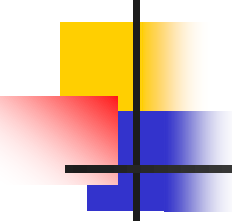


# Agrupamento de Dados e Aplicações

---

## Algoritmos Particionais Parte I

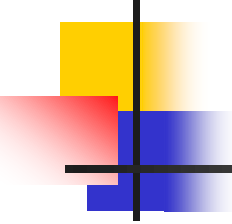
**Prof. Eduardo R. Hruschka**



# Créditos

---

- Este material consiste de adaptações e extensões dos originais:
  - Elaborados por Eduardo Hruschka e Ricardo Campello
  - de (Tan et al., 2006)
  - de E. Keogh (SBBD 2003)
  - de G. Piatetsky-Shapiro (KDNuggets)



# Aula de Hoje

---

- Definições
- Algoritmos Particionais sem Sobreposição
  - k-Means
    - Algoritmo Básico
    - Inicialização e Implementações Eficientes
    - Fundamentação Teórica (perspectiva de otimização)
  - K-Means Paralelo e Distribuído
  - Variantes do k-Means
    - K-Mediana, k-Medóides, ...
  - Estimativa do Número de Grupos (conceitos preliminares)

# Definição de Partição de Dados (Revisão)

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

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

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

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

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

# Matriz de Partição

- **Matriz de Partição** possui  $k$  linhas (no. de grupos) e  $N$  colunas (no. de objetos) na qual cada elemento  $\mu_{ij}$  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  $\sum_i(\mu_{ij}) = 1 \quad \forall j$  for respeitada, então denomina-se de *matriz de partição **rígida, exclusiva** ou **sem sobreposição***

# Matriz de Partição

- **Exemplo:**

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

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

# Métodos Particionais (Sem Sobreposição)

- Métodos *particionais* sem sobreposição referem-se a algoritmos de agrupamento que buscam por uma matriz de partição rígida de um conjunto de objetos  $\mathbf{X}$ .

**Encontrar uma Matriz de Partição  $\mathbf{U}(\mathbf{X})$ :** Equivale a particionar o conjunto  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  de  $N$  objetos em uma coleção  $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_k\}$  de  $k$  grupos disjuntos  $\mathbf{C}_i$  tal que  $\mathbf{C}_1 \cup \mathbf{C}_2 \cup \dots \cup \mathbf{C}_k = \mathbf{X}$ ,  $\mathbf{C}_i \neq \emptyset$ , e  $\mathbf{C}_i \cap \mathbf{C}_j = \emptyset$  para  $i \neq j$

# Particionamento como Problema Combinatório

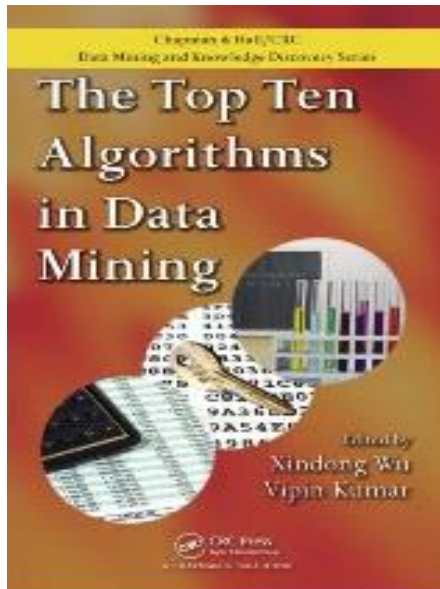
- **Problema:** Assumindo que  $k$  seja conhecido, o número de possíveis formas de agrupar  $N$  objetos em  $k$  *clusters* é dado por (Liu, 1968):

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

- Por exemplo,  $NM(100, 5) \approx 56.6 \times 10^{67}$ .
  - Em um computador com capacidade de avaliar  $10^9$  partições/s, precisaríamos  $\approx 1.8 \times 10^{50}$  séculos para processar todas as avaliações
- Como  $k$  em geral é desconhecido, problema é ainda maior
  - **NP-Hard:** Avaliação computacional exaustiva é impraticável
- **Solução:** formulações alternativas...

# Algoritmo k-Means

- ❑ Começaremos nosso estudo com um algoritmo clássico
  - ❑ listado entre os **Top 10 Most Influential Algorithms in DM**



- 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 Info. Systems*, vol. 14, pp. 1-37, 2008

# Algoritmo k-Means

## ❑ Referência mais aceita como original:

J. B. MacQueen, *Some methods of classification and analysis of multivariate observations*, In Proceedings 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, California, USA, 1967, 281–297

## ❑ Porém...

"K-means has a rich and diverse history as it was independently discovered in different scientific fields by Steinhaus (1956), Lloyd (proposed in 1957, published in 1982), Ball & Hall (1965) and MacQueen (1967)" [Jain, *Data Clustering: 50 Years Beyond K-Means*, Patt. Rec. Lett., 2010]

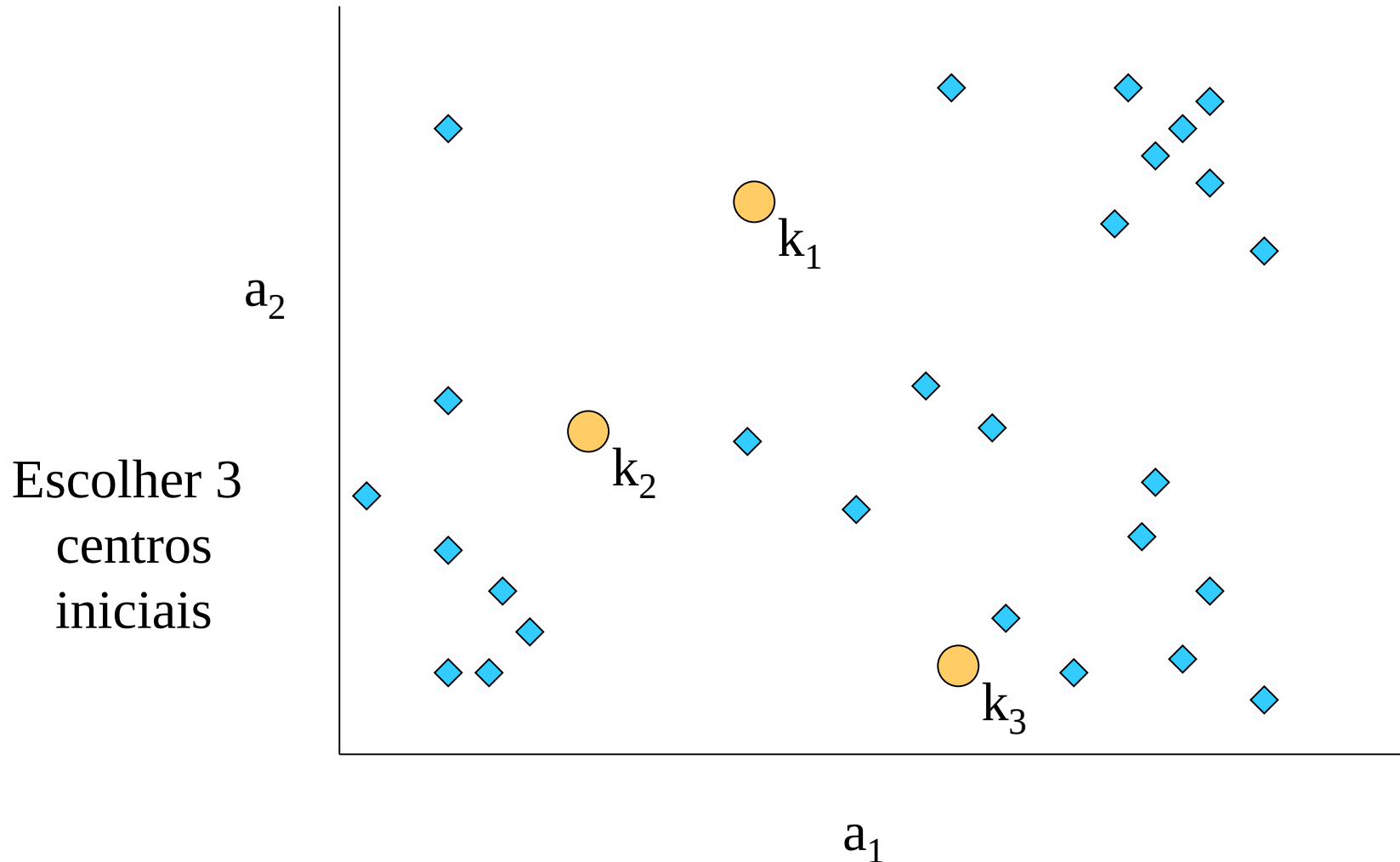
## ❑ ... e tem sido assunto por mais de meio século !

Douglas Steinley, *K-Means Clustering: A Half-Century Synthesis*, British Journal of Mathematical and Statistical Psychology, Vol. 59, 2006

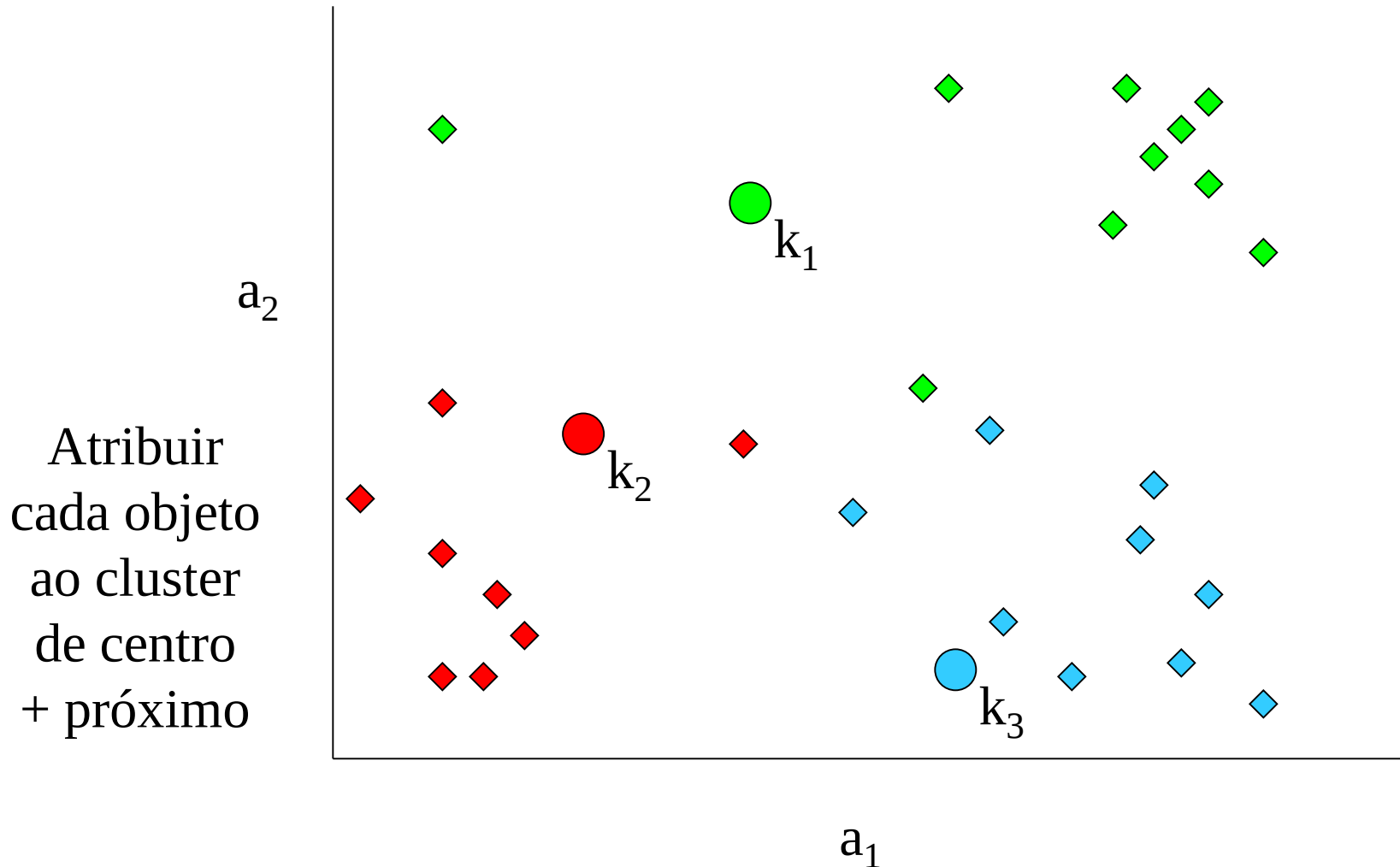
# k-Means

- 1) Escolher aleatoriamente  $k$  protótipos (centros) para os clusters
- 2) Atribuir cada objeto para o cluster de centro mais *próximo* (segundo alguma 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 mínimo de mudanças nos centróides

# k-Means - passo 1:

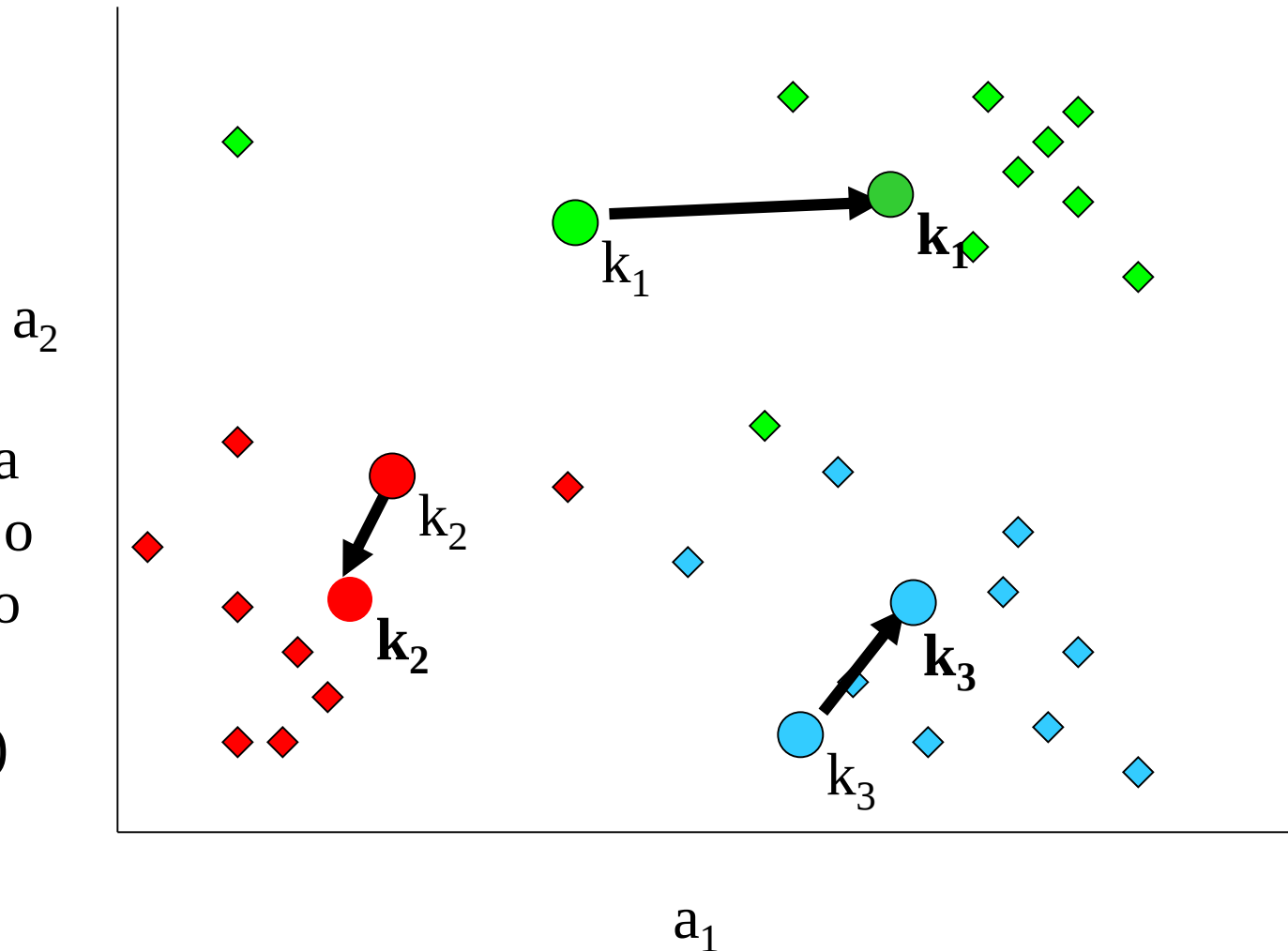


# k-Means - passo 2:



# k-Means - passo 3:

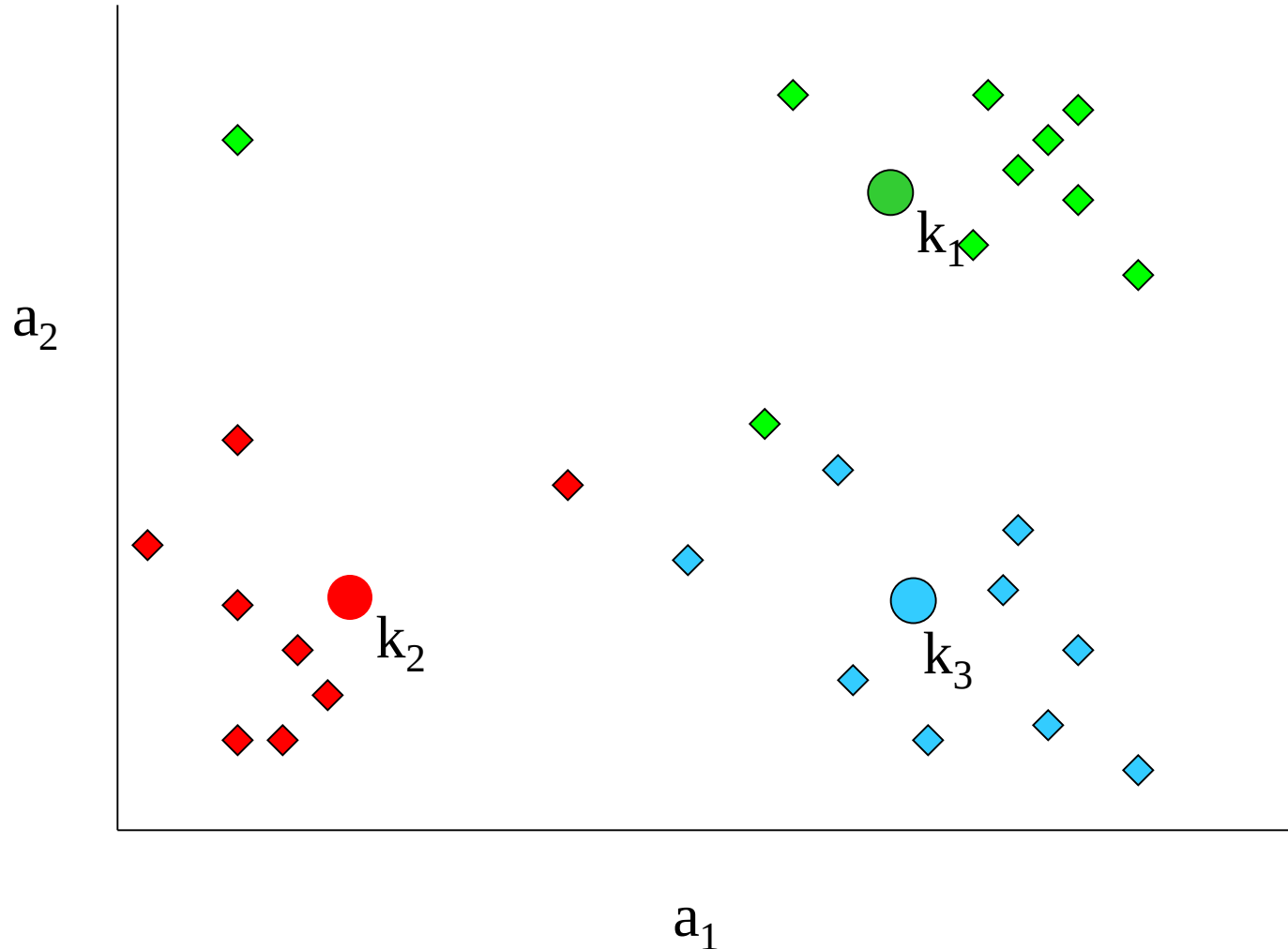
Mover cada  
centro para o  
vetor médio  
do cluster  
(centróide)



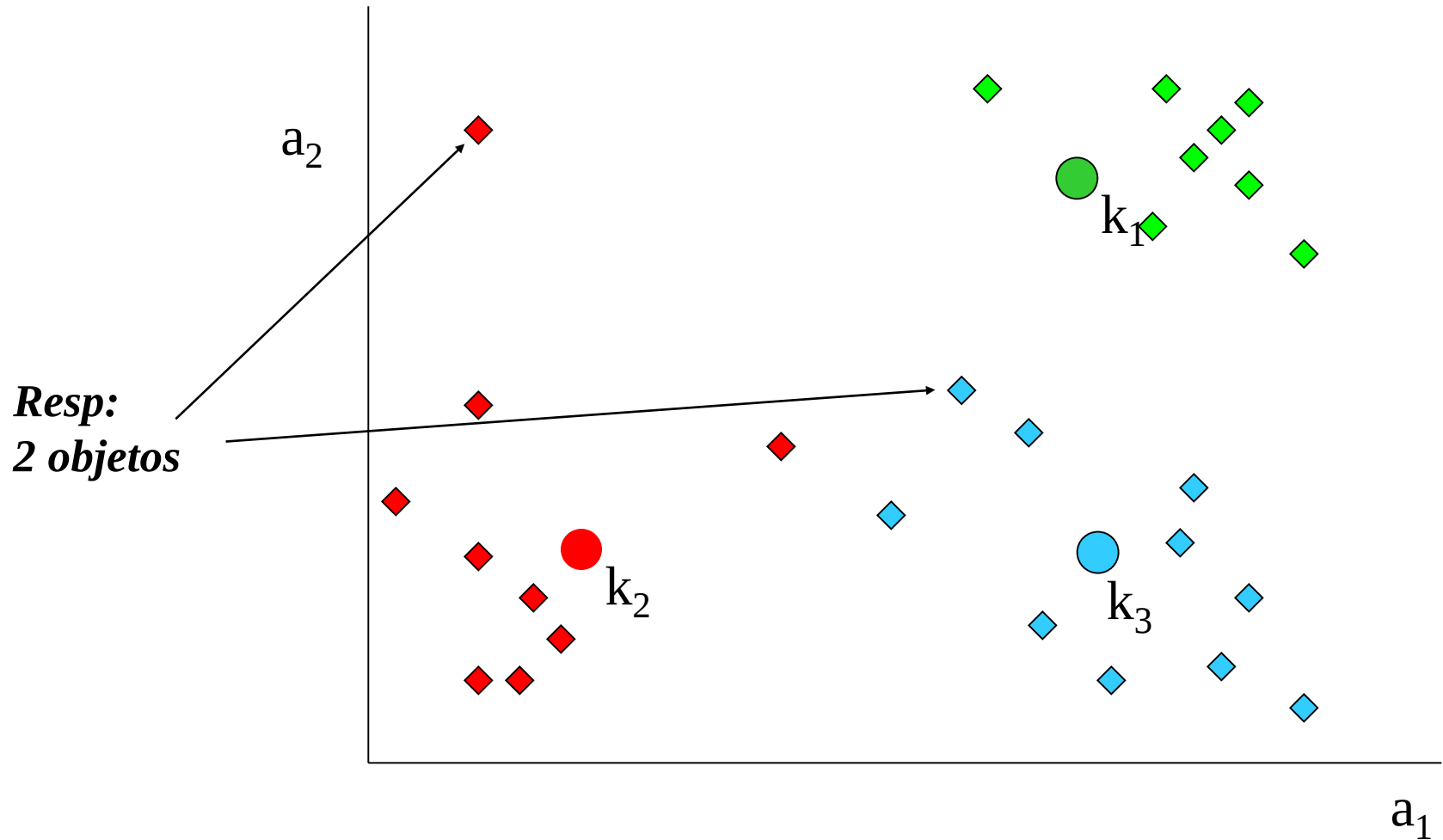
# k-Means:

Re-atribuir  
objetos aos  
clusters de  
centróides  
mais  
próximos

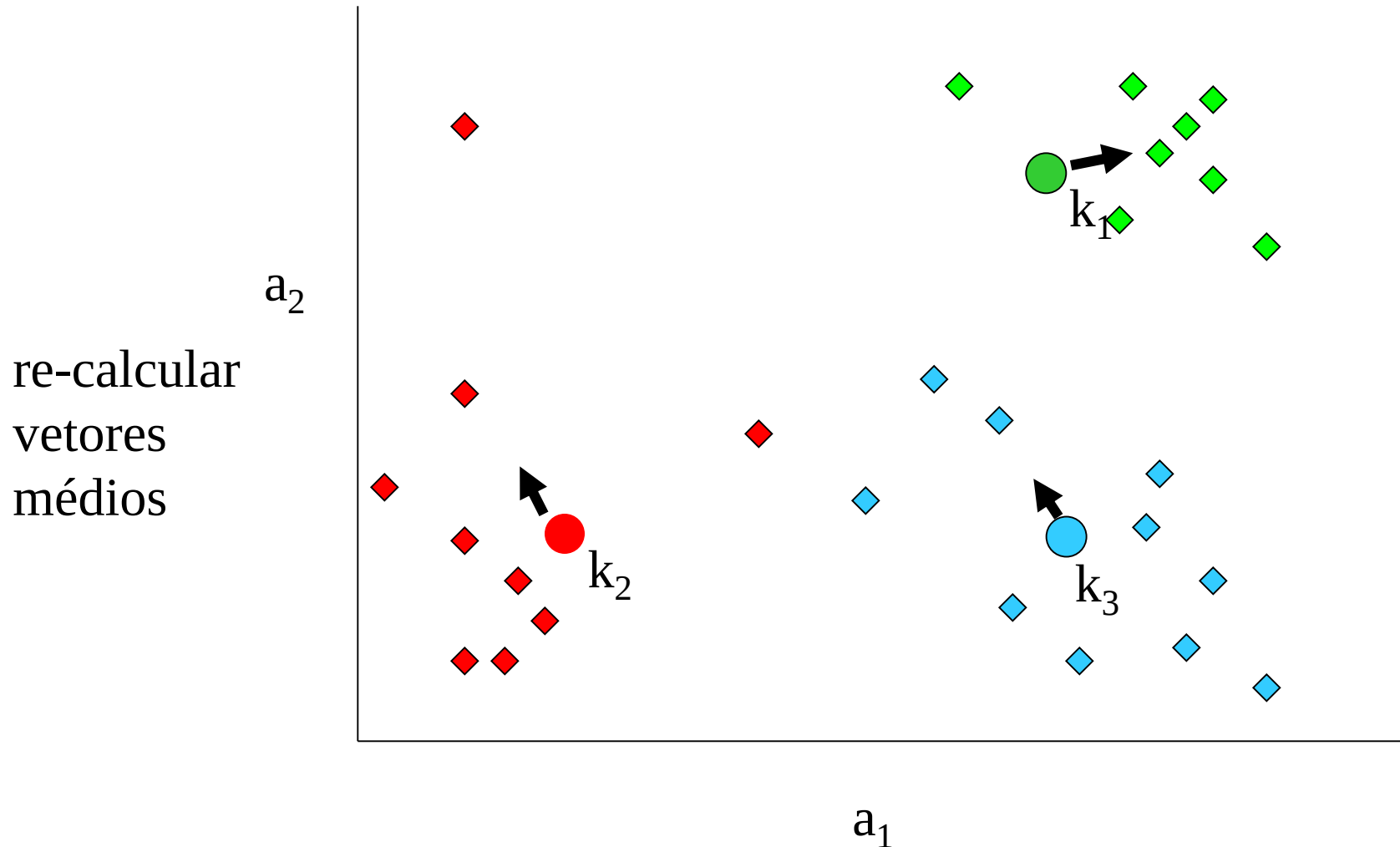
Quais objetos  
mudarão de  
cluster?



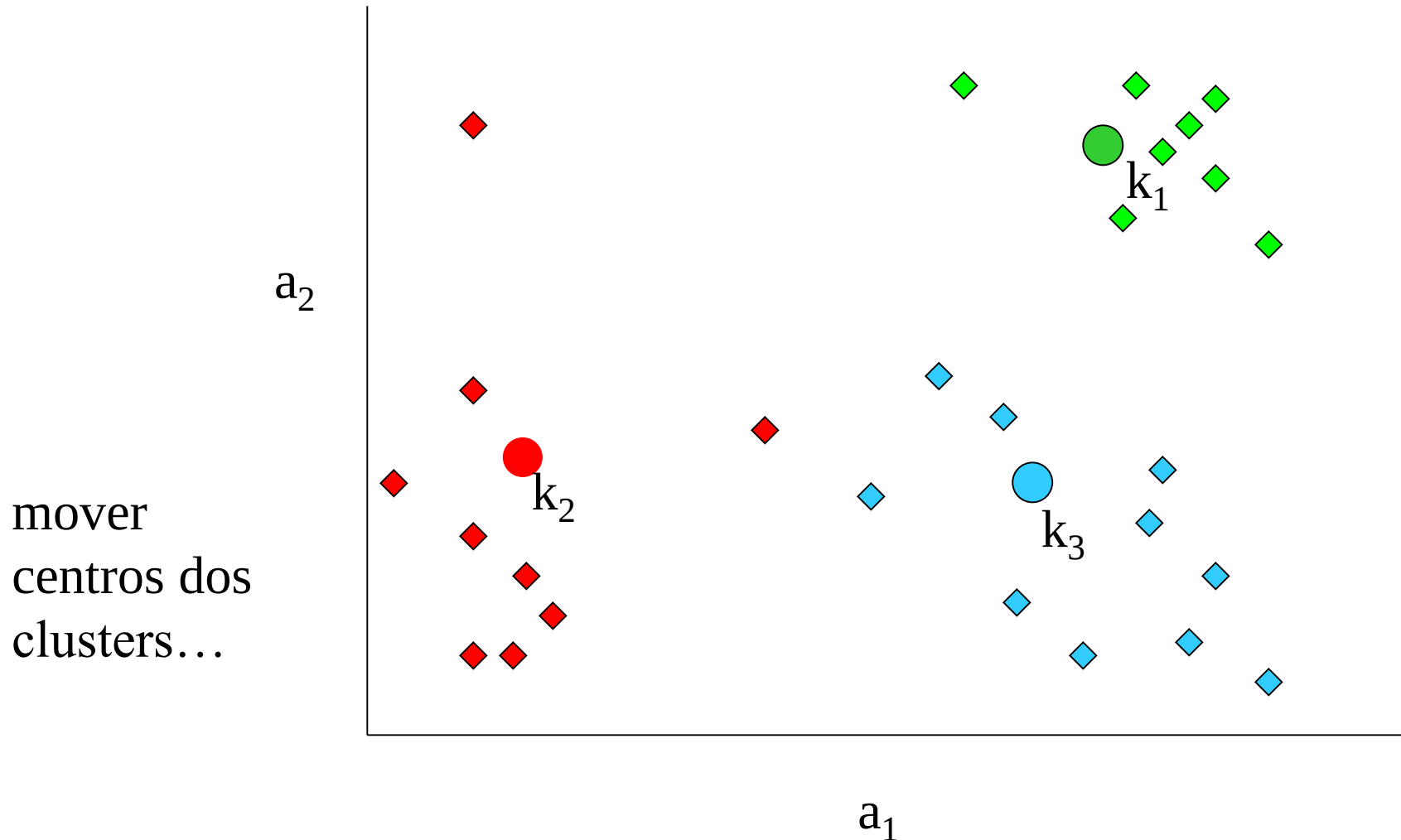
# k-Means:



# k-Means:

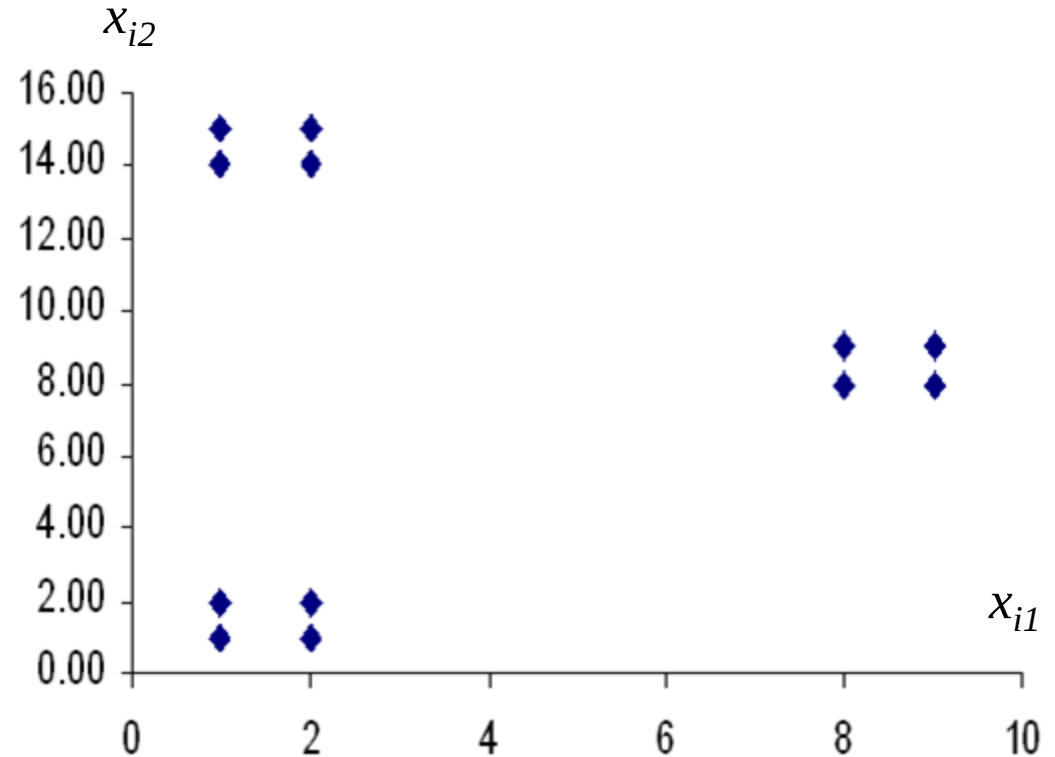


# k-Means:



# Exercício

| Objeto $x_i$ | $x_{i1}$ | $x_{i2}$ |
|--------------|----------|----------|
| 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  $[5 \ 10]$  e outros a sua escolha

# K-Means sob Perspectiva de Otimização

- Algoritmo minimiza a seguinte função objetivo:
  - **SSE** = *Sum of Squared Errors* (**variâncias intra-cluster**)

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

onde  $d$  = Euclidiana e  $\bar{\mathbf{x}}_c$  é o centróide do  $c$ -ésimo grupo:

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

# K-Means sob a Perspectiva de Otimização:

- Consideremos:
  - conjunto de objetos  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$
  - conjunto de  $k$  centróides quaisquer  $\{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \dots, \bar{\mathbf{x}}_k\}$
- Podemos reescrever o critério SSE de forma equivalente como:

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

- 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)**

# K-Means sob a Perspectiva de Otimização:

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

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
- $\mu_{cj} = 1$  para  $c$  que fornece o menor valor do erro quadrático

**\* Atribuir  $\mu_{cj} = 1$  para o grupo mais próximo.**

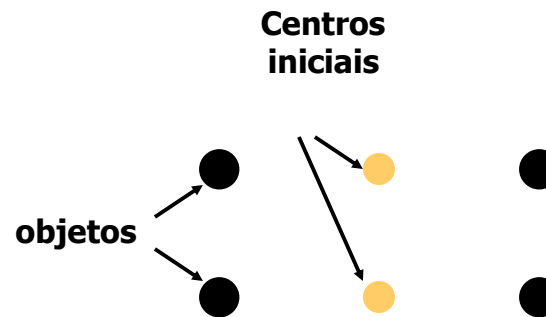
b) Minimizar  $J$  com respeito a  $\{\bar{\mathbf{x}}_c\}$ , fixando-se  $\{\mu_{cj}\}$  (**Passo M**)

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

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

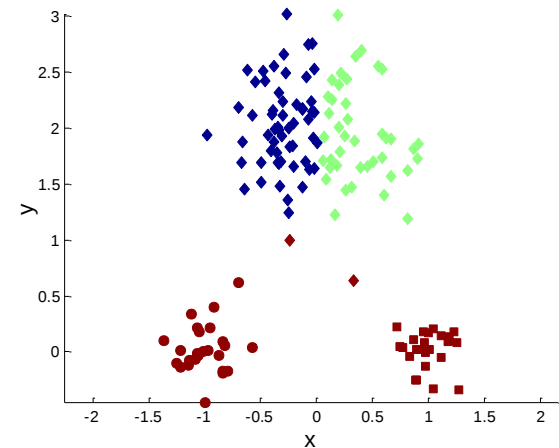
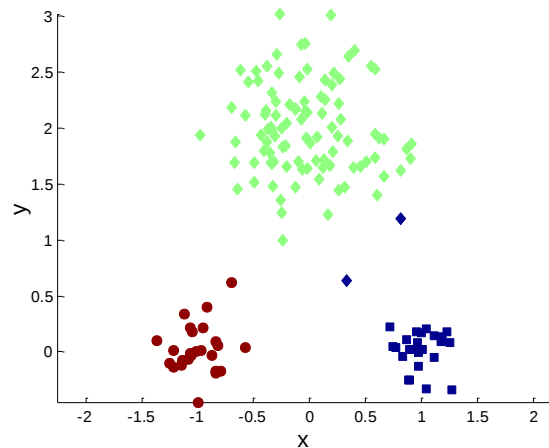
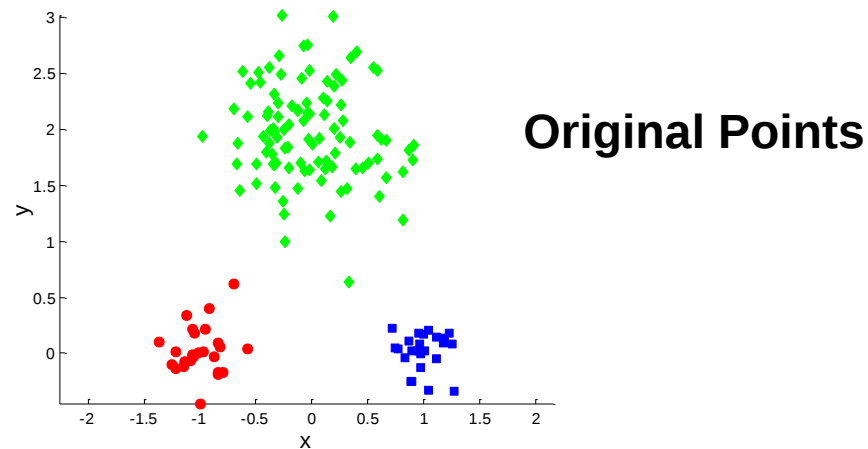
# Discussão

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

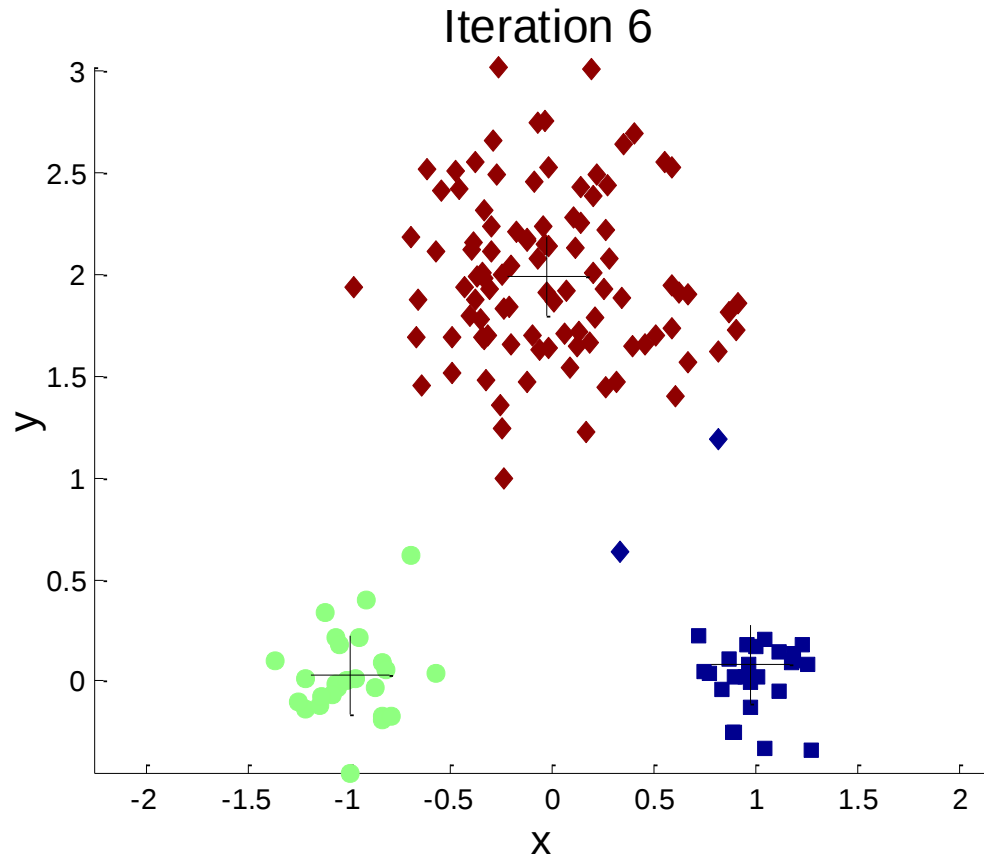


- Como evitar ?

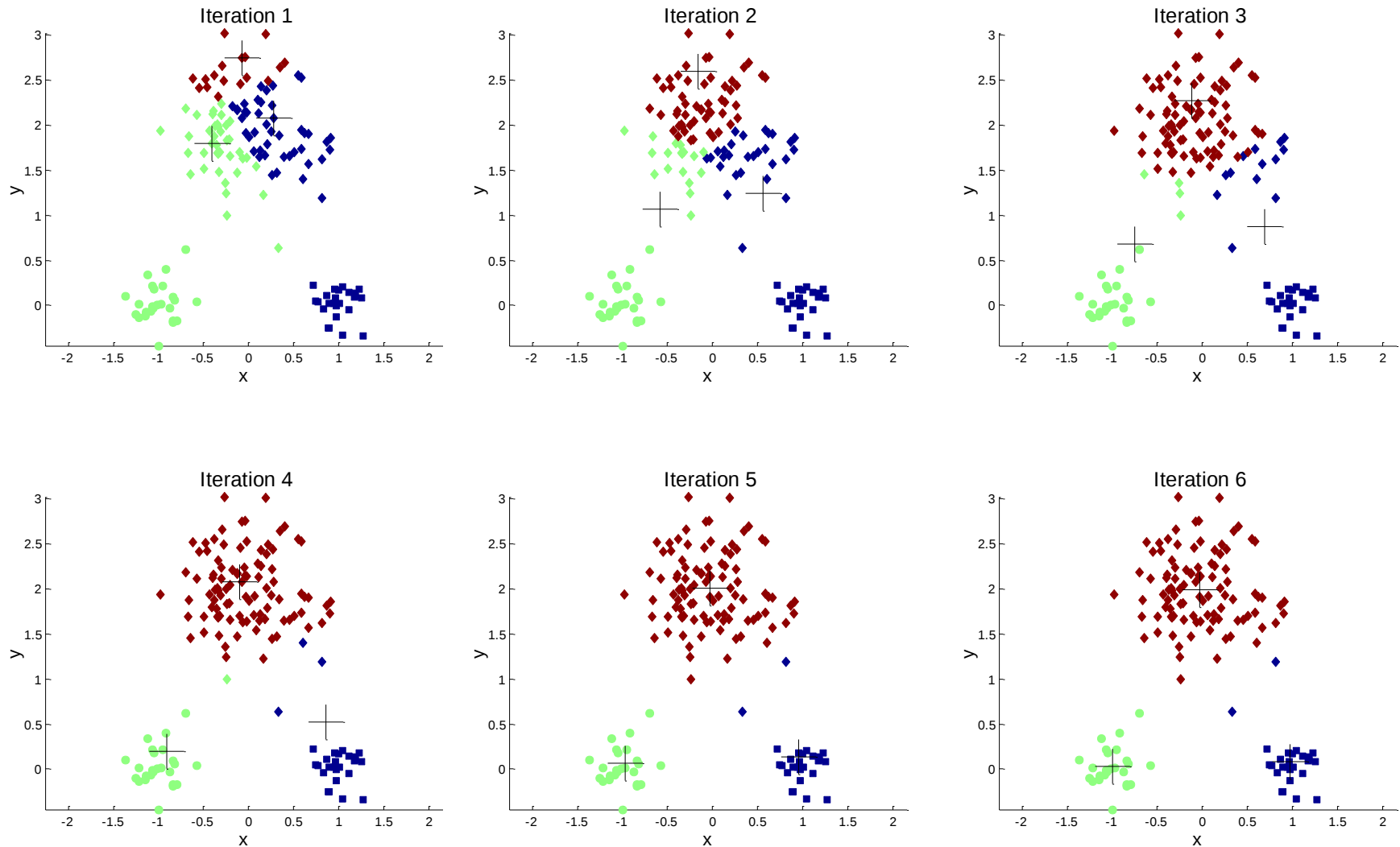
# Two different K-means Clusterings



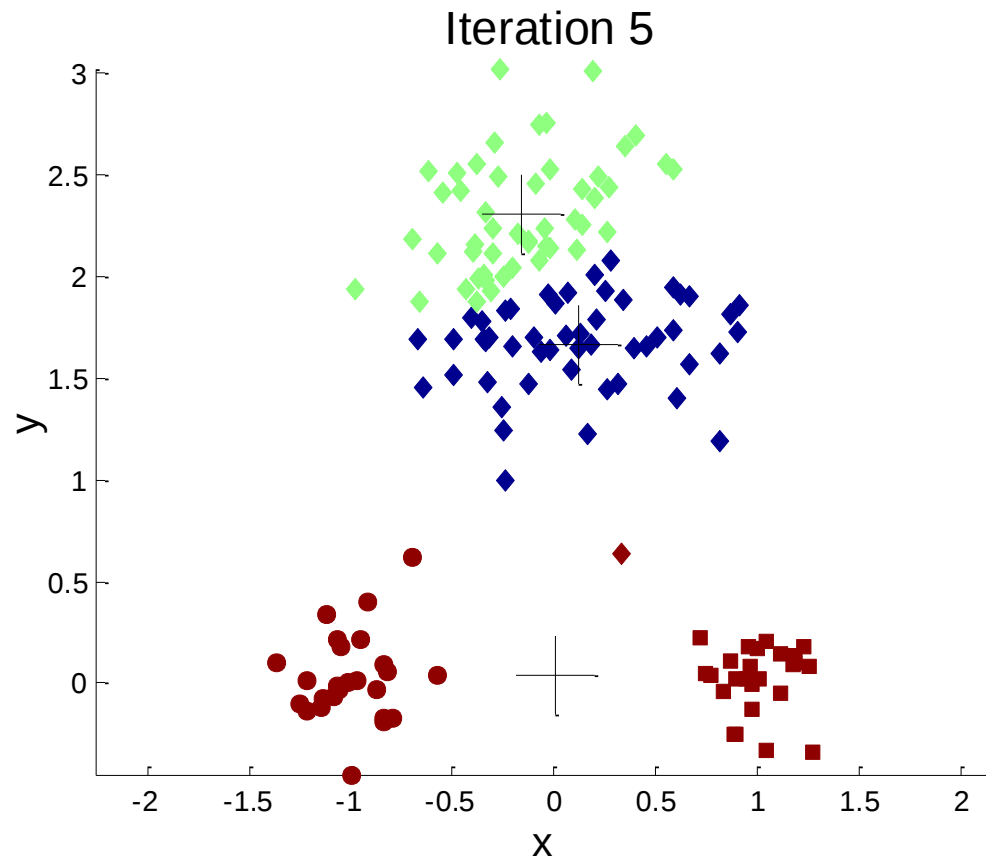
# Importance of Choosing Initial Centroids



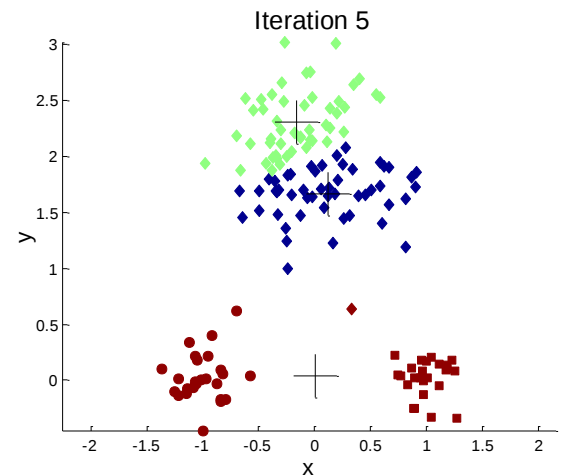
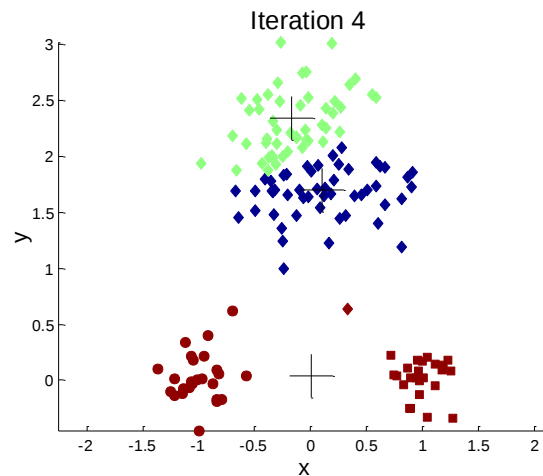
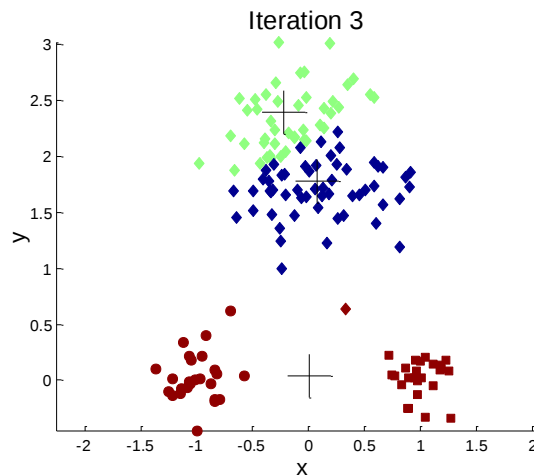
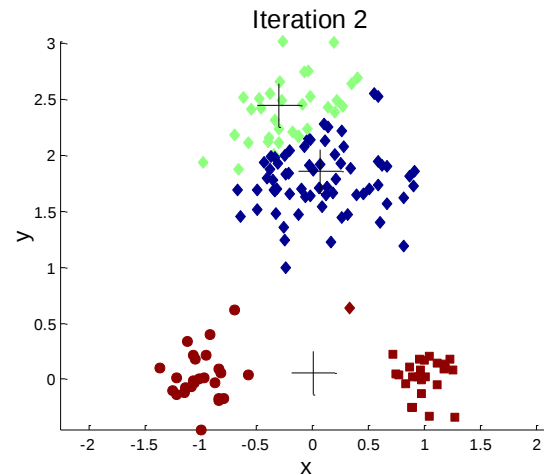
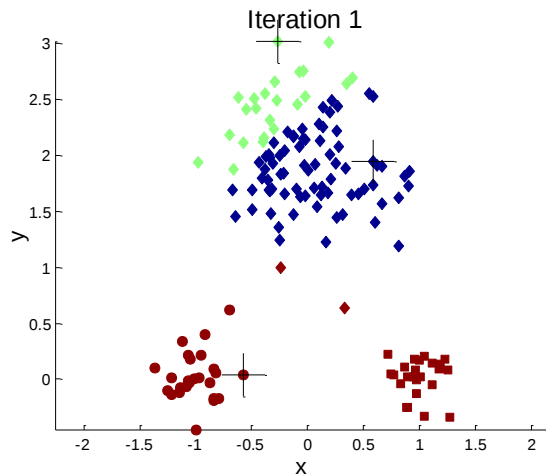
# Importance of Choosing Initial Centroids



# Importance of Choosing Initial Centroids ...



# Importance of Choosing Initial Centroids ...



# Análise da Seleção dos Protótipos Iniciais

- ❑ **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.
- ❑ Assumamos grupos balanceados, com uma mesma quantidade  $g = N / k$  de objetos cada:
  - Podemos calcular a probabilidade de selecionar 1 protótipo de cada grupo diferente como:

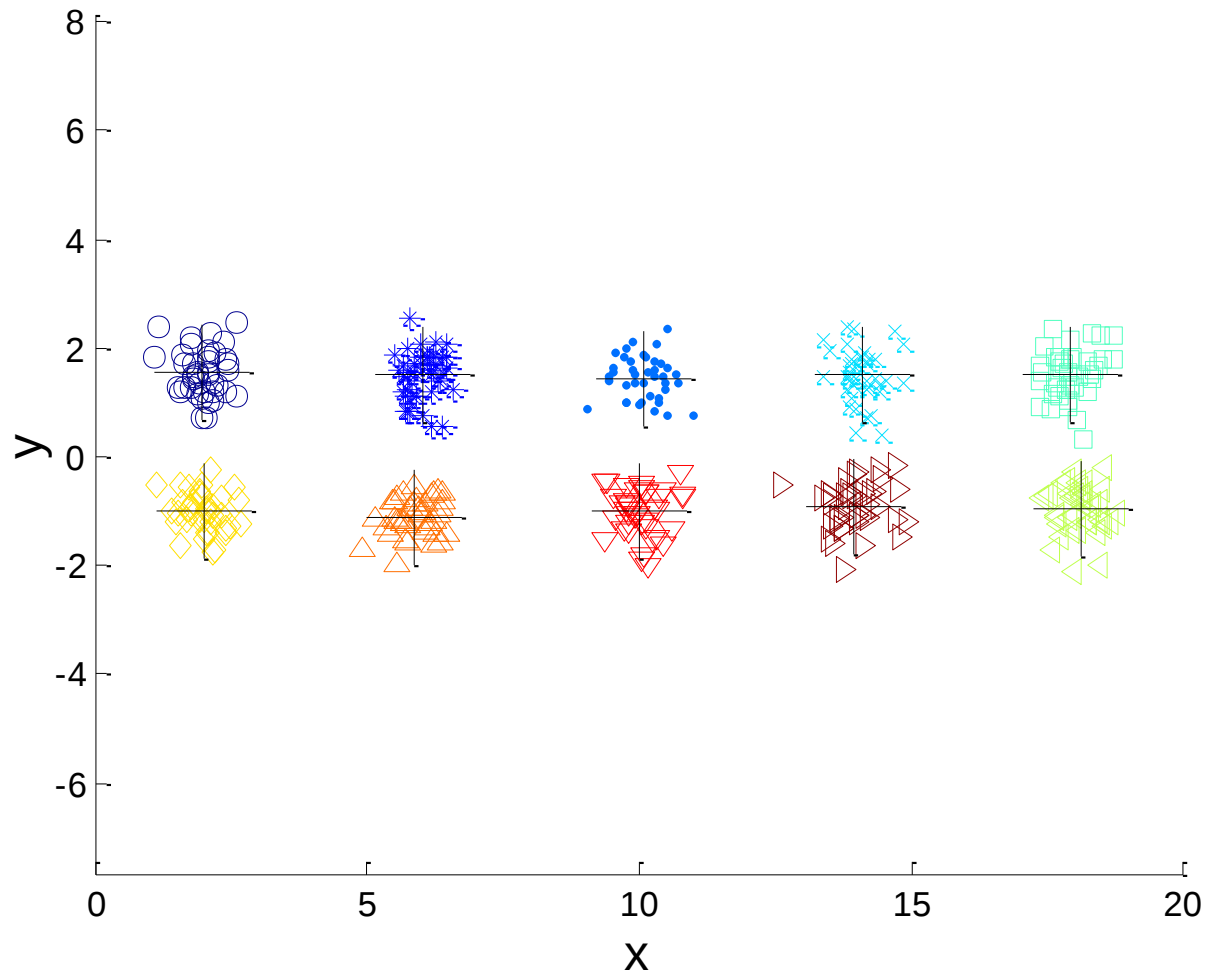
$$P = \frac{\text{no. de maneiras de selecionar 1 objeto de cada grupo (com } N / k \text{ objetos)}}{\text{no. de maneiras de selecionar } k \text{ dentre } N \text{ objetos}}$$

# Análise da Seleção dos Protótipos Iniciais

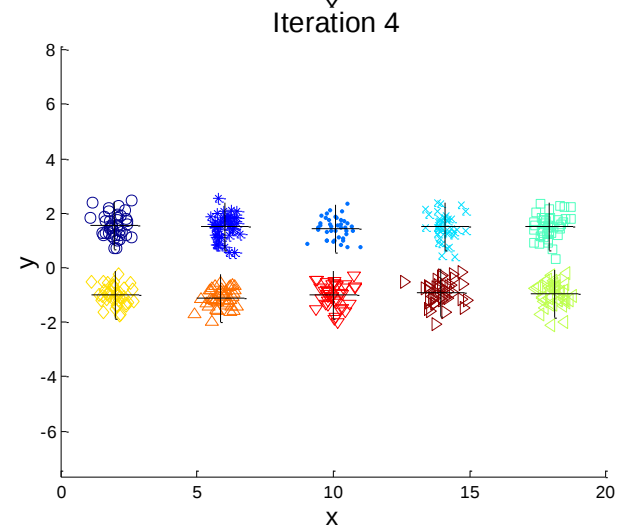
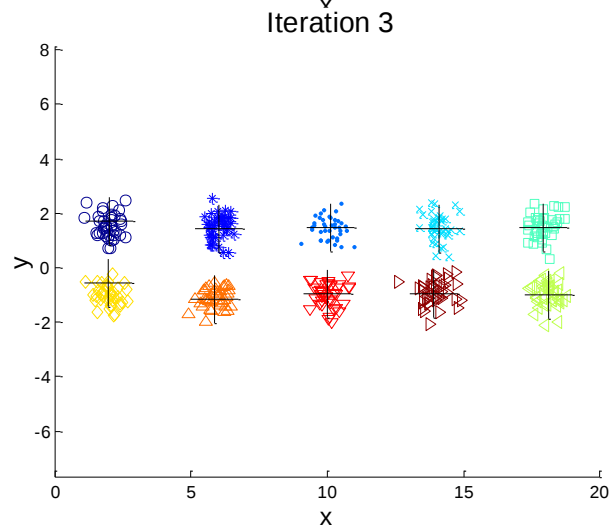
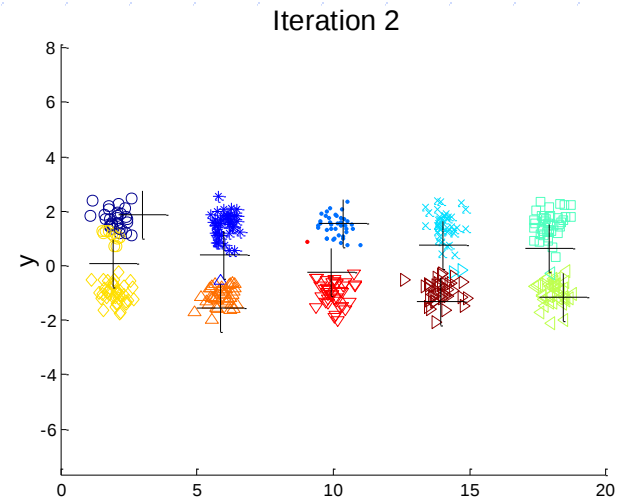
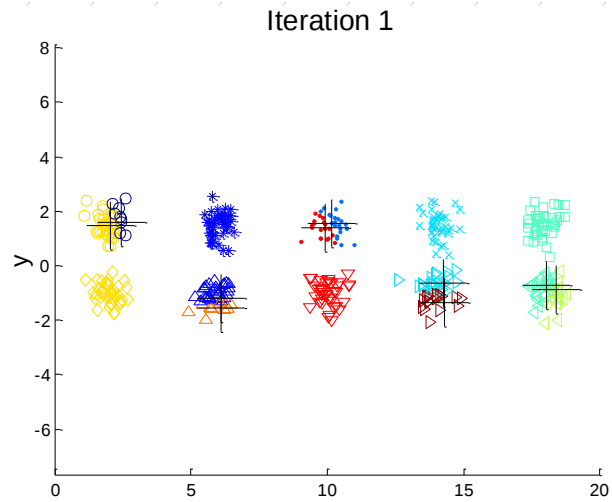
- ❑ N° de formas de selecionar k protótipos (denominador)
  - ❑ cada um dos  $N = k \cdot g$  objetos pode ser selecionado em cada um dos k sorteios, com reposição, logo tem-se  $(k \cdot g)^k$  formas
- ❑ N° de formas de escolher 1 protótipo por grupo (numerador)
  - ❑ No 1º sorteio, qualquer um dos  $N = k \cdot g$  objetos pode ser selecionado. No 2º sorteio, qualquer objeto exceto aqueles g do mesmo grupo do 1º sorteio podem ser selecionados, ou seja,  $k \cdot g - g = (k - 1) \cdot g$  podem ser selecionados, e assim por diante. Logo, tem-se  $k \cdot g \times (k - 1) \cdot g \times \dots \times g = k!g^k$
- ❑ Portanto, tem-se  $P = k!g^k / k^k g^k \rightarrow P = k! / k^k$
- ❑ Exemplo: se  $k = 10$ ,  $P = 0.00036$

## Exemplo: Iniciando com 2 centróides iniciais em um grupo de cada par...

Iteration 4

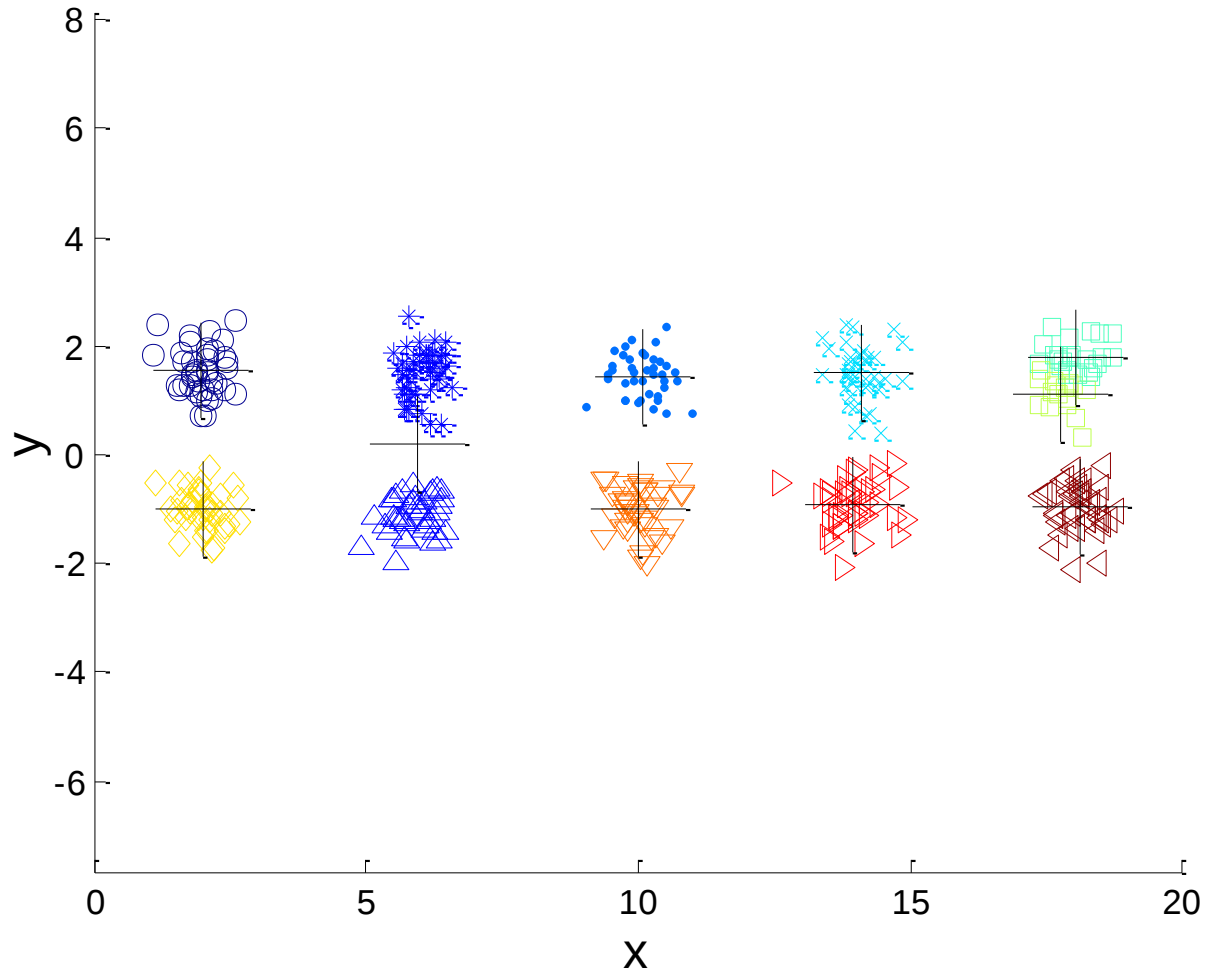


# Ilustrando todas as iterações:

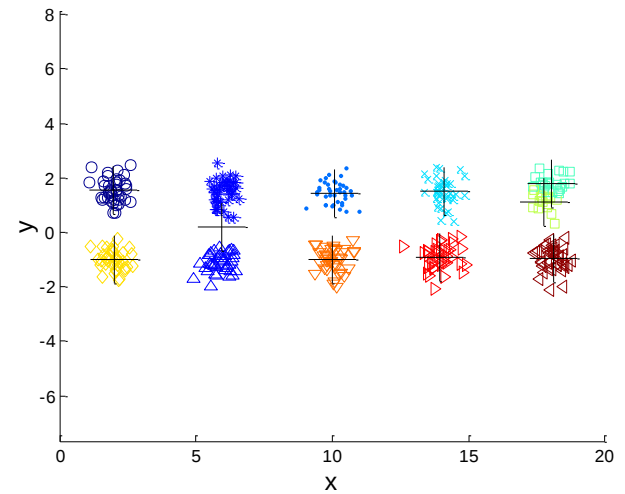
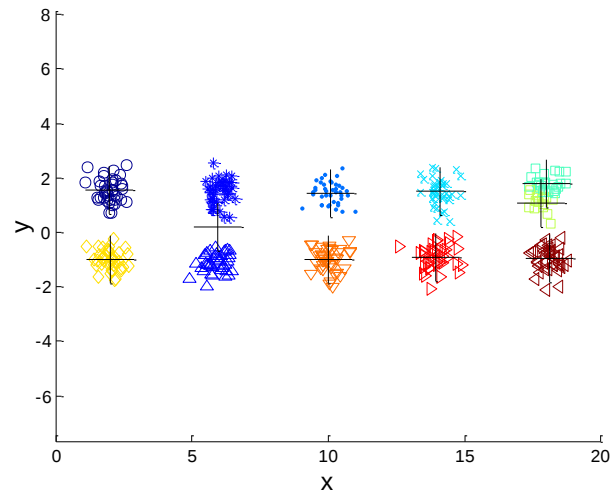
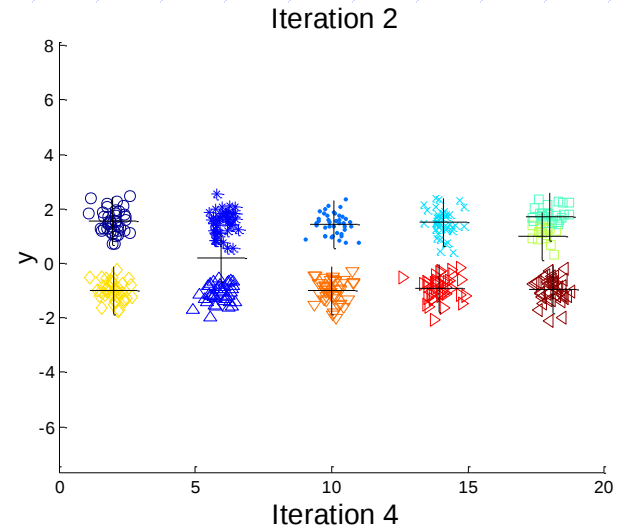
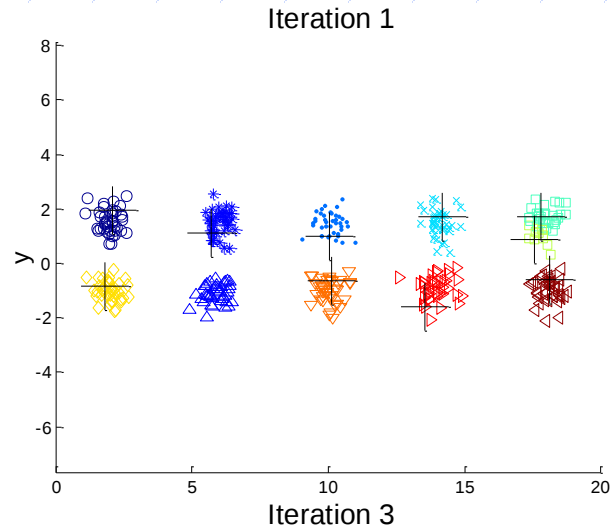


# Agora vejamos outra inicialização:

Iteration 4



# Ilustrando todas as iterações:



# Alternativas para Inicialização

- ❑ Múltiplas Execuções (inicializações aleatórias):
  - ❑ Funciona bem em muitos problemas.
  - ❑ Em bases de dados complexas, pode demandar um no. enorme de execuções, em particular para  $k$  grande.
- ❑ Agrupamento Hierárquico:
  - ❑ agrupa-se uma amostra dos dados
  - ❑ tomam-se os centros da partição com  $k$  grupos

# Alternativas para Inicialização

## ❑ Seleção “Informada” :

- ❑ toma-se 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
- ❑ **Nota:** para reduzir o esforço computacional e minimizar a probabilidade de seleção de outliers
  - processa-se apenas uma amostra dos dados

## ❑ Busca Guiada:

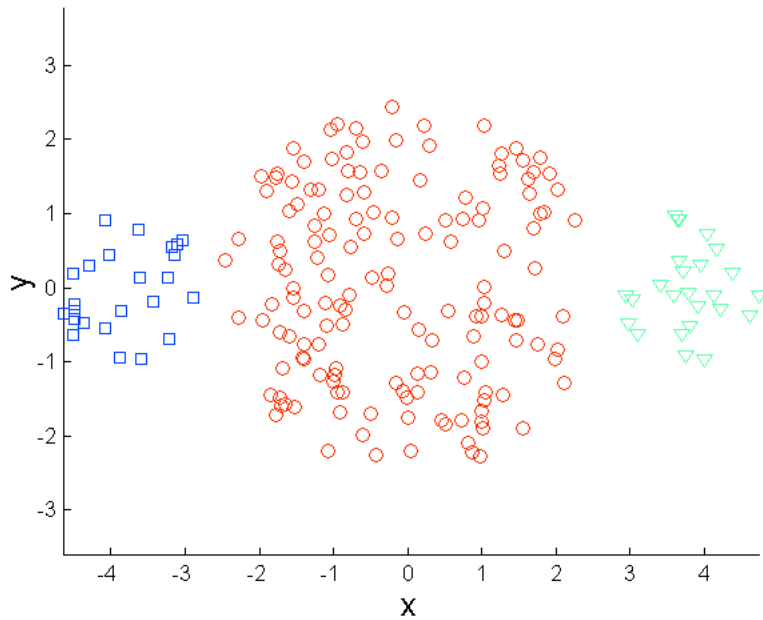
- ❑ **X-means, k-means evolutivo, ...**

# Discussão

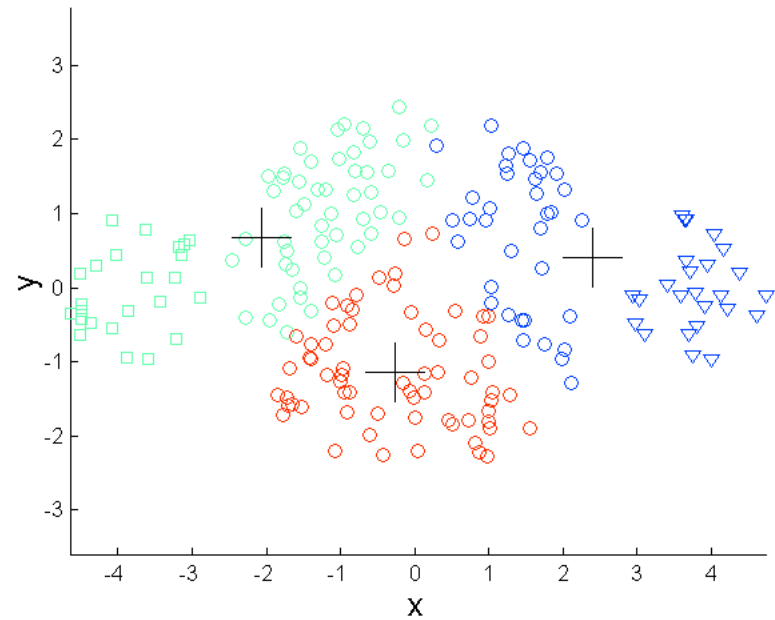
---

- ❑ k-means é mais susceptível a problemas quando clusters são de diferentes
  - Tamanhos
  - Densidades
  - Formas não-globulares

# Differing Sizes

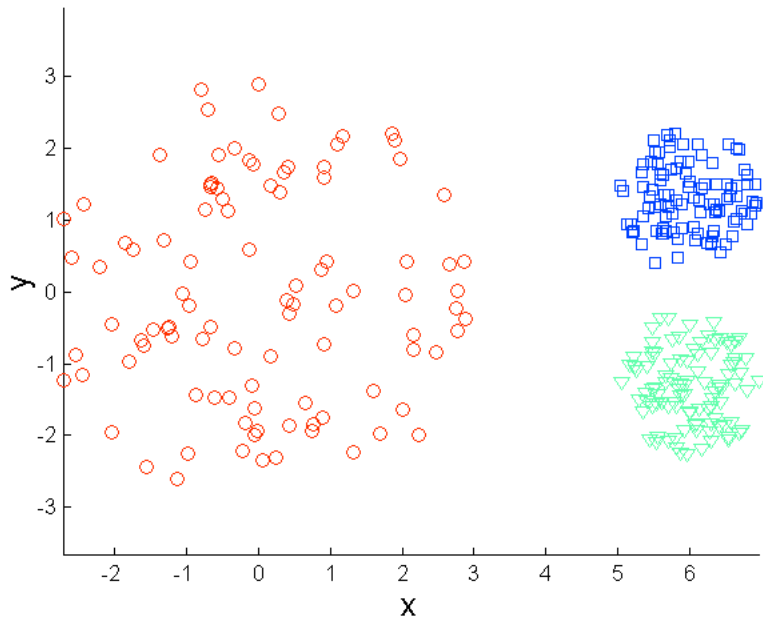


**Original Points**

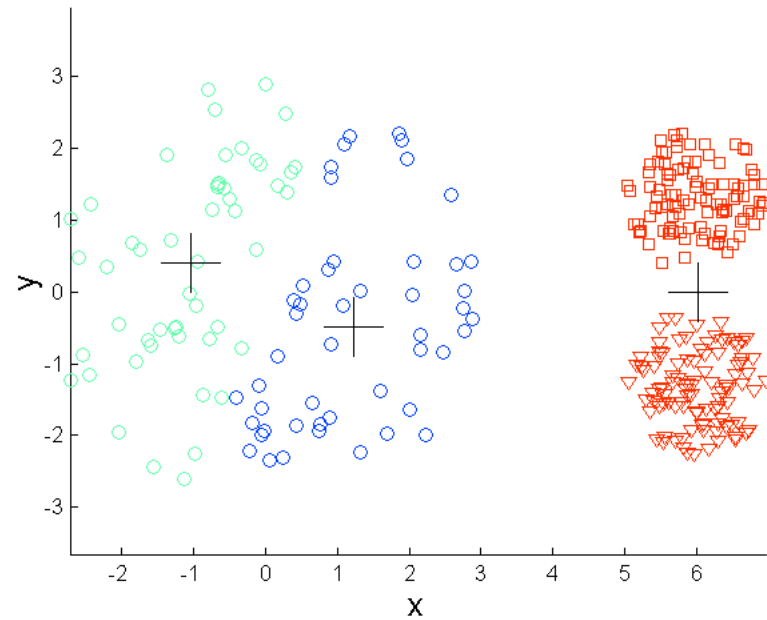


**K-means (3 Clusters)**

# Differing Density



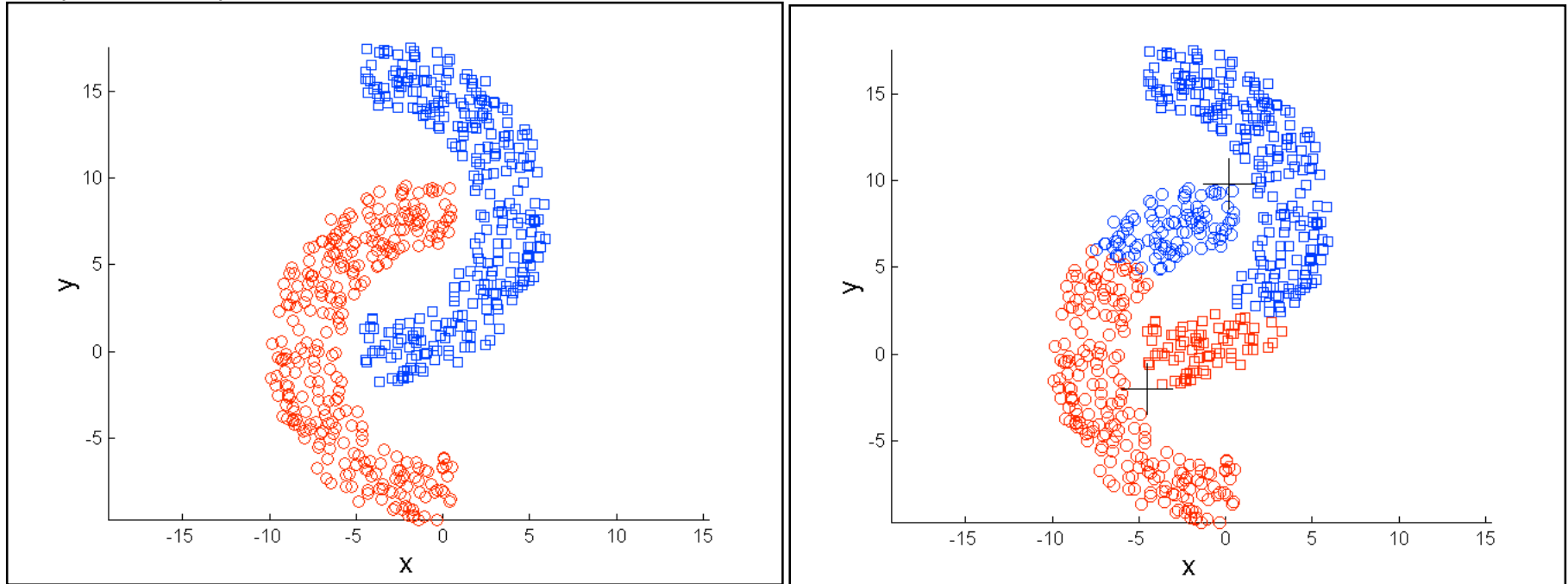
**Original Points**



**K-means (3 Clusters)**

# Formas Não-Globulares

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
  - Grandes BDs (muitos objetos & atributos) e necessidade de interpretação dos resultados (e.g. segmentação de mercado...)

# Implementações Eficientes

- Desempenho computacional pode ser melhorado...
  - **Estruturas de Dados**, e.g.
    - **kd-trees**
  - **Algoritmos**, e.g.
    - **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
        - Não demanda recalcular tudo novamente
      - **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)

# K-Means Paralelo / 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.

# Exercício

| Objeto $x_i$ | $x_{i1}$ | $x_{i2}$ | Processador |
|--------------|----------|----------|-------------|
| 1            | 1        | 2        | A           |
| 2            | 2        | 1        | B           |
| 3            | 1        | 1        | A           |
| 4            | 2        | 2        | B           |
| 5            | 8        | 9        | A           |
| 6            | 9        | 8        | B           |
| 7            | 9        | 9        | B           |
| 8            | 8        | 8        | A           |
| 9            | 1        | 15       | B           |
| 10           | 2        | 15       | A           |
| 11           | 1        | 14       | B           |
| 12           | 2        | 14       | A           |

Executar k-means paralelo nos dados ao lado, com  $k=3$ , a partir dos protótipos iniciais  $[6 \ 6]$ ,  $[4 \ 6]$  e  $[5 \ 10]$

# Resumo do k-means

## Vantagens

- Simples e intuitivo
- Complexidade computacional **linear** em todas as variáveis críticas
- Eficaz em muitos cenários de aplicação e produz resultados de interpretação simples
- Considerado um dos 10 mais influentes algoritmos em Data Mining (Wu & Kumar, 2009).

## 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**, ou seja, sem sobreposição)
- Limitado a atributos numéricos
- Sensível a *outliers*

# Generalização:

- Banerjee et al. (*Clustering with Bregman Divergences*, JMLR, 2005) apresentam uma visão unificada para a classe de algoritmos de agrupamento baseados em centróides (*ao estilo k-means*);
- Estudo teórico minucioso baseado em Divergentes de Bregman:

$$d\phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla \phi(\mathbf{y}) \rangle$$

- Distância Euclidiana ao quadrado, Mahalanobis, KL, perda quadrática, perda logística, etc.;
  - Exemplo:  $\phi(\mathbf{x}) = \mathbf{x}^T \mathbf{x}$  é estritamente convexa e diferenciável em  $\mathbb{R}^n$ ...
- Algoritmos derivados mantêm simplicidade, garantias teóricas e escalabilidade do k-médias;
- Aplicável a diferentes tipos de atributos --- diferentes funções convexas escolhidas para diferentes subconjuntos de atributos.

# Algumas Variantes do k-means

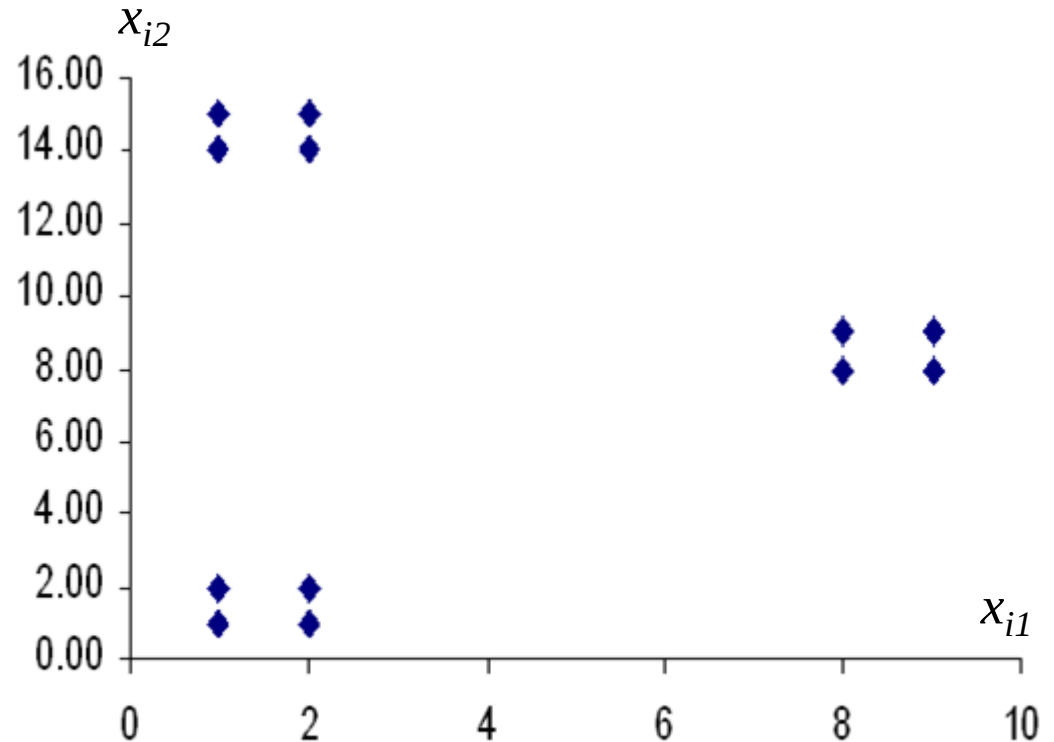
- **K-medianas:** Substituir as médias pelas medianas
  - Média de 1, 3, 5, 7, 9 é 5
  - Média de 1, 3, 5, 7, 1009 é 205
  - Mediana de 1, 3, 5, 7, 1009 é 5
  - **Vantagem:** menos sensível a outliers\*
  - **Desvantagem:** implementação mais complexa
    - cálculo da mediana em cada atributo...
- Pode-se mostrar que minimiza a soma das **distâncias de Manhattan** dos objetos aos centros (medianas) dos grupos

# Algumas Variantes do k-means

- **K-medóides:** Substituir cada centróide por um objeto representativo do cluster, denominado **medóide**
  - Medóide = objeto mais próximo aos demais objetos do cluster
    - mais próximo em média (empates resolvidos aleatoriamente)
  - **Vantagens:**
    - menos sensível a outliers
    - permite cálculo relacional (apenas matriz de distâncias)
      - logo, pode ser aplicado a bases com atributos categóricos
    - convergência assegurada com qualquer medida de (dis)similaridade
  - **Desvantagem:** Complexidade quadrática com nº. de objetos (N)

# Exercício

| Objeto $x_i$ | $x_{i1}$ | $x_{i2}$ |
|--------------|----------|----------|
| 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-medóides com  $k=3$  nos dados acima, com medóides iniciais dados pelos objetos 5, 6 e 8

# Algumas Variantes do k-means

- **K-means para Fluxos de Dados (Data Streams):**
  - Em geral, usa conceito de vizinhos mais próximos (K-NN)
  - Objetos dinamicamente incorporados ao cluster mais próximo
    - Atualização do centróide do cluster pode ser **incremental**
      - centróide atualizado a cada novo objeto incorporado
      - mas isso introduz dependência de ordem dos dados...
  - Heurísticas podem ser usadas para criar e/ou remover clusters
  - Ver Silva, Faria, Barros, Hruschka, de Carvalho, Gama, *Data Stream Clustering: A Survey*, **ACM Computing Surveys**.

# Algumas Variantes do k-means

- **Métodos de Múltiplas Execuções de k-means:**
  - Executam k-means repetidas vezes a partir de diferentes valores de  $k$  e de posições iniciais dos protótipos
    - Ordenado:  $n_p$  inicializações de protótipos para cada  $k \in [k_{\min}, k_{\max}]$
    - Aleatório:  $n_T$  inicializações de protótipos 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

# Questão...

- J poderia ser utilizada para escolher a melhor partição dentre um conjunto de candidatas ?
  - Resposta é sim se todas têm o mesmo no. k de clusters (fixo)
  - Mas 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  $k \in [k_{\min}, k_{\max}]$

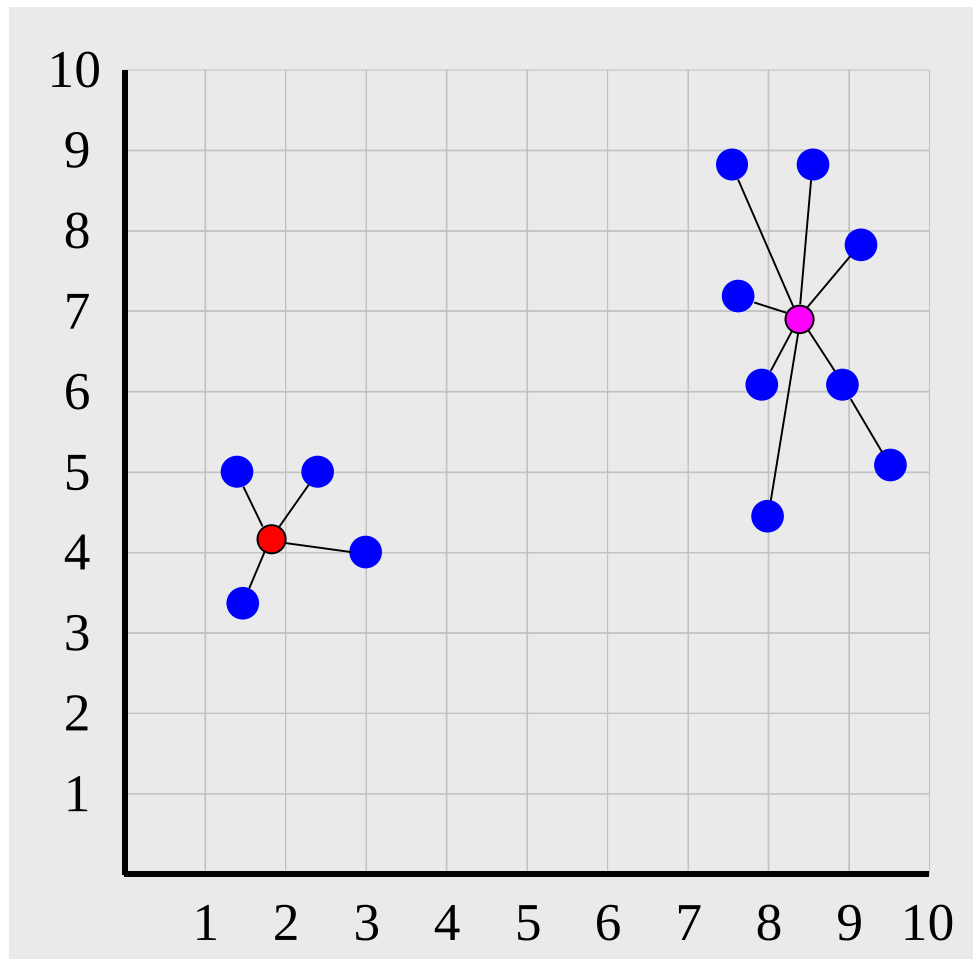
# Questão...

- Para responder a questão anterior, vamos considerar o método de múltiplas execuções ordenadas de k-means, com uso da função objetivo J

Erro Quadrático:

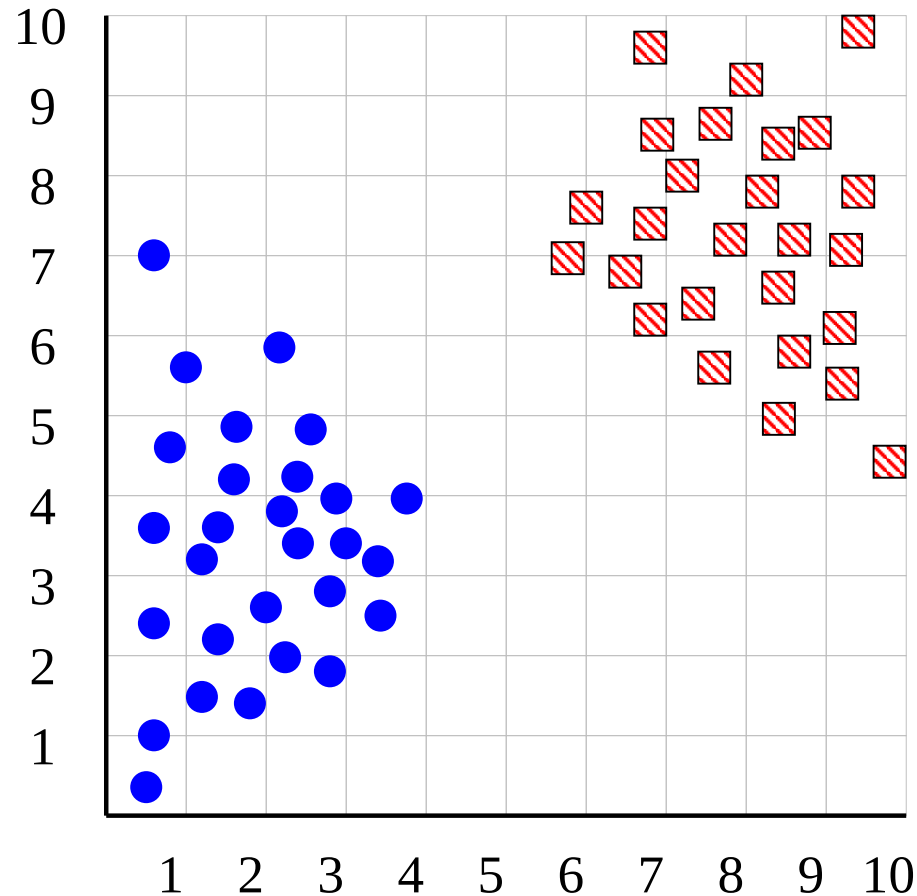
$$J = \sum_{i=1}^k \sum_{\mathbf{x}_j \in \mathbf{C}_i} d(\mathbf{x}_j, \bar{\mathbf{x}}_i)^2$$

Função Objetivo

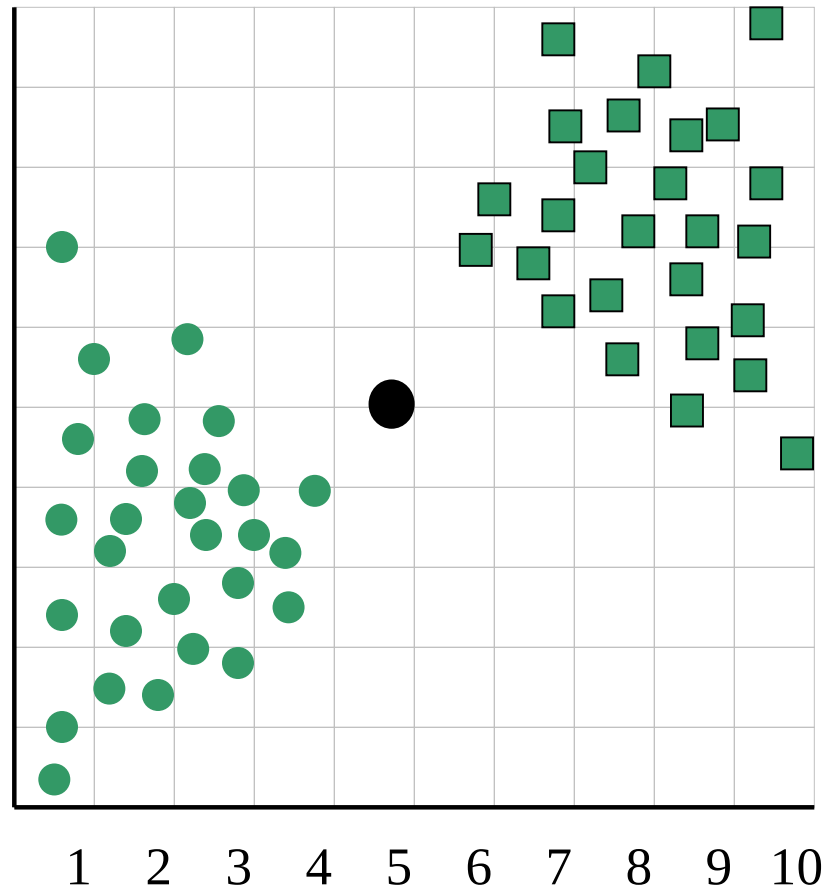


# Questão...

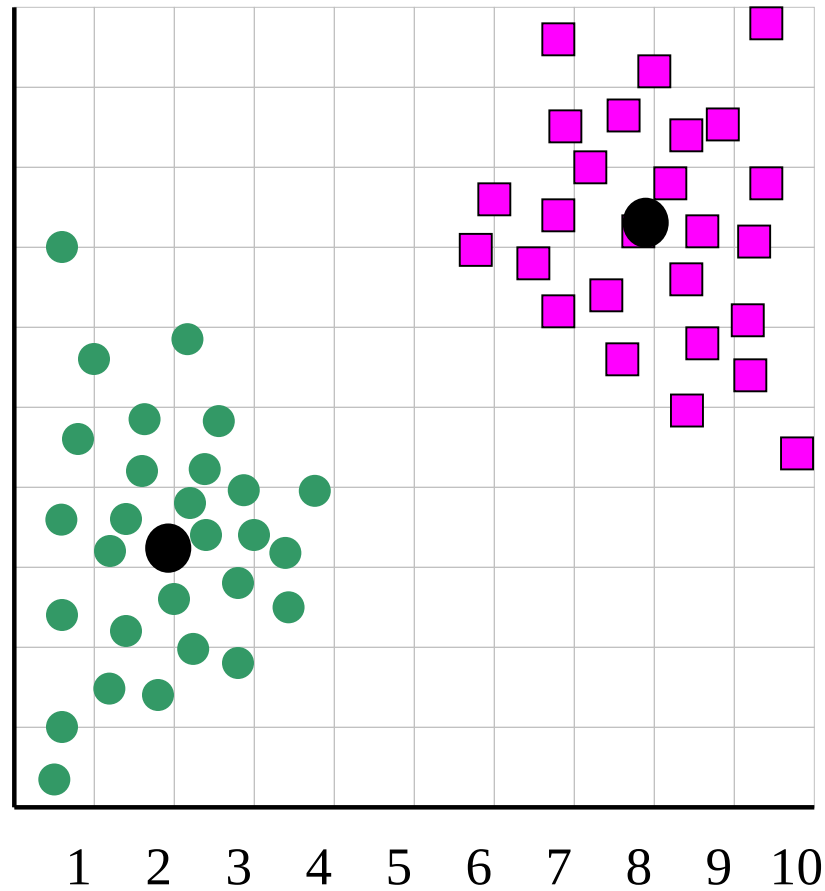
- Considere o seguinte exemplo:



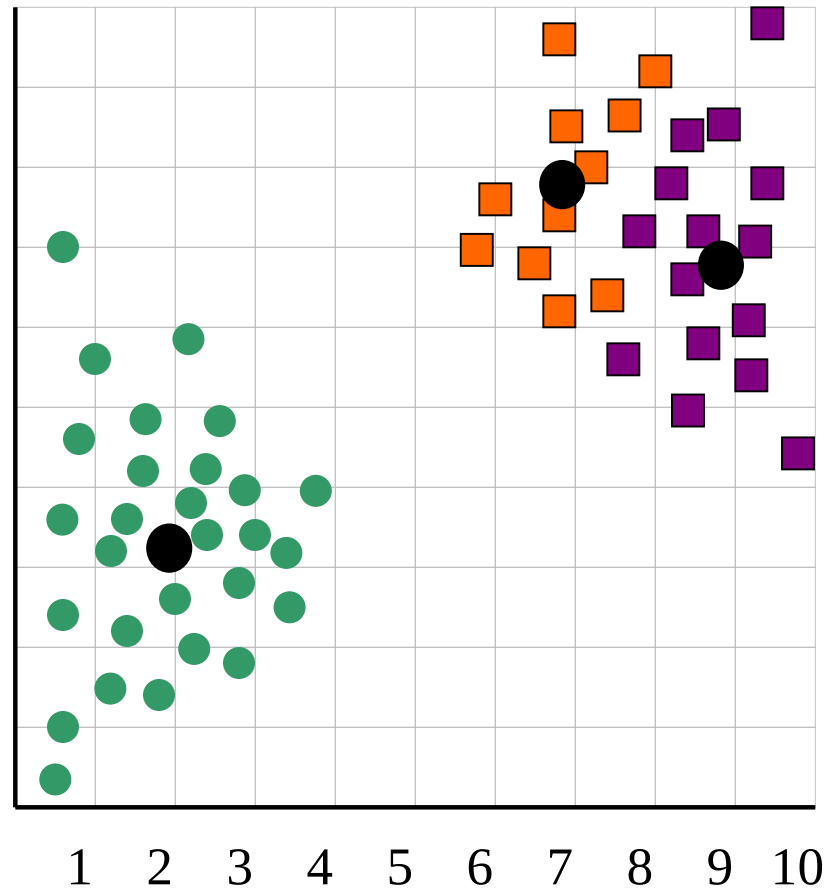
Para  $k = 1$ , o valor da função objetivo é 873,0



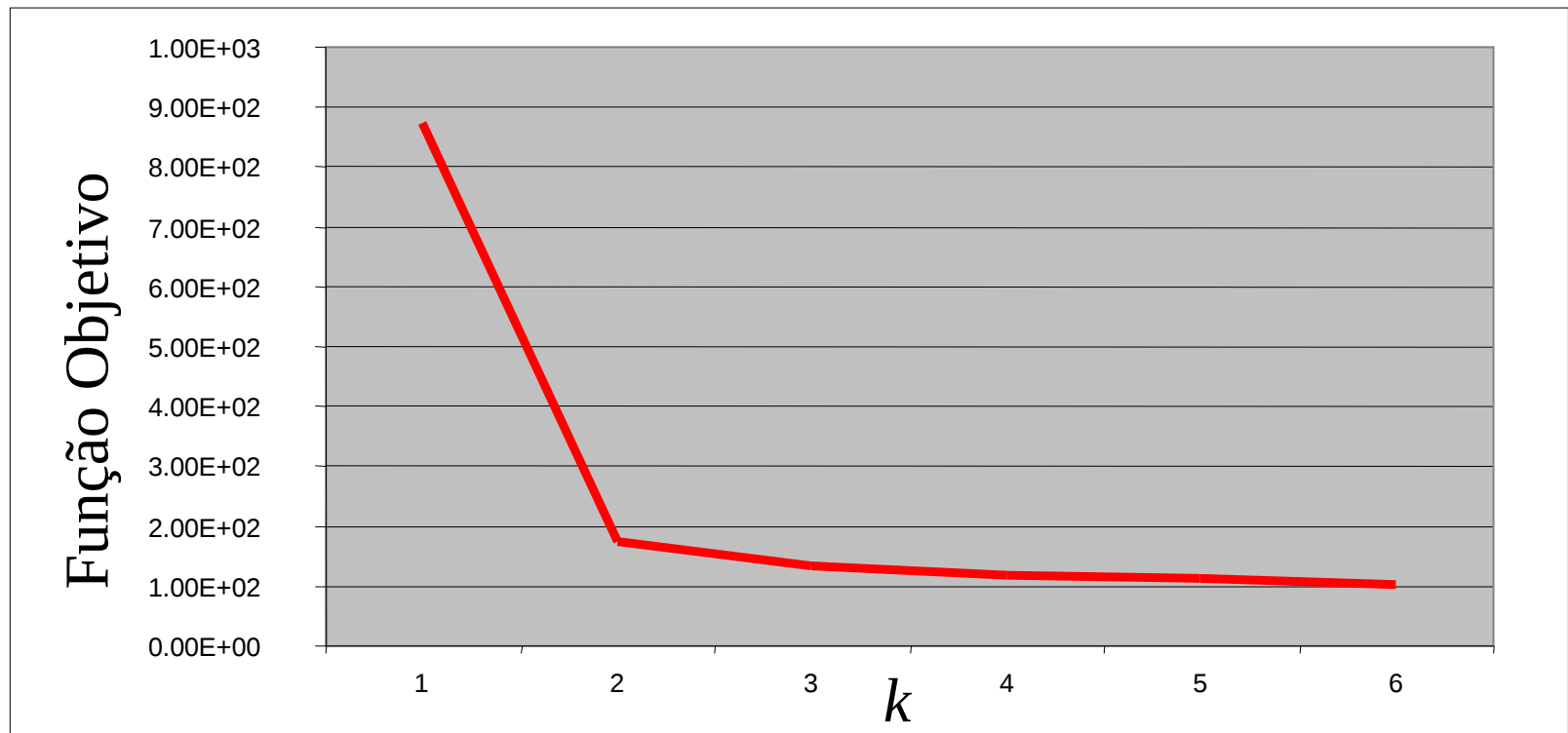
Para  $k = 2$ , o valor da função objetivo é 173,1



Para  $k = 3$ , o valor da função objetivo é 133,6

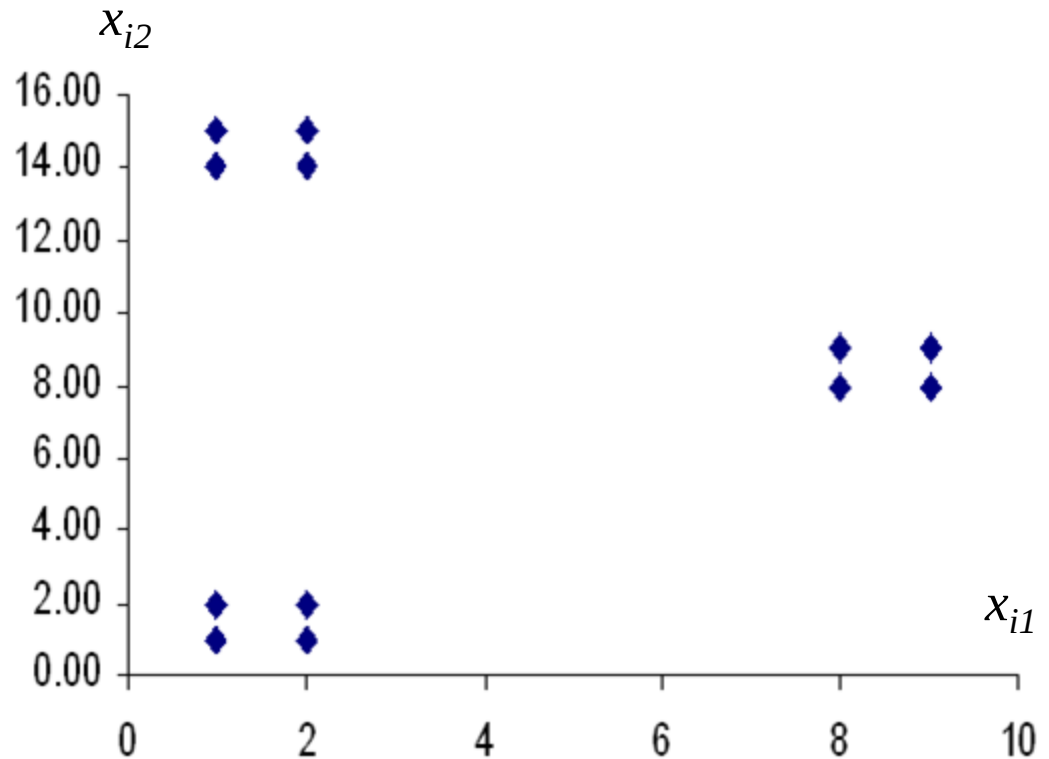


Podemos então repetir este procedimento e plotar os valores da função objetivo  $J$  para  $k=1, \dots, 6, \dots$  e tentar identificar um “joelho” :



# Exercício

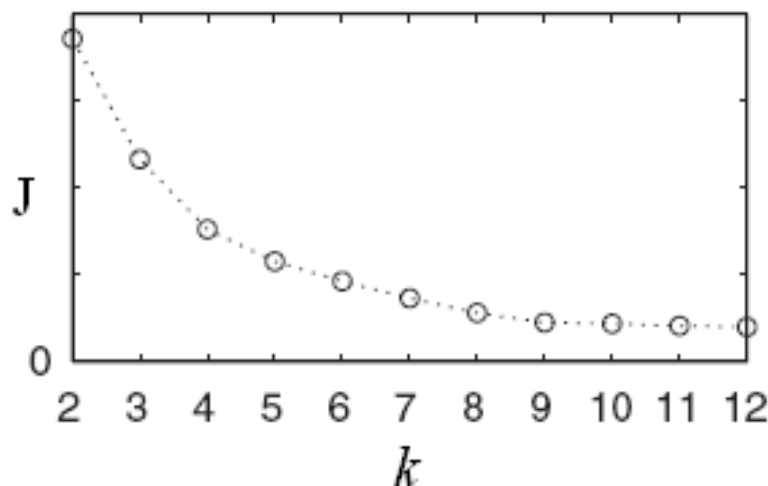
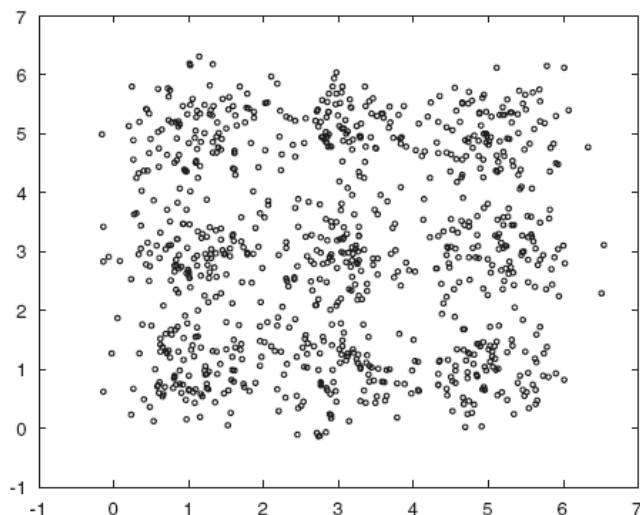
| Objeto $x_i$ | $x_{i1}$ | $x_{i2}$ |
|--------------|----------|----------|
| 1            | 1        | 2        |
| 2            | 2        | 1        |
| 3            | 1        | 1        |
| 4            | 2        | 2        |
| 5            | 8        | 9        |
| 6            | 9        | 8        |
| 7            | 9        | 9        |
| 8            | 8        | 8        |
| 9            | 1        | 15       |
| 10           | 2        | 15       |
| 11           | 1        | 14       |
| 12           | 2        | 14       |



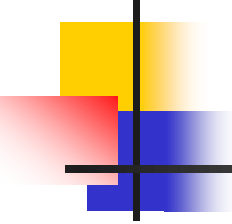
- Executar k-means com  $k=2$  até  $k=5$  nos dados acima e representar graficamente a f. objetivo  $J$  em função de  $k$

# Questão...

- Infelizmente os resultados não são sempre tão claros quanto no exemplo anterior... Vide exemplo abaixo...



- Além disso, como utilizar essa metodologia em variantes baseadas em busca guiada, que otimizam  $k$  ?
  - X-means, k-means evolutivo, ...
- Exploraremos mais em **validação de agrupamentos**



# Referências

---

- 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