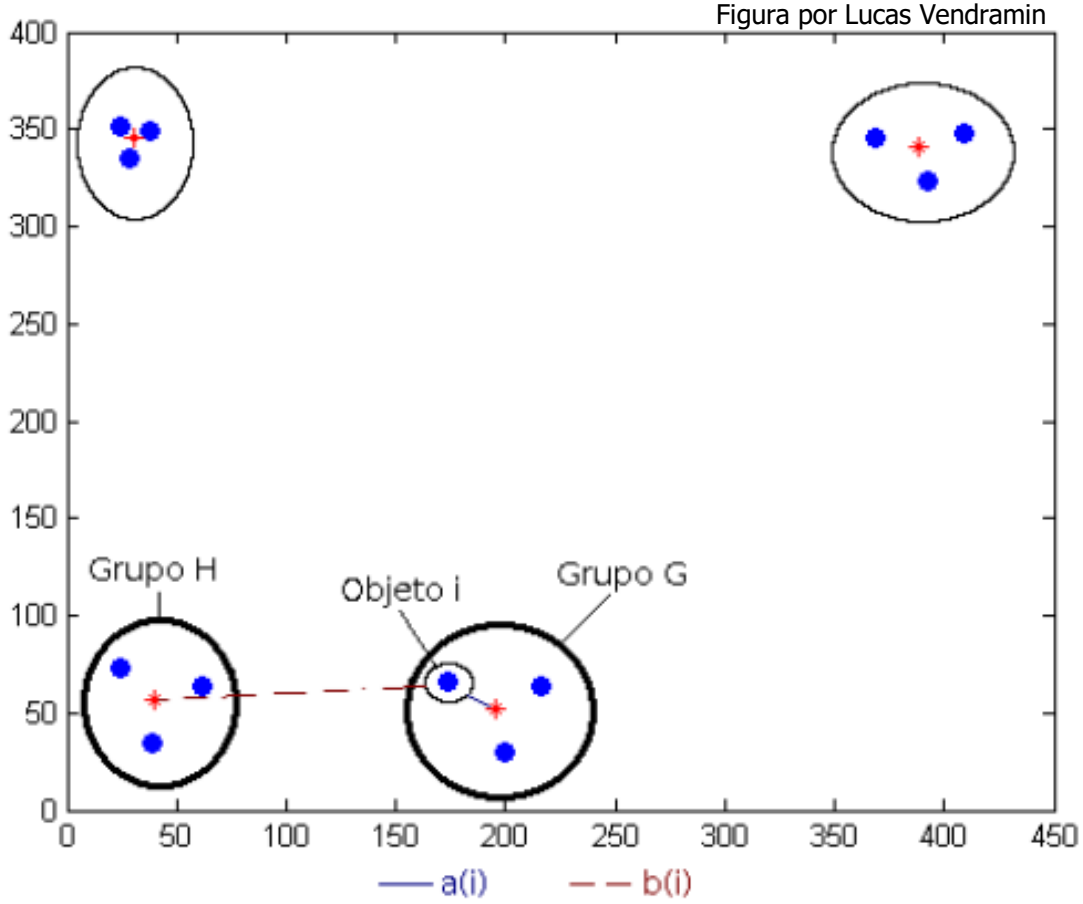
Propriedade Favorável: SWC $\in$ [–1 ,+1]

Silhueta simplificada

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$$SWC = \frac{1}{N} \sum_{i=1}^{N} s(i)$$

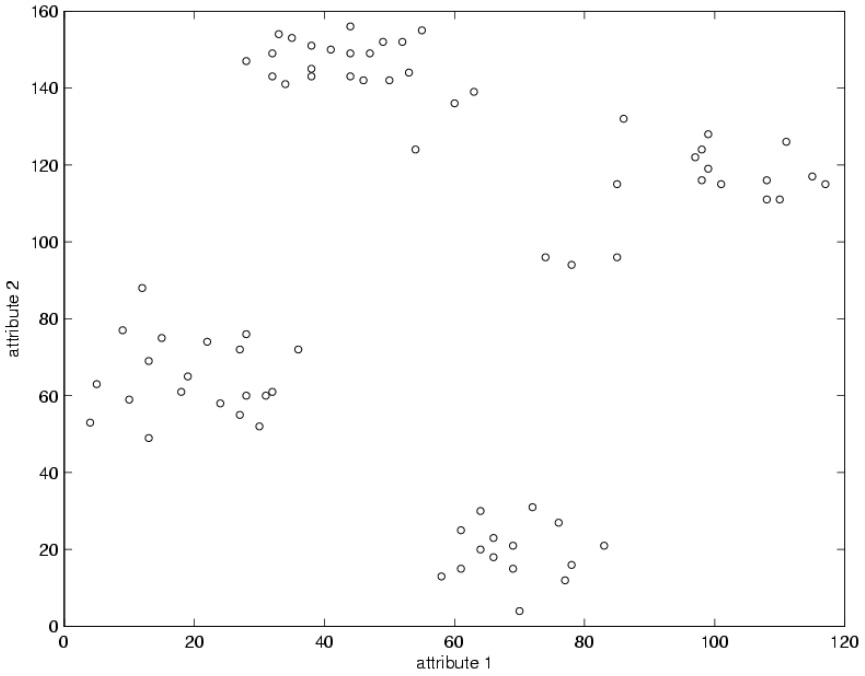
Figura por Lucas Vendramin

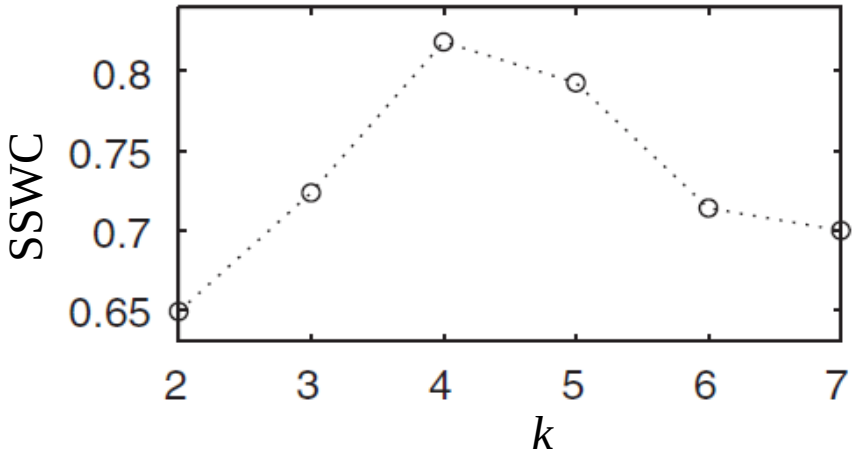


Silhueta Simplificada: $a(i)$ e $b(i)$ são calculados como a distância do i-ésimo objeto ao centróide do cluster em questão - O(N).

Exemplo:

- Relembrando a subjetividade do problema:
  - Quantos grupos abaixo?
  - Sob a perspectiva deste **critério** (SSWC) temos:





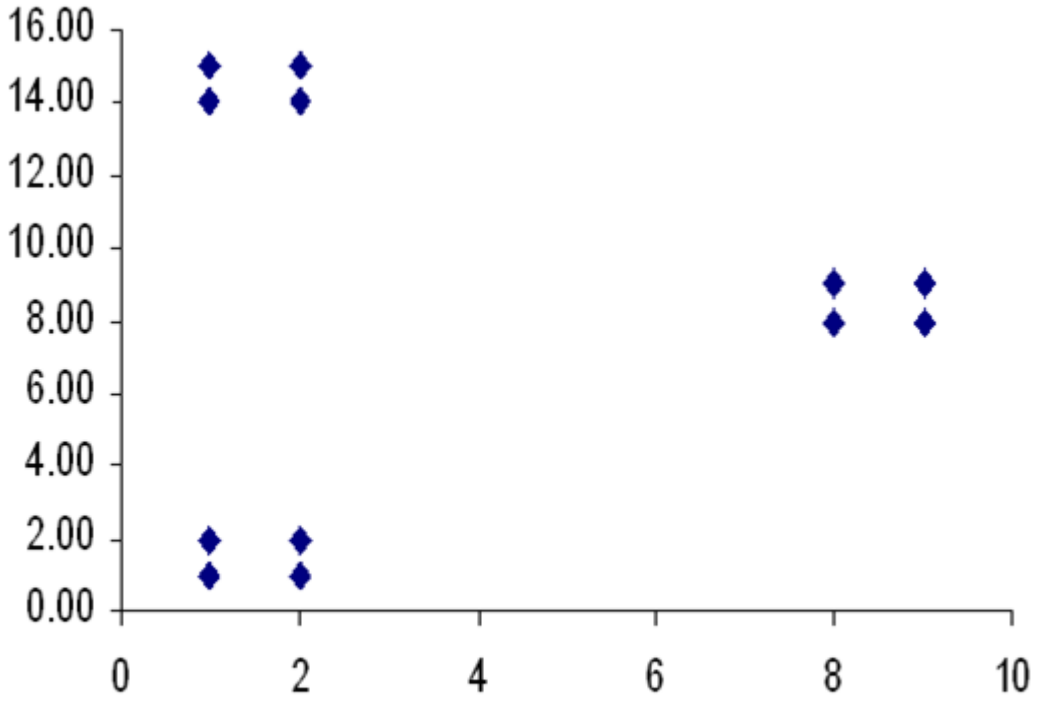
Existem vários outros critérios

-	Criterion	Complexity
	Calinski-Harabasz (VRC)	$O(nN)$
	Davies-Bouldin (DB)	$O(n(k^2+N))$
	Dunn	$O(nN^2)$
	Silhouette Width Criterion (SWC)	$O(nN^2)$
	Alternative Silhouette (ASWC)	$O(nN^2)$
	Simplified Silhouette (SSWC)	$O(nNk)$
	Alternative Simplified Silhouette (ASSWC)	$O(nNk)$
	PBM	$O(n(k^2+N))$
	C-Index	$O(N^2(n+log_2N))$
	Gamma	$O(nN^2+N^4/k])$
	G(+)	$O(nN^2+N^4/k])$
	Tau	$O(nN^2+N^4/k])$
	Point-Biserial	$O(nN^2)$
	$C/\sqrt{k}$	$O(nN)$
*	Trace(W)	$O(nN)$
*	Trace(CovW)	$O(nN)$
*	Trace($W^{-1}B$)	$O(n^2N+n^3)$
*	|T|/|W|	$O(n^2N+n^3)$
*	Nlog(|T|/|W|)	$O(n^2N+n^3)$
*	k^2W	$O(n^2N+n^3)$
*	log(SSB/SSW)	$O(n(k^2+N))$
*	Ball-Hall	$O(nN)$
*	McClain-Rao	$O(nN^2)$

Vendramin, Campello, Hruschka "*Relative Clustering Validity Criteria: A Comparative Overview*" **Statistical Analysis and Data Mining**, Vol. 3, p. 209-235, 2010.

Exercício

Objeto $\mathbf{x}_i$	x_{i1}	x_{i2}
1	1	2
2	2	1
3	1	1
4	2	2
5	**8**	**9**
6	**9**	**8**
7	**9**	**9**
8	**8**	**8**
9	1	15
10	2	15
11	1	14
12	2	14



- Calcule o valor para as silhuetas para a partição *correta* acima e também para uma partição formada por dois clusters à sua escolha.

Agenda

- Motivação e conceitos
- Definições preliminares
- k-means
- Estimando o número de clusters a partir dos dados
- Bisecting k-means
- k-medoids
- EM para misturas de Gaussianas
- Avaliação de agrupamentos

Bisecting *k*-means (particional-hierárquico):

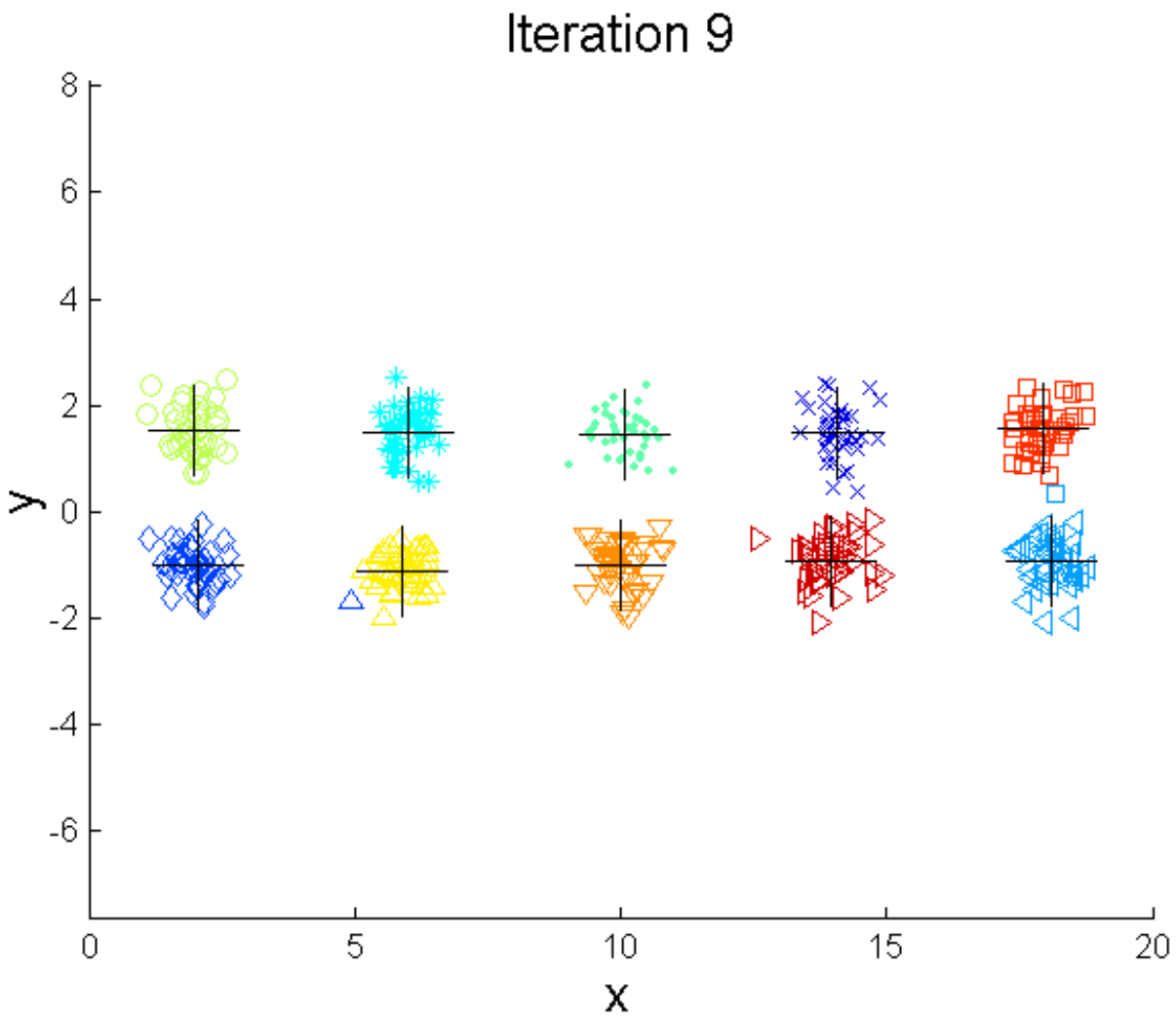
Recursivamente particiona a base de dados em dois grupos, gerando uma "árvore de partições". Lembrar que:

$$P = \frac{\text{no. de maneiras de selecionar 1 objeto de cada grupo (N / k objetos)}}{\text{no. de maneiras de selecionar k dentre N objetos}} = \frac{k!}{k^k}$$

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

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

Exemplo:



Notas sobre 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, por exemplo usando outro critério de avaliação de qualidade dos grupos, para eleger o "pior":
  - Diâmetro máximo (sensível a *outliers*)
  - SSE normalizado pelo no. de objetos do grupo (mais robusto)

Complexidade computacional

- *k*-means roda em O(Nkn)*. Para *k* = 2 tem-se O(Nn). Presumindo que *no_of_iterations* = 1 no passo 4 temos:

➢ **Pior Caso**: cada divisão separa apenas 1 objeto dos demais

– O(Nn + (N-1)n + (N-2)n + ... + 2n) → **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(nN $\log_2$N)**

* Assumindo distância com complexidade linear no no. de atributos

Agenda

- Motivação e conceitos
- Definições preliminares
- k-means
- Estimando o número de clusters a partir dos dados
- Bisecting k-means
- k-medoids
- EM para misturas de Gaussianas
- Avaliação de agrupamentos

k-medoids

- Substituir centróide por um objeto representativo (*medoid*);
- Medoid é o objeto mais próximo aos demais objetos do grupo - mais próximo em média (empates resolvidos aleatoriamente);

- Menos sensível a *outliers*;
- permite cálculo relacional (requer apenas matriz de distâncias);
- Pode ser aplicado a bases com atributos categóricos;
- Converge com qualquer medida de (dis)similaridade
- Complexidade quadrática com nº. de objetos (N)

Agenda

- Motivação e conceitos
- Definições preliminares
- k-means
- Estimando o número de clusters a partir dos dados
- Bisecting k-means
- k-medoids
- EM para misturas de Gaussianas
- Avaliação de agrupamentos

EM para mistura de Gaussianas

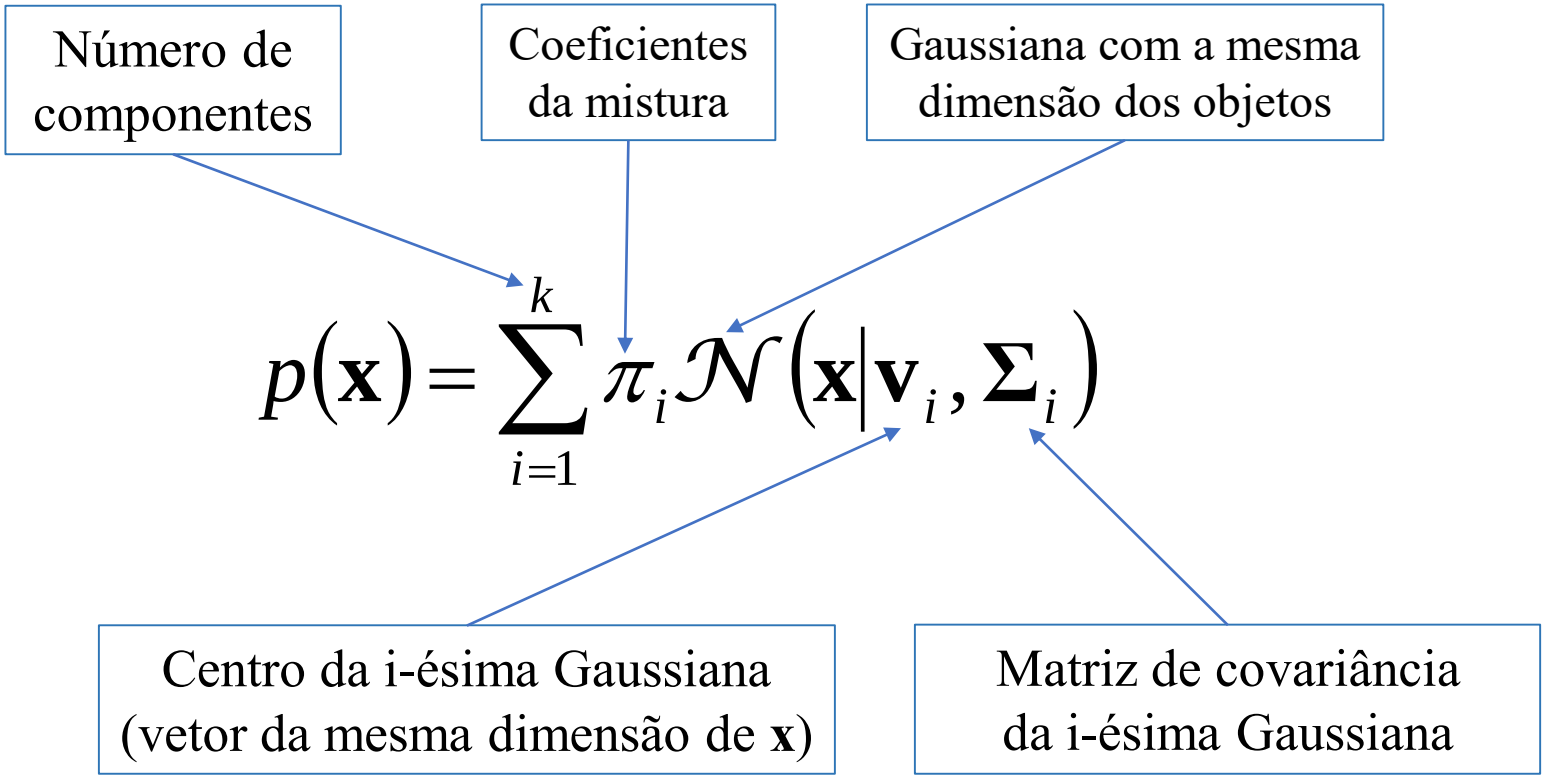
- O Algoritmo **EM** (Expectation Maximization) é um procedimento genérico para a modelagem probabilística de um conjunto de dados;

- Basicamente, **EM** otimiza os parâmetros de uma função de distribuição de probabilidades (p.d.f.) de forma que esta represente os dados da forma mais verossímil possível;

- Modelo mais utilizado: **Mistura de Gaussianas**

GMM (*Gaussian Mixture Model*)

Um GMM é representado pela *p.d.f* :

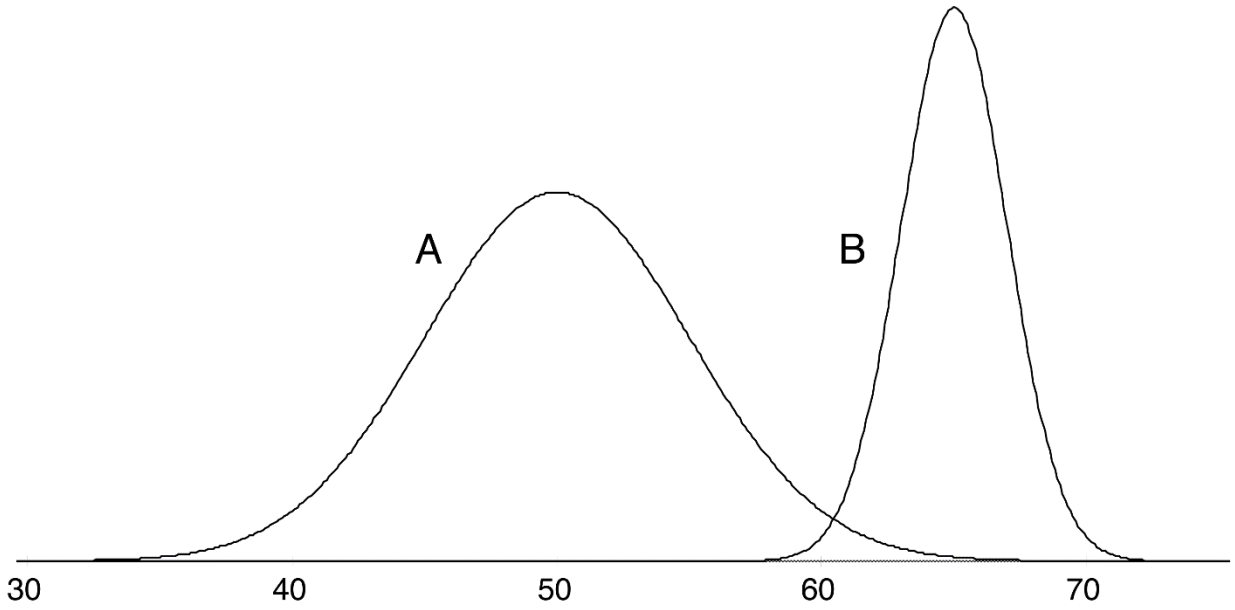


Exemplo

Objetos:

A	51	B	62	B	64	A	48	A	39	A	51
A	43	A	47	A	51	B	64	B	62	A	48
B	62	A	52	A	52	A	51	B	64	B	64
B	64	B	64	B	62	B	63	A	52	A	42
A	45	A	51	A	49	A	43	B	63	A	48
A	42	B	65	A	48	B	65	B	64	A	41
A	46	A	48	B	62	B	66	A	48		
A	45	A	49	A	43	B	65	B	64		
A	45	A	46	A	40	A	46	A	48		

Modelo:



μ_A=50, σ_A =5, p_A=0.6 μ_B=65, σ_B =2, p_B=0.4

Dado $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ de N observações *i.i.d* temos:

$$p(\mathbf{X}) = p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) = \prod_{j=1}^{N} p(\mathbf{x}_j) = \prod_{j=1}^{N} \sum_{l=1}^{k} \pi_l \mathcal{N}(\mathbf{x}_j | \mathbf{v}_l, \boldsymbol{\Sigma}_l)$$

Por conveniência matemática, utiliza-se da **log-verossimilhança**:

$$\ln(p(\mathbf{X} | \boldsymbol{\pi}, \boldsymbol{\Sigma}, \mathbf{v})) = \sum_{j=1}^{N} \ln\left(\sum_{l=1}^{k} \pi_l \mathcal{N}(\mathbf{x}_j | \mathbf{v}_l, \boldsymbol{\Sigma}_l) \right)$$

- Maximizar a verossimilhança pode ser visto como maximizar a compatibilidade entre as N observações e o modelo

- EM (Dempster et al., 1977) é um algoritmo de otimização que visa maximizar a (log) verossimilhança em dois passos:

- **Passo E** (*Expectation*)
  - Avalia as probabilidades a posteriori μ_{ij} ($i = 1, ..., k; j = 1, ..., N$) a partir das N observações $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ e do modelo corrente, dado pelos parâmetros $\mathbf{\Sigma} = \{\mathbf{\Sigma}_1, ..., \mathbf{\Sigma}_k\}$, $\mathbf{v} = \{\mathbf{v}_1, ..., \mathbf{v}_k\}$ e $\boldsymbol{\pi} = \{\pi_1, ..., \pi_k\}$.
- **Passo M** (*Maximization*)
  - Ajusta os parâmetros do modelo visando maximizar a log-verossimilhança.

Passos E e M

E: computar μ_{ij} $(i = 1, ..., k; j = 1, ..., N)$

$$\mu_{ij} = \frac{\pi_i \mathcal{N}\left(\mathbf{x}_j \middle| \mathbf{v}_i, \boldsymbol{\Sigma}_i\right)}{\sum_{l=1}^{k} \pi_l \mathcal{N}\left(\mathbf{x}_j \middle| \mathbf{v}_l, \boldsymbol{\Sigma}_l\right)}$$

$$\mathcal{N}\left(\mathbf{x}_j \middle| \mathbf{v}_i, \boldsymbol{\Sigma}_i\right) = \frac{1}{(2\pi)^{n/2} \det\left(\boldsymbol{\Sigma}_i\right)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}_j - \mathbf{v}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_j - \mathbf{v}_i)\right\}$$

M: computar

$$\mathbf{v}_i = \frac{1}{N_i} \sum_{j=1}^{N} \mu_{ij} \mathbf{x}_j$$ ➡ centróide ponderado

$$\boldsymbol{\Sigma}_i = \frac{1}{N_i} \sum_{j=1}^{N} \mu_{ij} (\mathbf{x}_j - \mathbf{v}_i)(\mathbf{x}_j - \mathbf{v}_i)^T$$ ➡ covariância ponderada

$$\pi_i = \frac{N_i}{N}$$ ➡ Coeficientes = prob. a priori do i-ésimo componente

$$N_i = \sum_{j=1}^{N} \mu_{ij}$$ ➡ Nº efetivo de pontos atribuídos ao i-ésimo grupo

Algoritmo EM

1. **Inicialização** (via *k*-means)
   - protótipos $\mathbf{v}_i$ = centróides finais do *k*-means
   - covariâncias $\sum_i$ = matrizes de covariância dos grupos
   - probabilidades μ_{ij} (para N_i e π_i) = matriz de partição final
2. **Passo E**
3. **Passo M**
4. **Avaliação do Critério de Parada** (função de log-verossimilhança)
5. **Interrupção ou Retorno ao Passo 2**

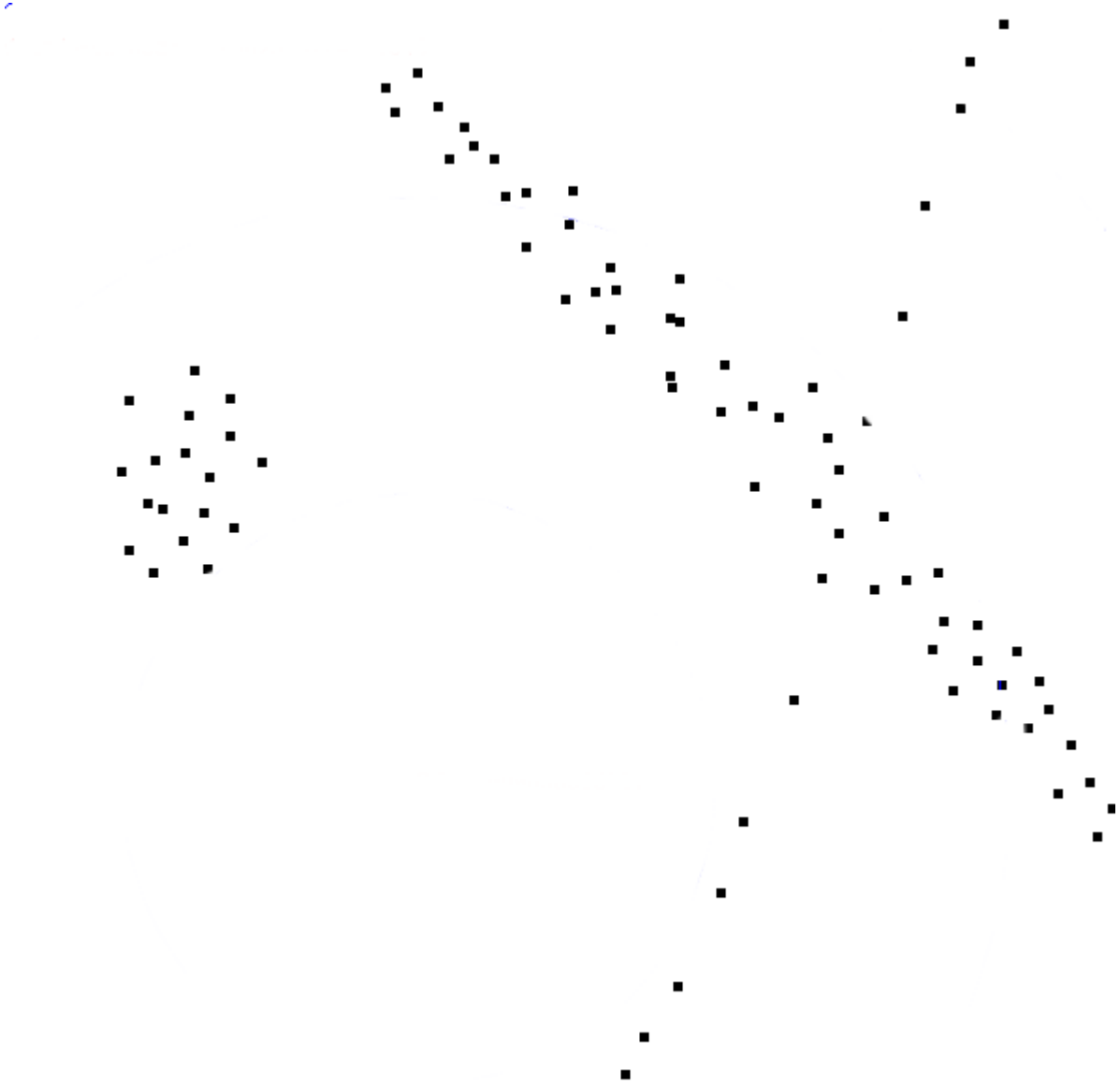
EM x *k*-means

- EM produz informação muito mais rica sobre os dados (Probabilidades associadas a cada objeto / cluster);
- Probabilidades produzidas por EM podem facilmente ser convertidas em uma partição rígida;

➢ Essa partição é capaz de representar clusters alongados, elipsoidais, com atributos correlacionados;

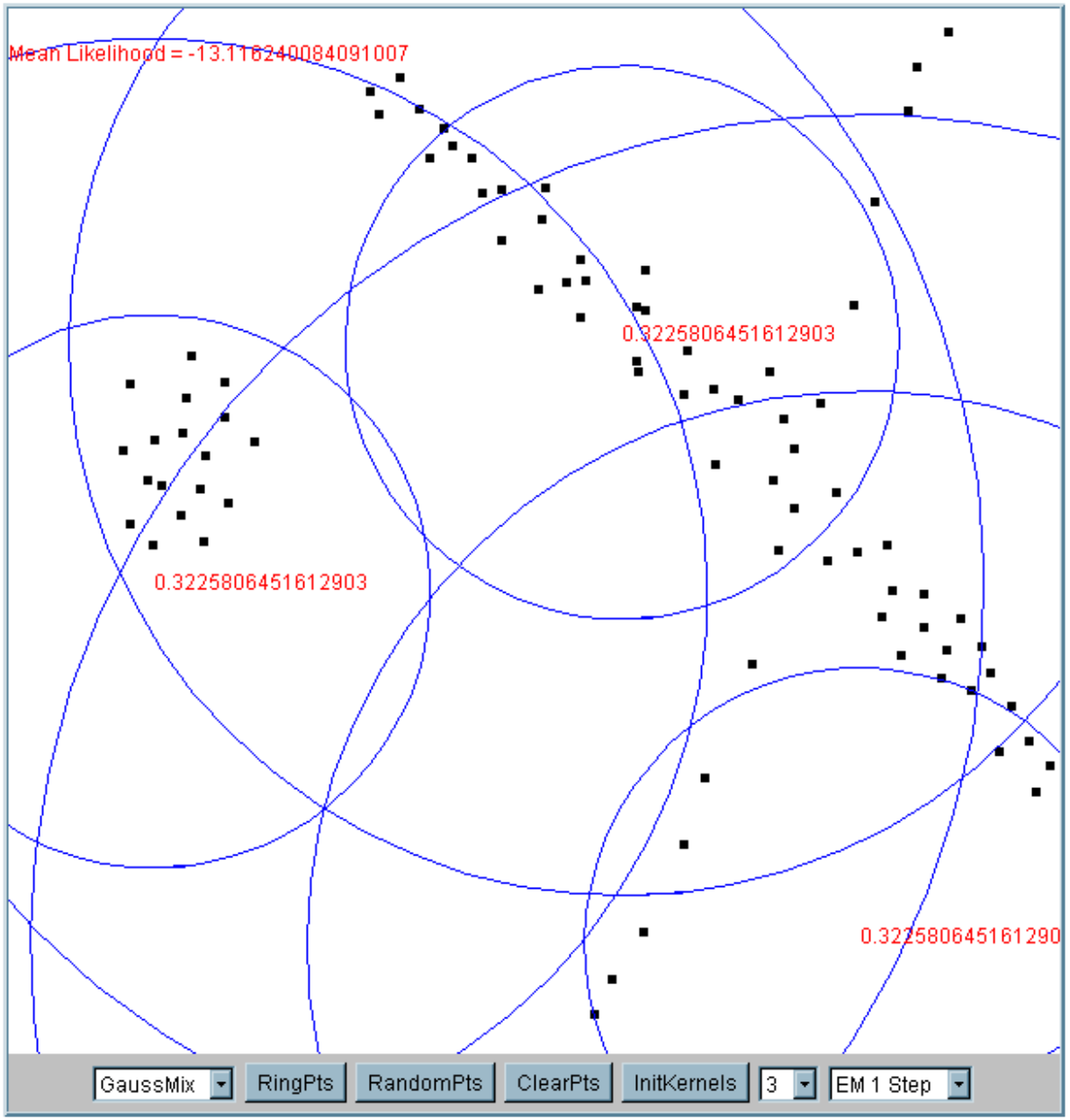
- No entanto, todas as vantagens acima vêm com um elevado custo computacional associado:
  - Cálculo das Normais Multi-Dimensionais demanda as inversas das matrizes de covariância $\sum_i$ - $O(n^3)$;
  - *k*-means é um caso particular de EM. Ambos estão sujeitos a mínimos locais.

Rodando o EM (exemplo)

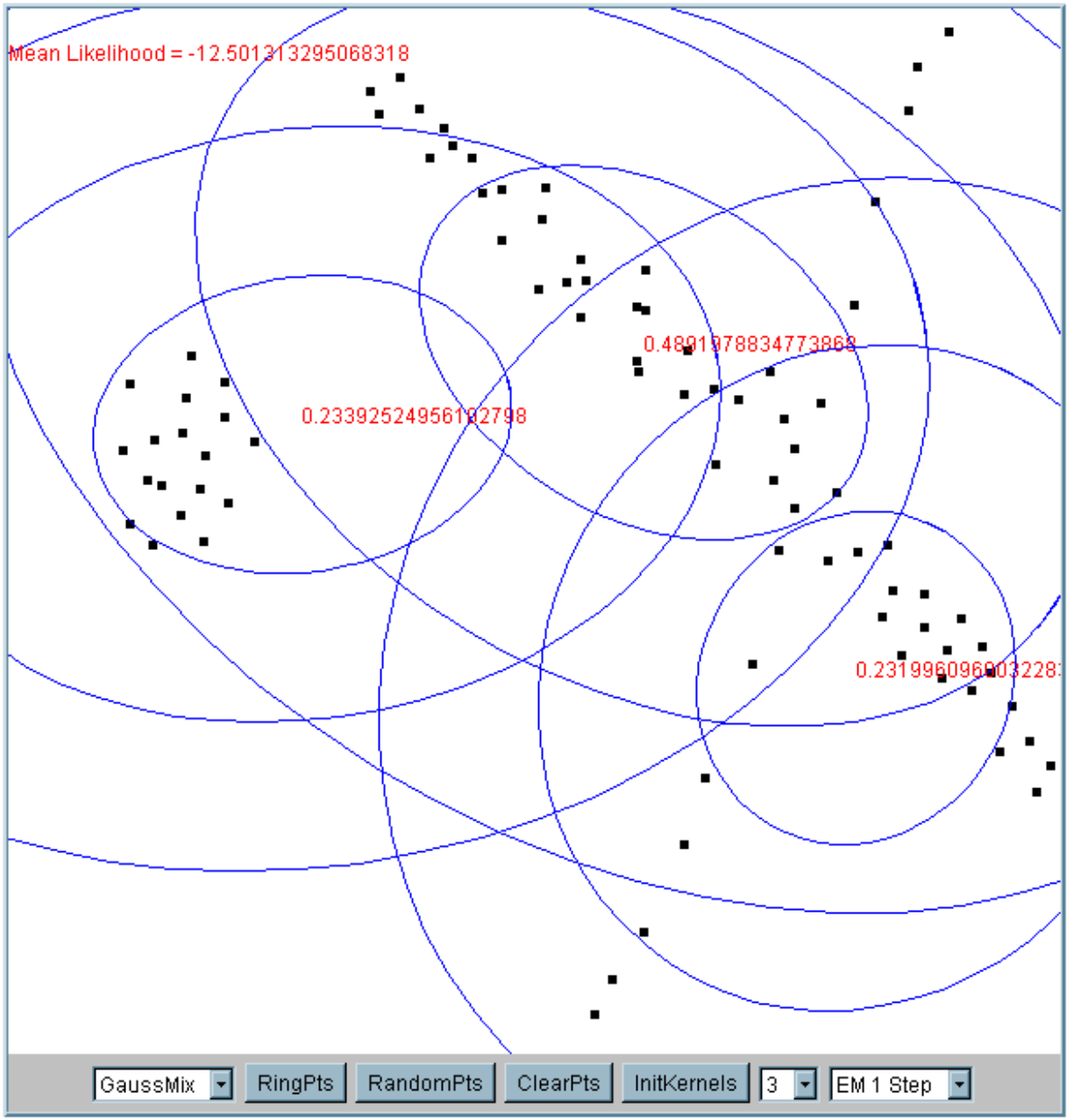


Fonte do exemplo: Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Manaus.

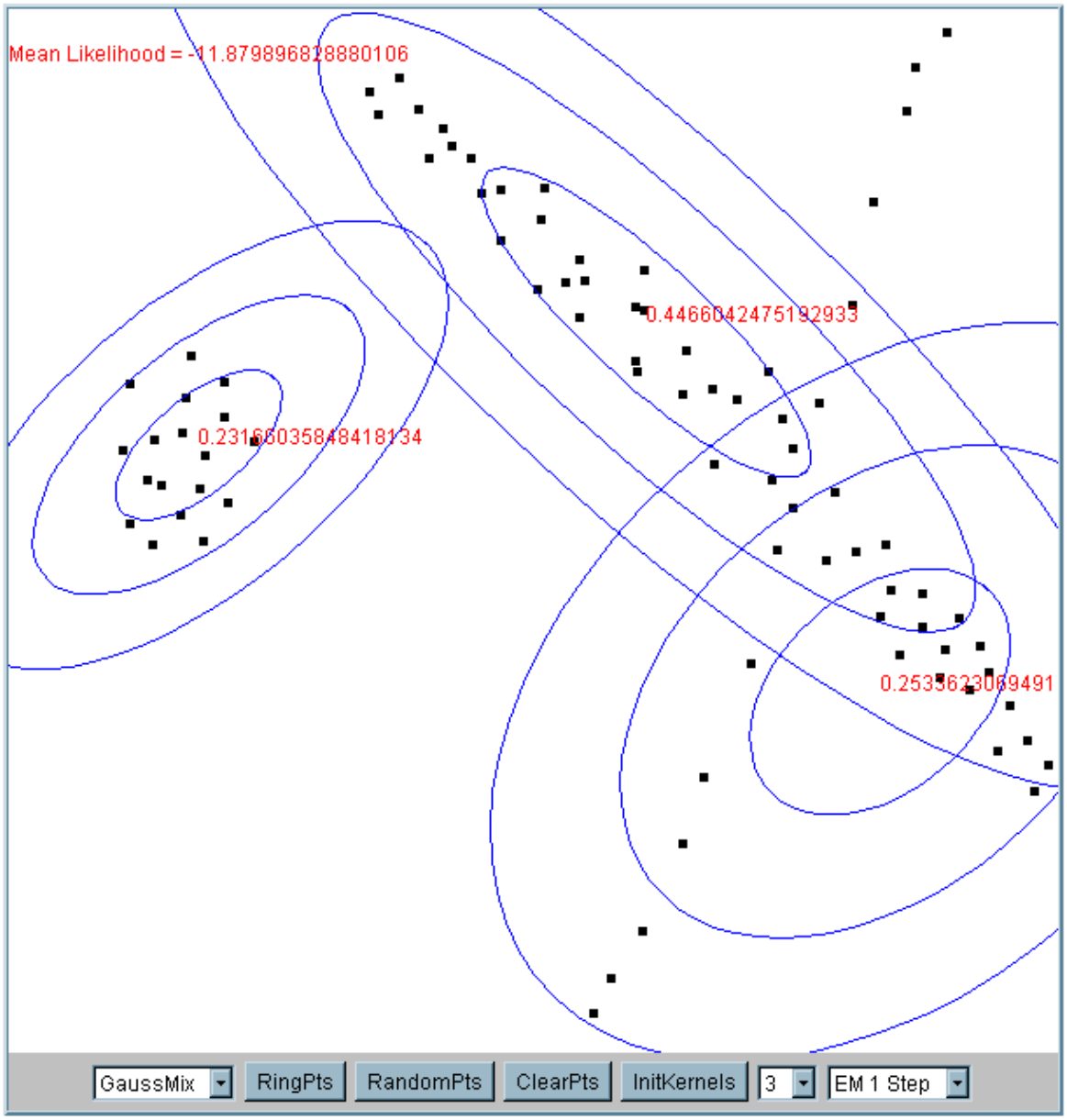
Iteração 1


Mean Likelihood = -13.116240084091007
0.3225806451612903
0.3225806451612903
0.322580645161290
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM 1 Step

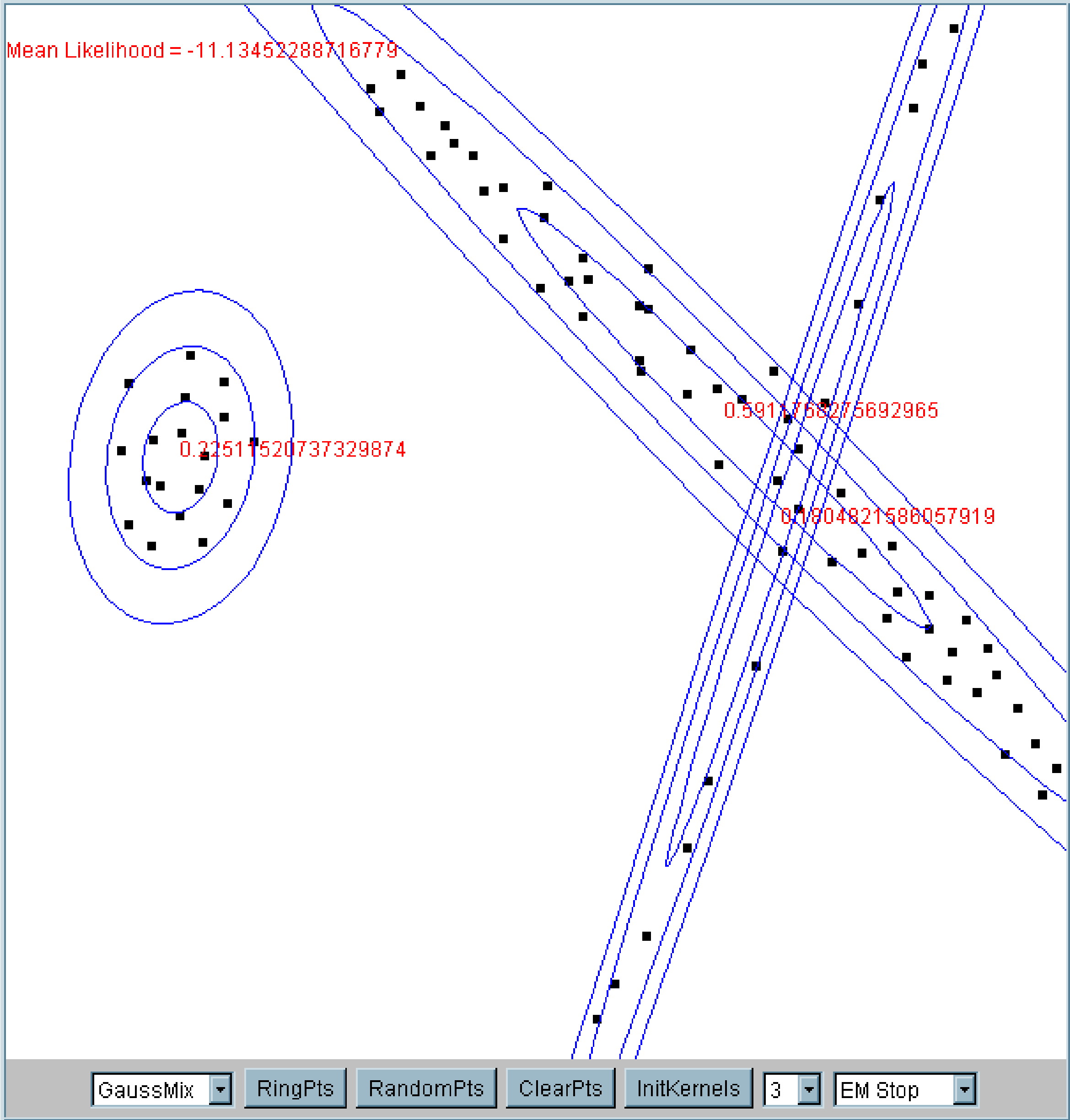
Iteração 2


Mean Likelihood = -12.501313295068318
0.23392524956102798
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM 1 Step

Iteração 5


Mean Likelihood = -11.879896828880106
0.4466042475192933
0.2316603584841813 4
0.2533623069491
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM 1 Step

Iteração 25



Mean Likelihood = -11.13452288716779
0.22511520737329874
0.5911758275692965
0.18048215860 57919
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM Stop

Exercício

Objeto	*x*
1	-1.31
2	-0.43
3	0.34
4	3.57
5	2.76
6	0.30
7	9.06
8	4.45
9	2.87
10	4.42

Execute manualmente iterações do EM na base de dados ao lado (n = 1, N = 10), com k = 2. Tome protótipos iniciais arbitrários e os demais parâmetros inicializados a partir destes, de maneira análoga à inicialização via k-means.

Ilustre o resultado obtido de forma gráfica

Agenda

- Motivação e conceitos
- Definições preliminares
- k-means
- Estimando o número de clusters a partir dos dados
- Bisecting k-means
- k-medoids
- EM para misturas de Gaussianas
- Avaliação de agrupamentos

Validação

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

(Jain and Dubes, *Algorithms for Clustering Data*, 1988)

- **Validação** é um termo que se refere de forma ampla aos diferentes procedimentos para avaliar de maneira objetiva e quantitativa os resultados de análise de agrupamento.

- Cada um desses procedimentos pode nos ajudar a responder uma ou mais questões do tipo:

  - Encontramos grupos de fato?

    - grupos são pouco usuais ou facilmente encontrados ao acaso?

  - Qual a qualidade (relativa ou absoluta) dos grupos encontrados?

  - Qual é o número natural / mais apropriado de grupos?

- A maneira quantitativa para validação é alcançada através de algum tipo de **índice**. Há 3 tipos de **índices/critérios de validade**:

- **Externos**: Avalia o grau de correspondência entre a estrutura de grupos (partição ou hierarquia) sob avaliação e informação a priori na forma de uma solução de agrupamento esperada ou conhecida.

- **Internos**: Avalia o grau de compatibilidade entre a estrutura de grupos sob avaliação e os dados, usando apenas os próprios dados.

- **Relativos**: Avaliam qual dentre duas ou mais estruturas de grupos é melhor sob algum aspecto. Tipicamente são critérios internos capazes de quantificar a qualidade relativa.

➢ Já vimos exemplos de índices internos (J) e relativos (silhuetas). Vejamos agora exemplos de índices externos...

- Embora o problema de *clustering* seja não supervisionado, em alguns cenários o resultado de agrupamento desejado pode ser conhecido. Por exemplo:
  - Reconhecimento visual dos clusters naturais (bases 2D, 3D)
  - Especialista de domínio
  - Bases geradas sinteticamente com distribuições conhecidas
    - *Benchmark data sets*
  - Bases de classificação sob a hipótese de que classes são grupos
- Índices externos medem o nível de compatibilidade entre uma partição obtida e uma partição de referência dos mesmos dados

- Estudaremos os índices mais usados (Rand e Jaccard). Adotaremos a seguinte terminologia:
  - grupos da **partição de referência** (*golden truth*) → "**classes**"
  - grupos da **partição sob avaliação → clusters (grupos)**
- Podemos então definir as grandezas de interesse:

a: No. de pares da mesma classe e do mesmo cluster

b: No. de pares da mesma classe e de clusters distintos

c: No. de pares de classes distintas e do mesmo cluster

d: No. de pares de classes e clusters distintos

$$RI = \frac{a+d}{a+b+c+d}$$

Número de pares de objetos:

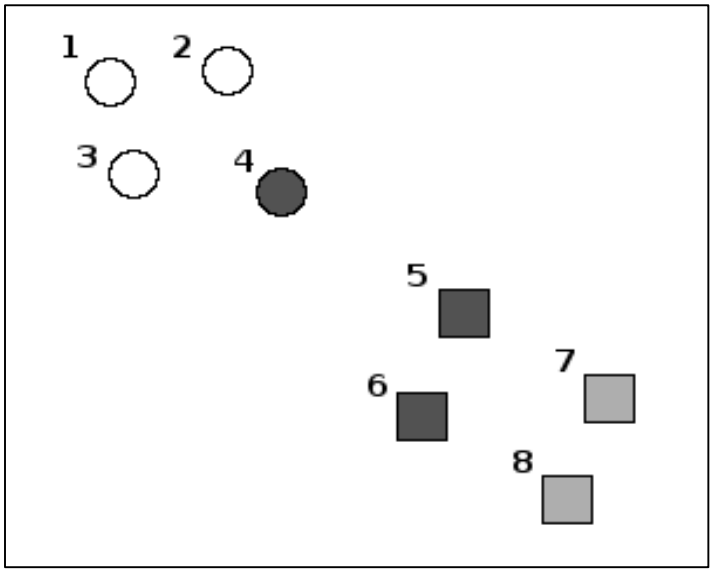
a: da mesma classe e do mesmo cluster (grupo)

b: da mesma classe e de clusters distintos

c: de classes distintas e do mesmo cluster

d: de classes distintas e de clusters distintos

Figura por Lucas Vendramin



2 Classes (Círculos e Quadrados)
3 Clusters (Preto, Branco e Cinza)

a = 5; b = 7; c = 2; d = 14

RI = 5+14/(5+7+2+14) = 0.6785

Limitações do *Rand Index*

- **Viés** de favorecer a comparação de partições com níveis mais elevados de granularidade, i.e., apresenta valores mais elevados ao comparar partições com mais grupos.
- Por quê?
  - mesmo peso para objetos agregados (termo **a**) ou separados (**d**)
  - termo **d** tende a dominar o índice
  - quanto mais grupos, mais pares pertencem a grupos distintos
    - isso é válido em qualquer uma das duas partições
    - probabilidade / incidência de pares em comum é maior

Índice de Jaccard

Elimina o termo **d** sob a ótica de que um agrupamento é uma coleção de agregações de pares de objetos (separações sendo apenas uma consequência):

$$Jc = \frac{a}{a+b+c}$$

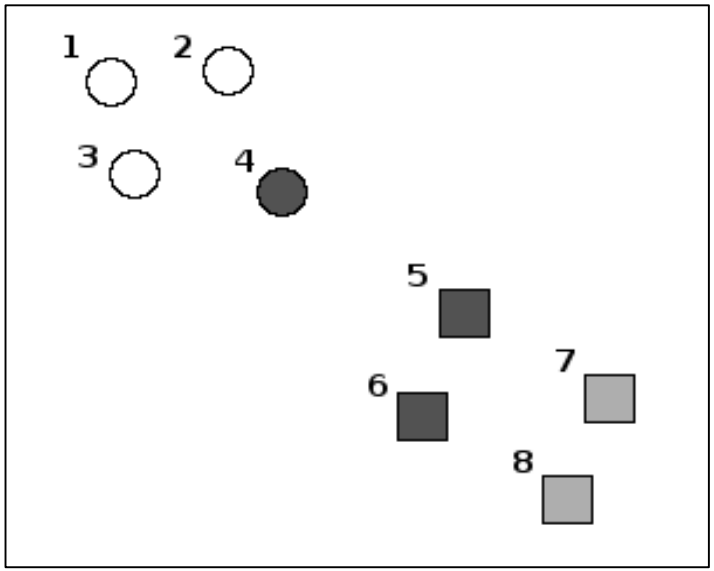
Número de pares de objetos:

a: da mesma classe e do mesmo cluster

b: da mesma classe e de clusters distintos

c: de classes distintas e do mesmo cluster

Figura por Lucas Vendramin



2 Classes (Círculos e Quadrados)
3 Clusters (Preto, Branco e Cinza)

a = 5; b = 7; c = 2

Jc = 5/(5+7+2) = 0.3571

Referências Bibliográficas

- Jain, A. K. and Dubes, R. C., Algorithms for Clustering Data, Prentice Hall, 1988.
- Kaufman, L., Rousseeuw, P. J., Finding Groups in Data – An Introduction to Cluster Analysis, Wiley, 2005.
- Tan, P.-N., Steinbach, M., and Kumar, V., *Introduction to Data Mining*, Addison-Wesley, 2006.
- Wu, X. and Kumar, V., *The Top Ten Algorithms in Data Mining*, Chapman & Hall/CRC, 2009.
- D. Steinley, *K-Means Clustering: A Half-Century Synthesis*, British J. of Mathematical and Stat. Psychology, V. 59, 2006.

Agenda

- Motivação e conceitos
- Definições preliminares
- k-means
- Estimando o número de clusters a partir dos dados
- Bisecting k-means
- k-medoids
- EM para misturas de Gaussianas
- Avaliação de agrupamentos