# Mineração de Dados 2017.2

Algoritmos de Agrupamento - Particionais

Thiago Ferreira Covões

# Créditos

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

# Definição de Partição de Dados (Revisão)

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

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

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

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

  - 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** é uma matriz com *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 ($\mathbf{C}_i$)

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

- 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:
  - *matriz de partição **rígida**, **exclusiva*** ou *sem **sobreposição***

# Matriz de Partição

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

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

# Métodos Particionais (Sem Sobreposição)

- Métodos *particionais* sem sobreposição referem-se a algoritmos de agrupamento que buscam (explícita ou implicitamente) por uma matriz de partição rígida de um conjunto de objetos **X**

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

# Particionamento como Problema Combinatório

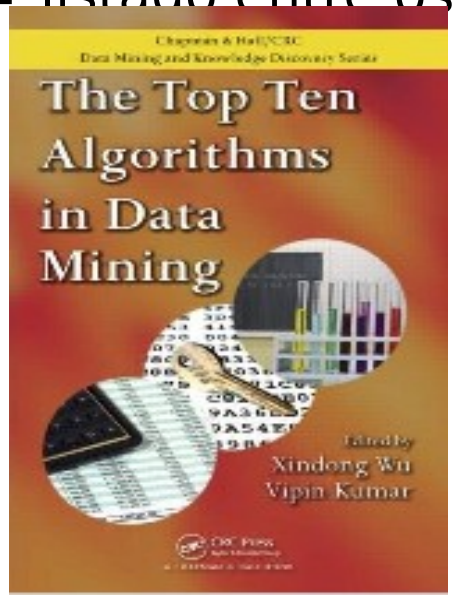
- **Problema**: Assumindo que *k* seja conhecido, o no. de possíveis formas de agrupar *N* objetos em *k clusters* é dado por (Liu, 1968):

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

- Por exemplo, NM(100, 5) $\approx$ 56.6x10$^{67}$.
  - Em um computador com capacidade de avaliar 10$^{9}$ partições/s, precisaríamos $\approx$ 1.8x10$^{50}$ 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 dos algoritmos mais clássicos da área de **mineração de dados** em geral
  - algoritmo das **k-médias** ou ***k-means***
  - listado entre os **Top 10 Most Influential Algorithms in**




- 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, 1967.

- **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

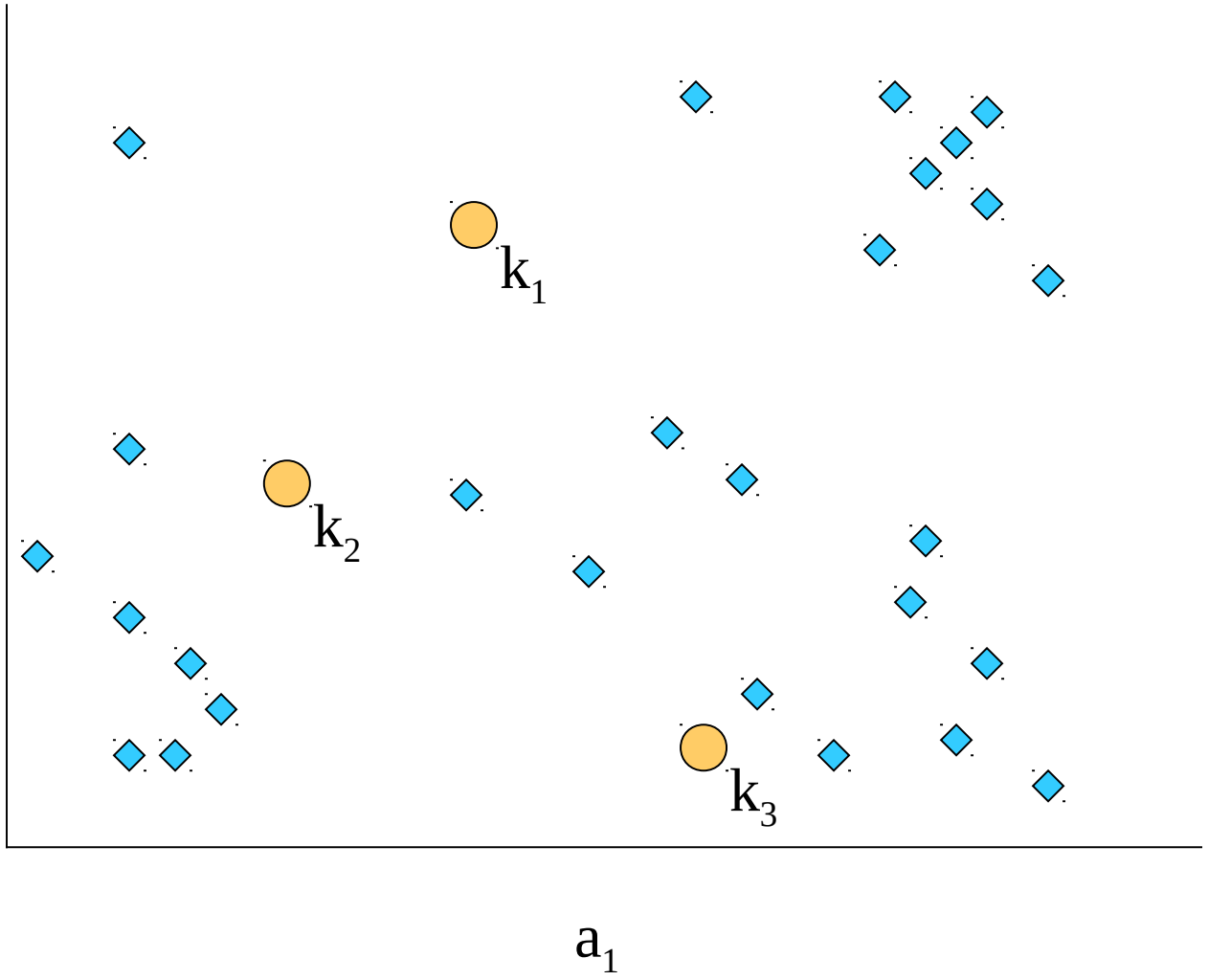
# k-Means

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

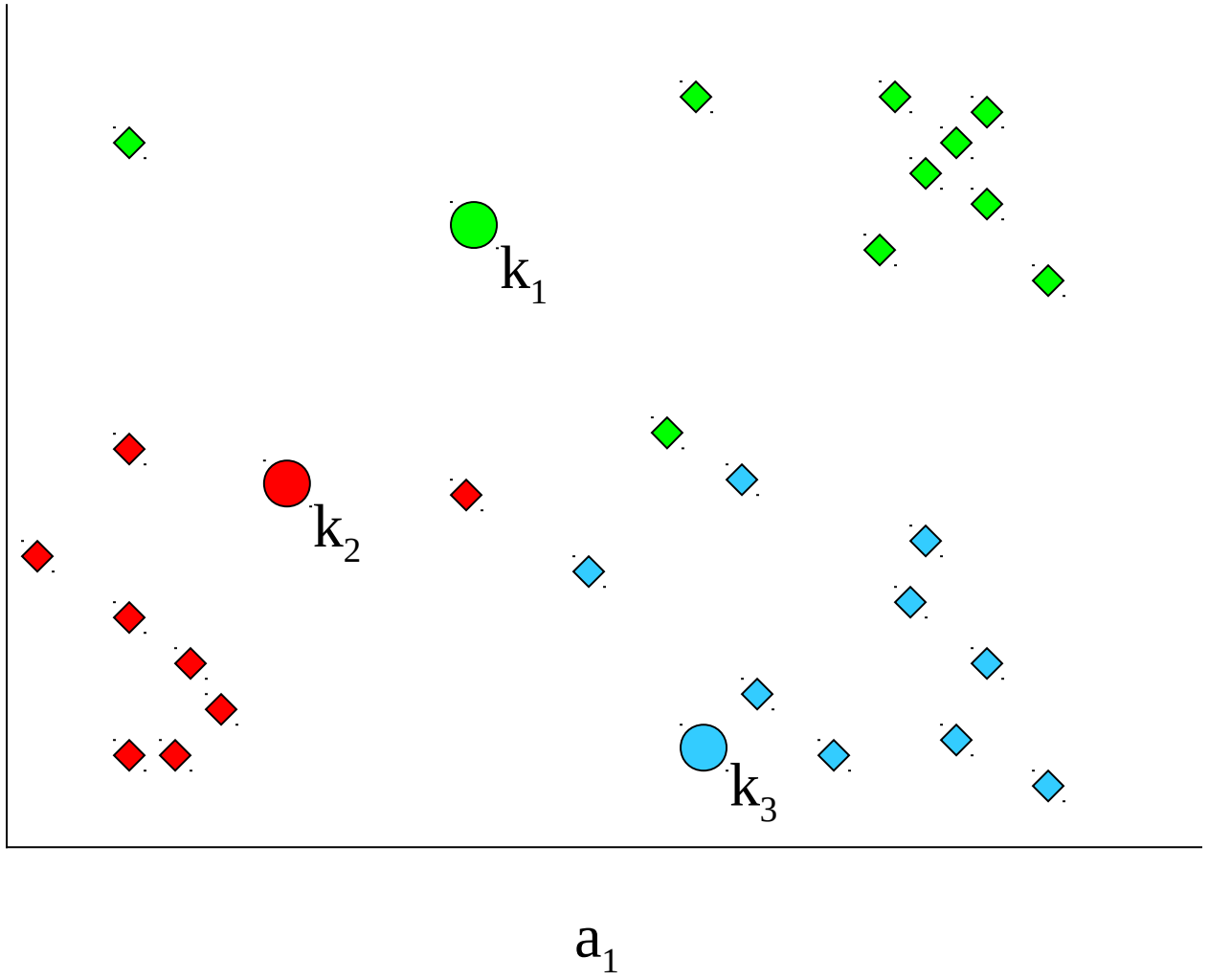
# k-Means - passo 1:




Escolher 3 centros iniciais

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

# k-Means - passo 2:

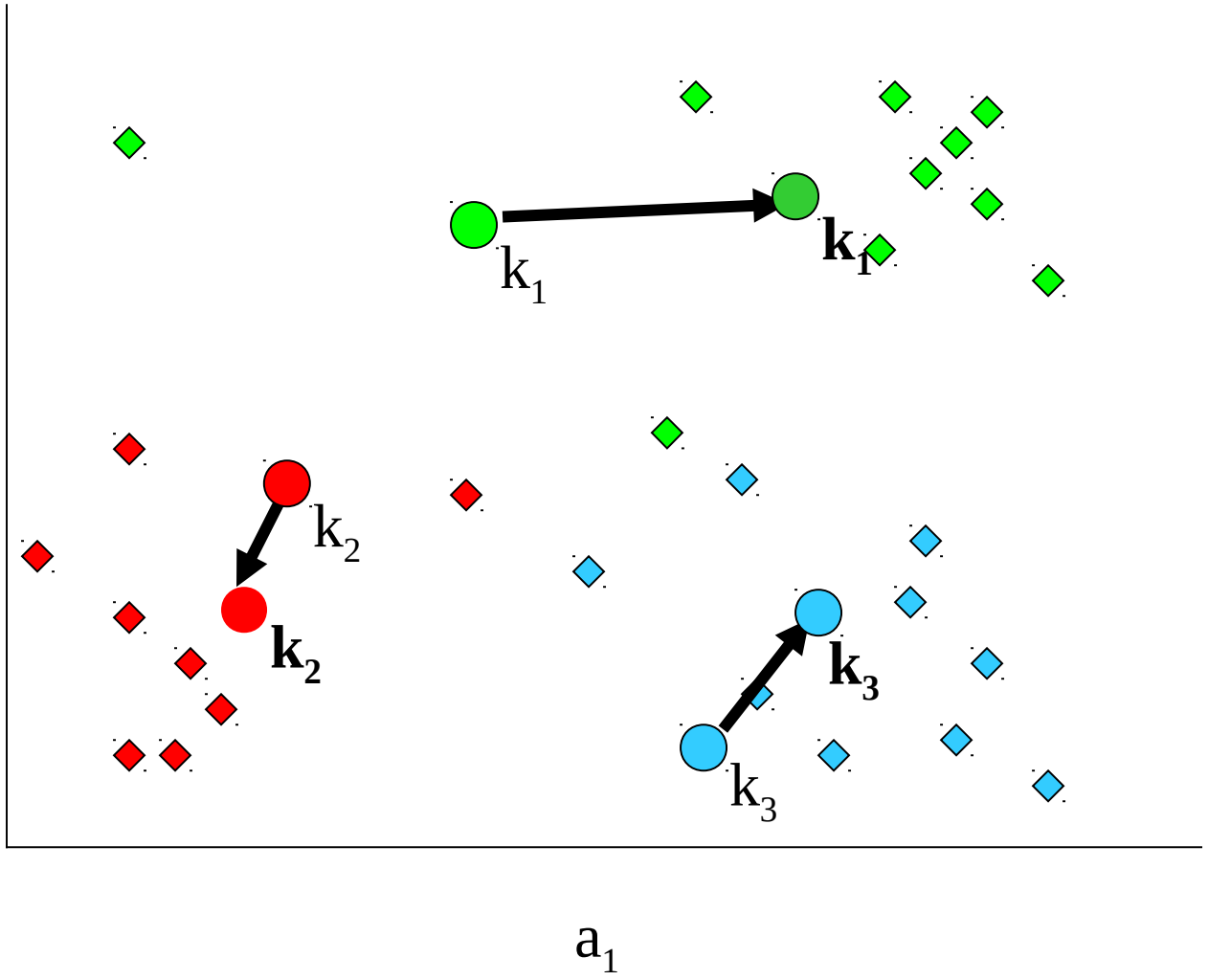



Atribuir cada objeto ao cluster de centro + próximo

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 (centróide)

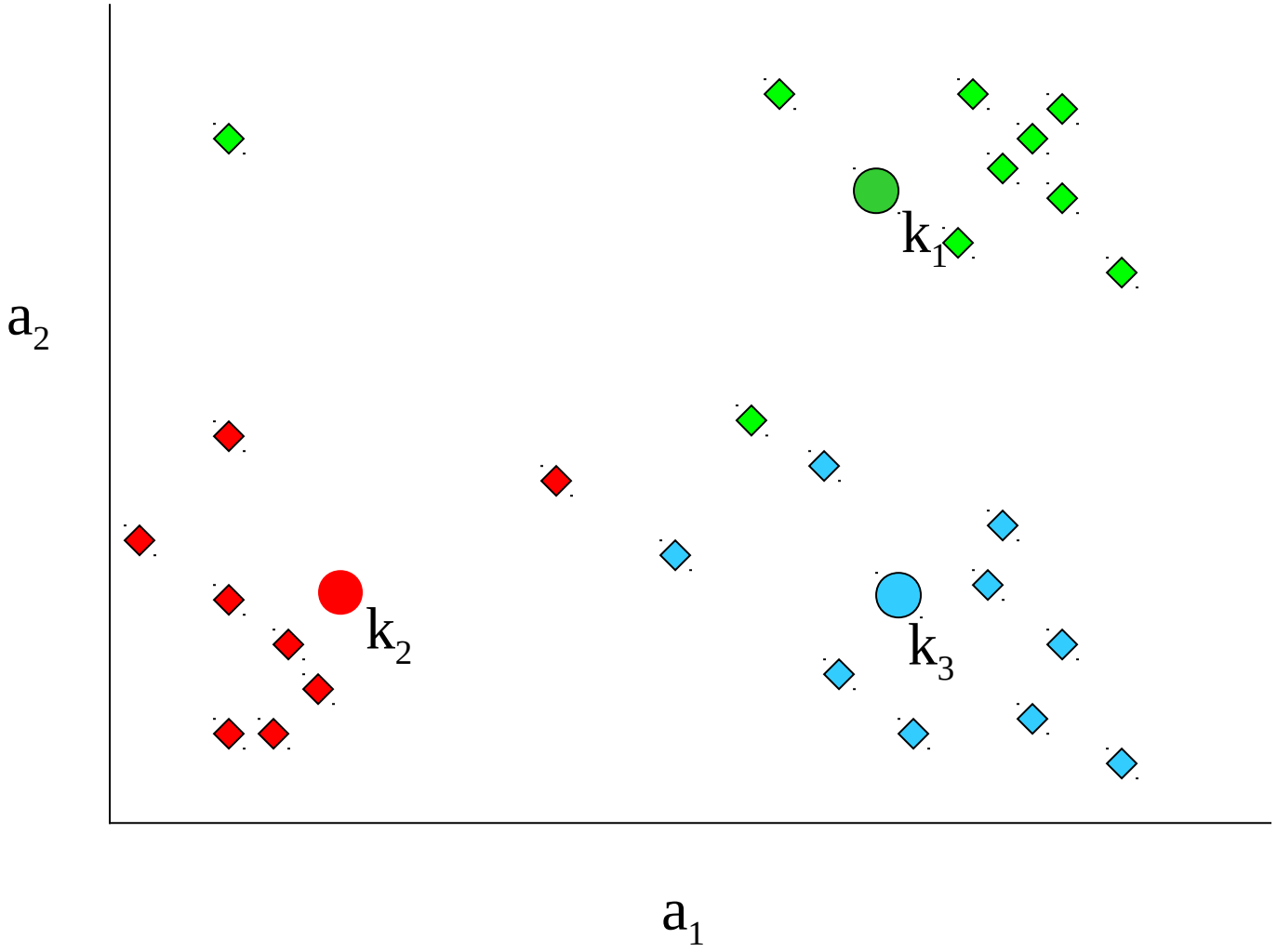



Slide baseado no curso de Gregory Piatetsky-Shapiro, disponível em http://www.kdnuggets.com

# k-Means:

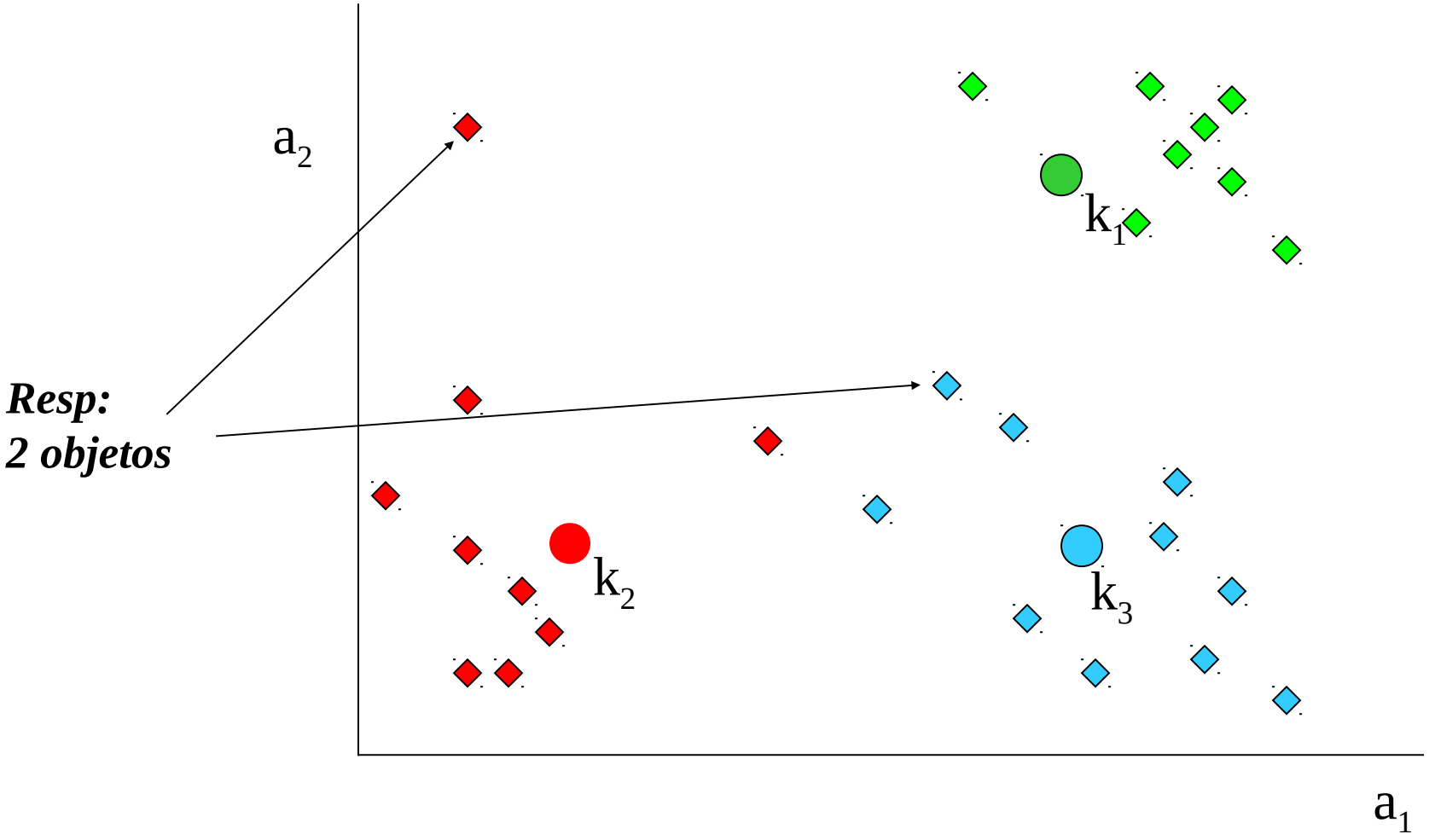
Re-atribuir objetos aos clusters de centróides mais próximos

Quais objetos mudarão de cluster?




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

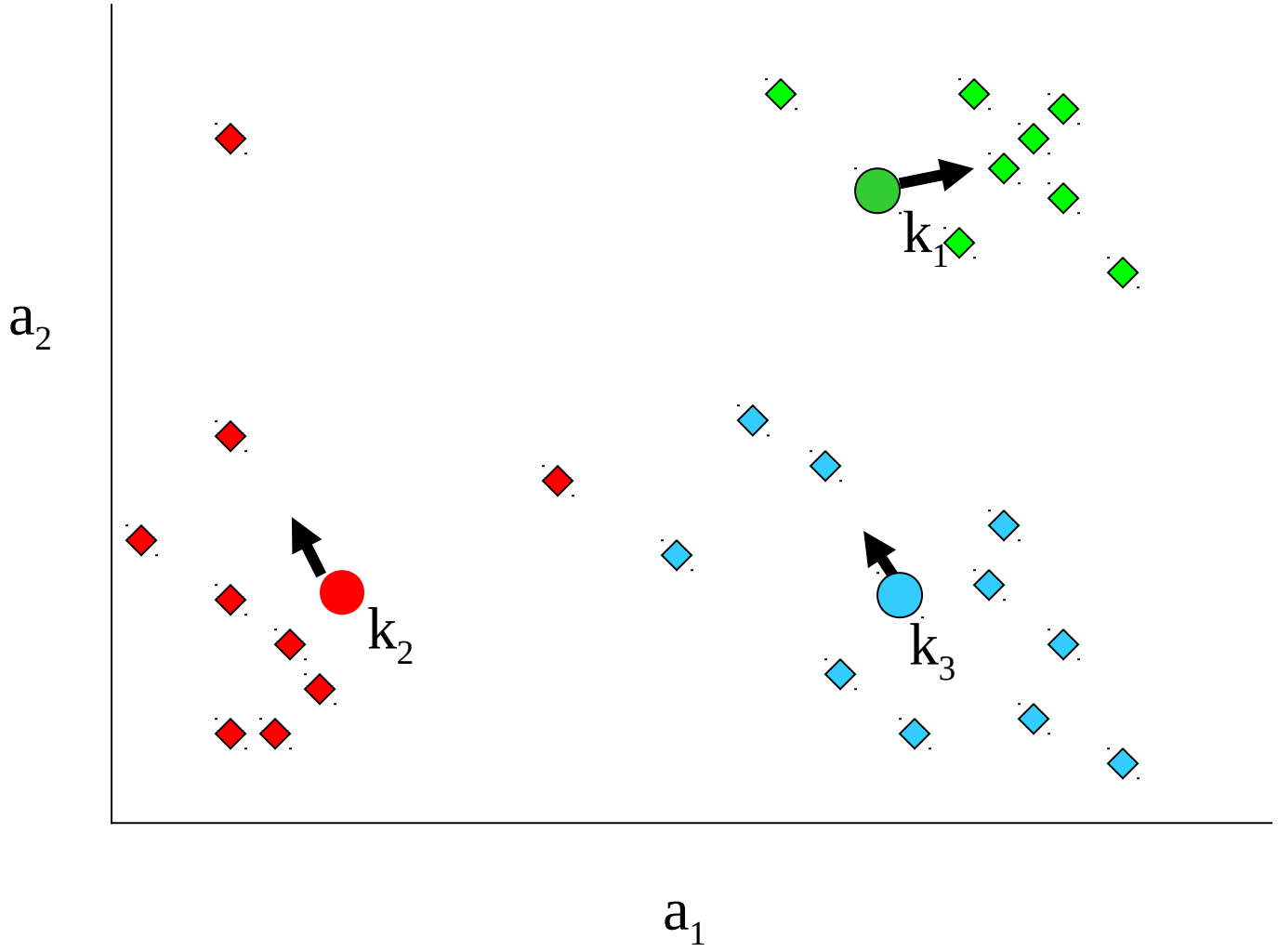
# k-Means:




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

# k-Means:

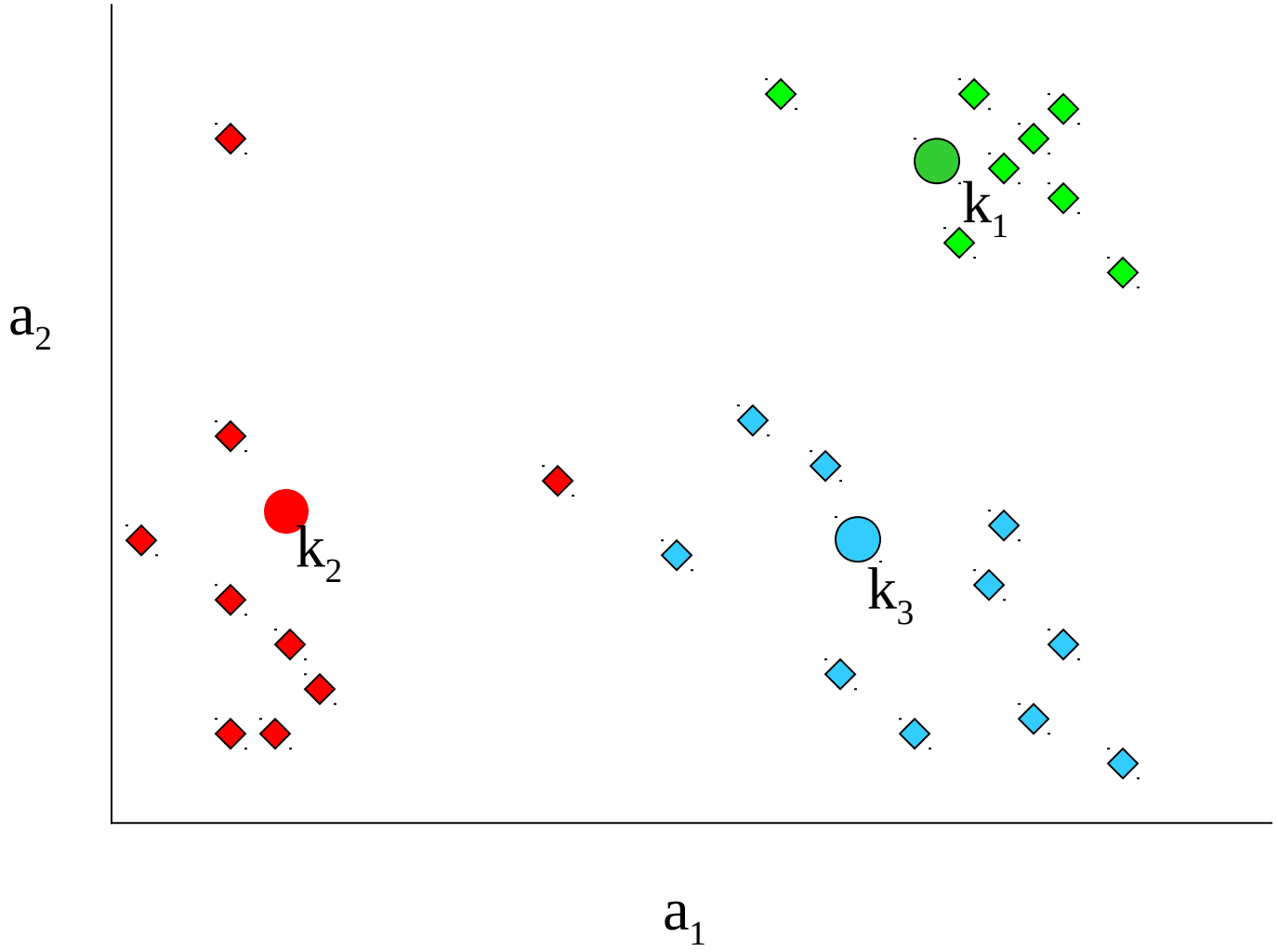
re-calcular vetores médios




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

# k-Means:

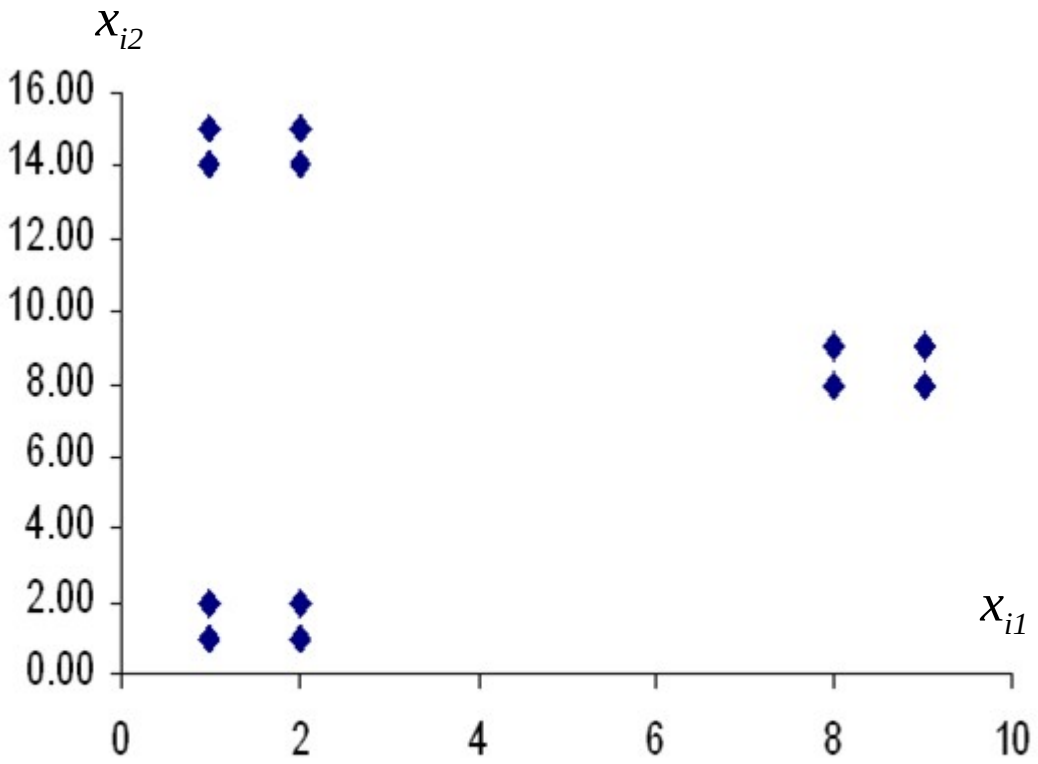
mover centros dos clusters…




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

# 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 |




- 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 Erros* (**variâncias intra-cluster)**

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

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

$$\bar{x}_c = \frac{1}{|C_c|} \sum_{x_j \in C_c} x_j$$

# K-Means sob a Perspectiva de Otimização:

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

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

- Desejamos minimizar $J$ com respeito a $\{\overline{\mathbf{x}}_c\}$ e $\{\mu_{cj}\}$
- Pode-se fazer isso via um procedimento iterativo (2 passos):

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

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

# K-Means sob a Perspectiva de Otimização:

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

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**)

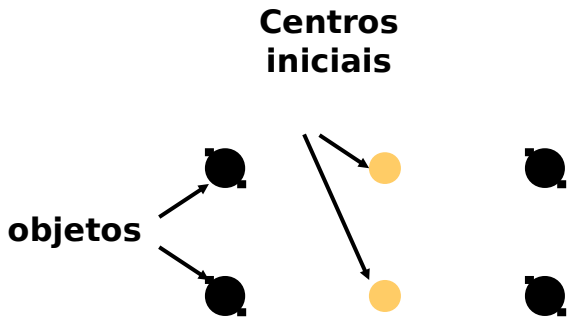
- Derivar $J$ com respeito a cada $\bar{\mathbf{x}}_c$ e igualar a zero:

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

$$\rightarrow\bar{\mathbf{x}}_c=\frac{\sum_{j=1}^{N}\mu_{cj}x_j}{\sum_{j=1}^{N}\mu_{cj}}$$

# Discussão

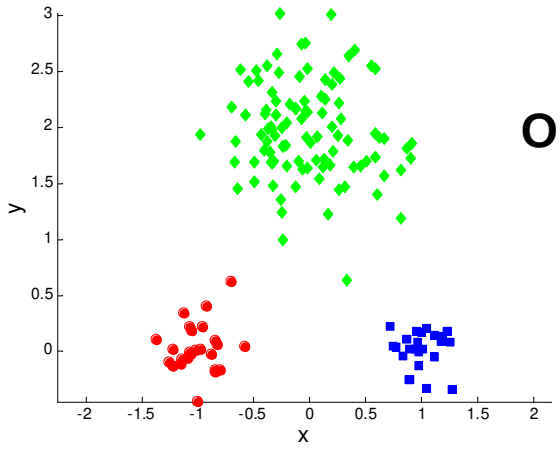
- Resultado pode variar significativamente dependendo da escolha das sementes (protótipos) iniciais
- k-means pode “ficar preso” em ótimos locais
  - Exemplo:



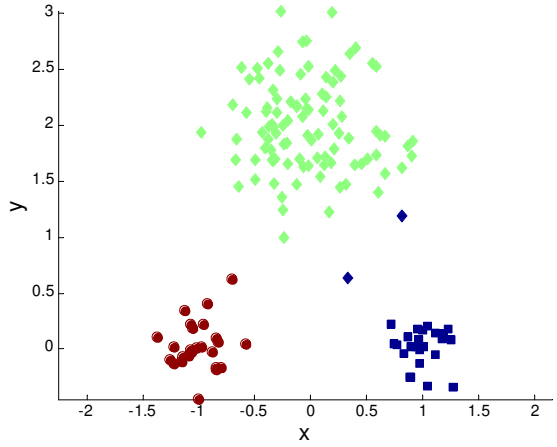

- Como evitar ... ?

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

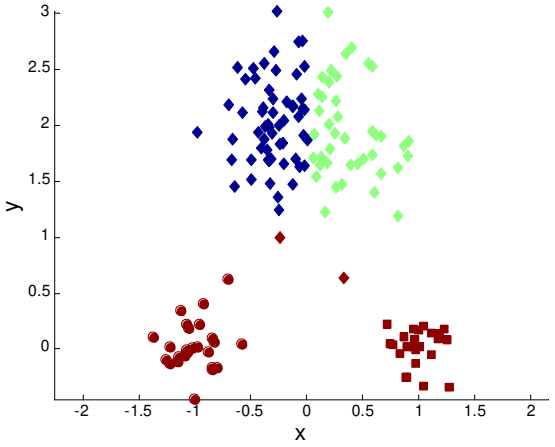
# Two different K-means Clusterings




**Original Points**

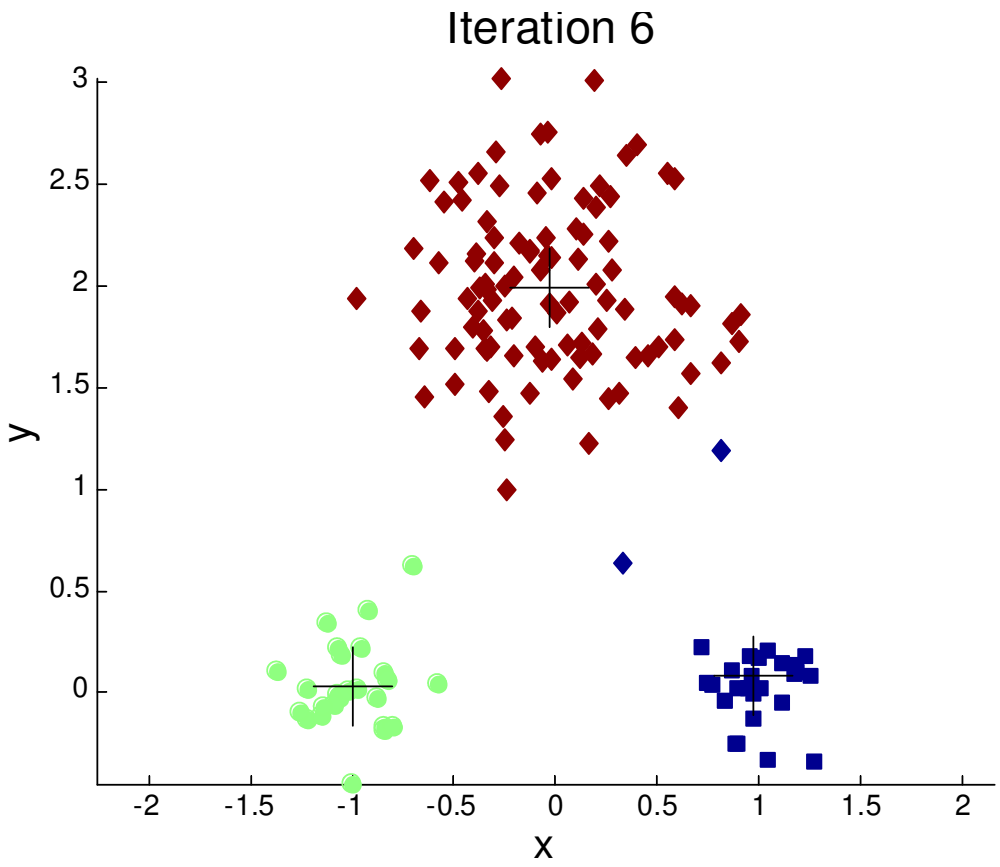



**Optimal Clustering**

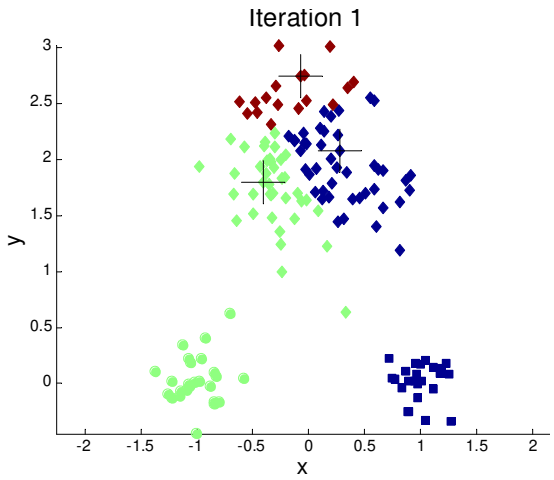


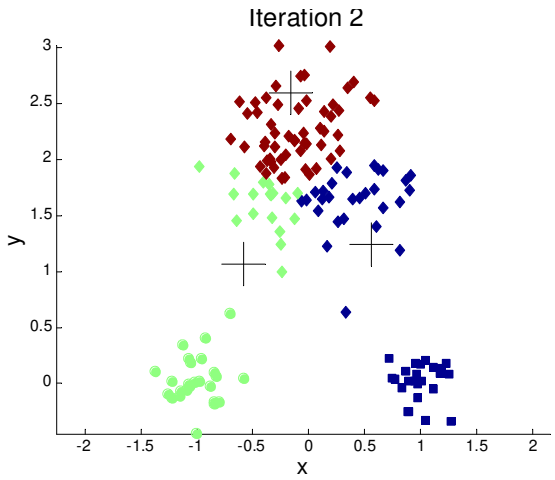

**Sub-optimal Clustering**

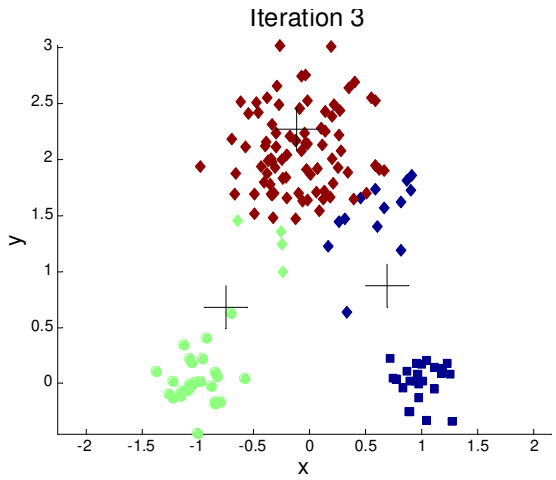
# Importance of Choosing Initial Centroids

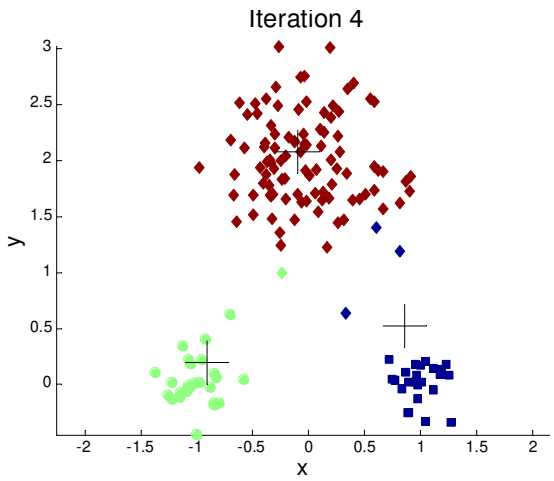


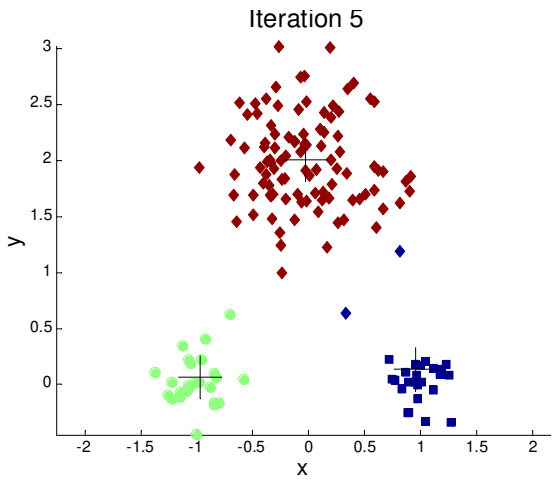
# Importance of Choosing Initial Centroids

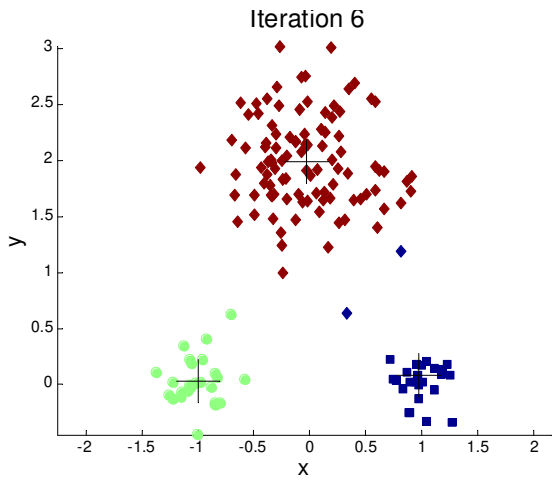


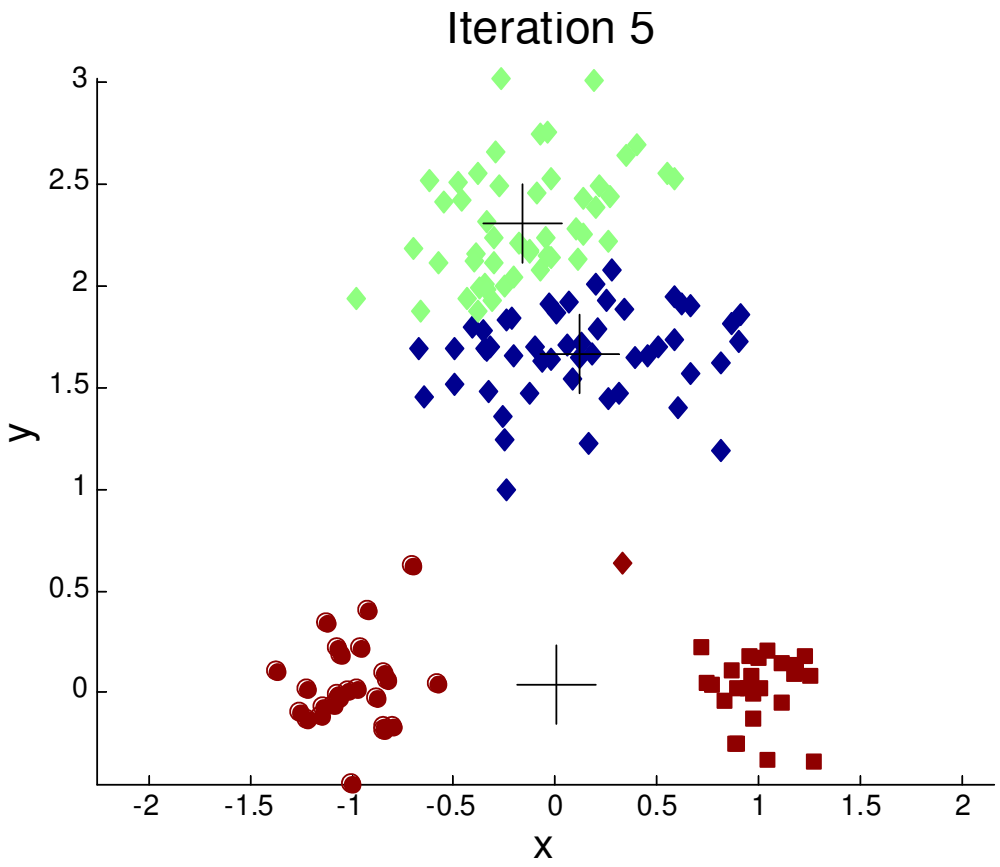



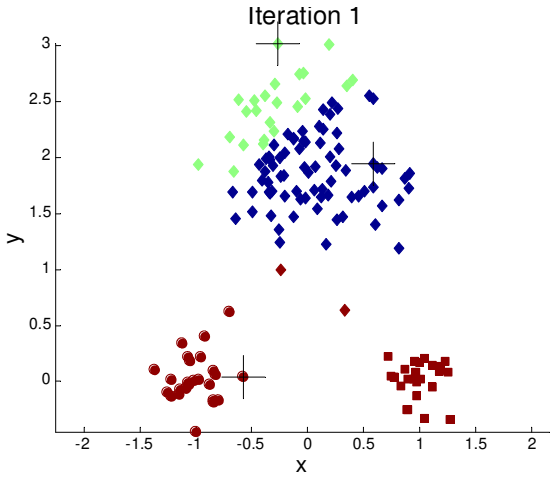



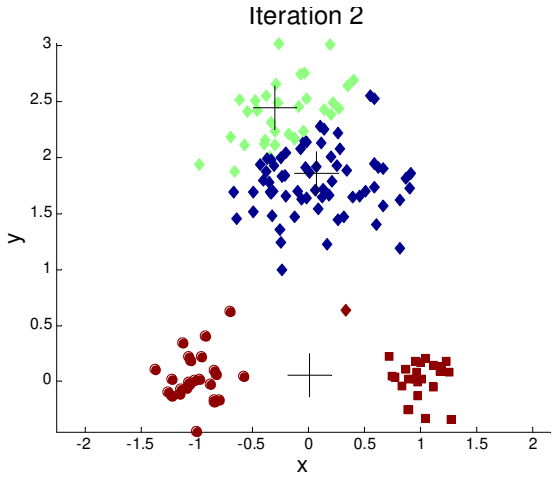



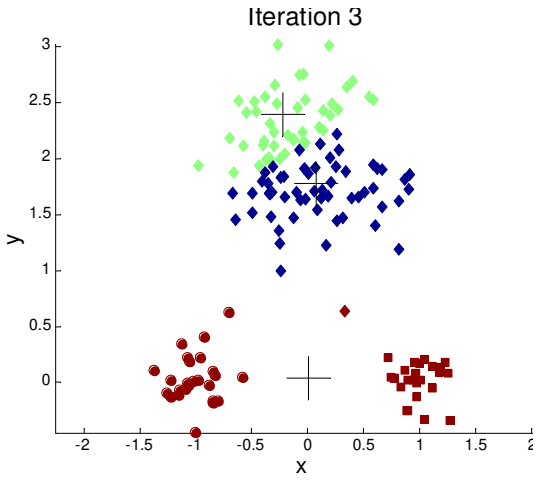
# Importance of Choosing Initial Centroids …

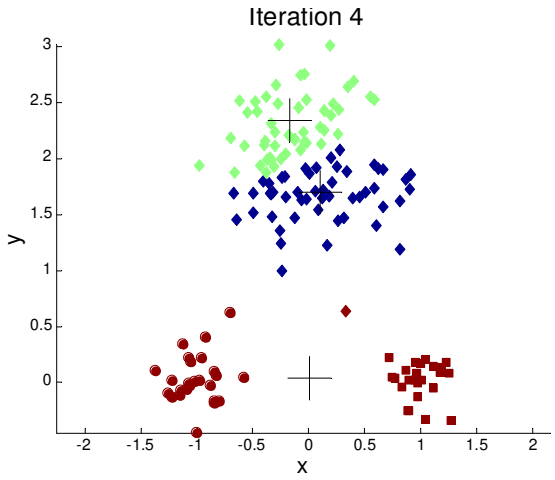


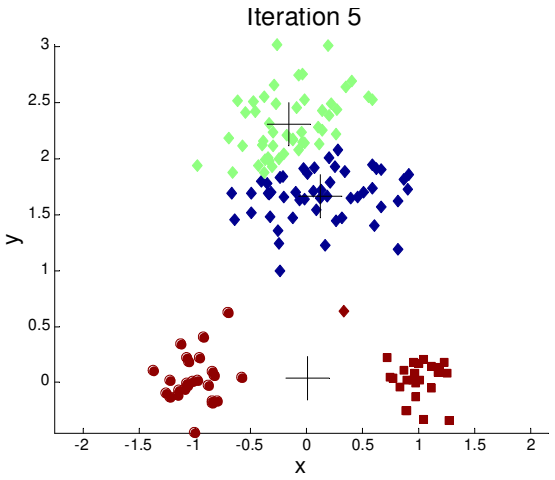
# Importance of Choosing Initial Centroids …



# Análise da Seleção dos Protótipos Iniciais

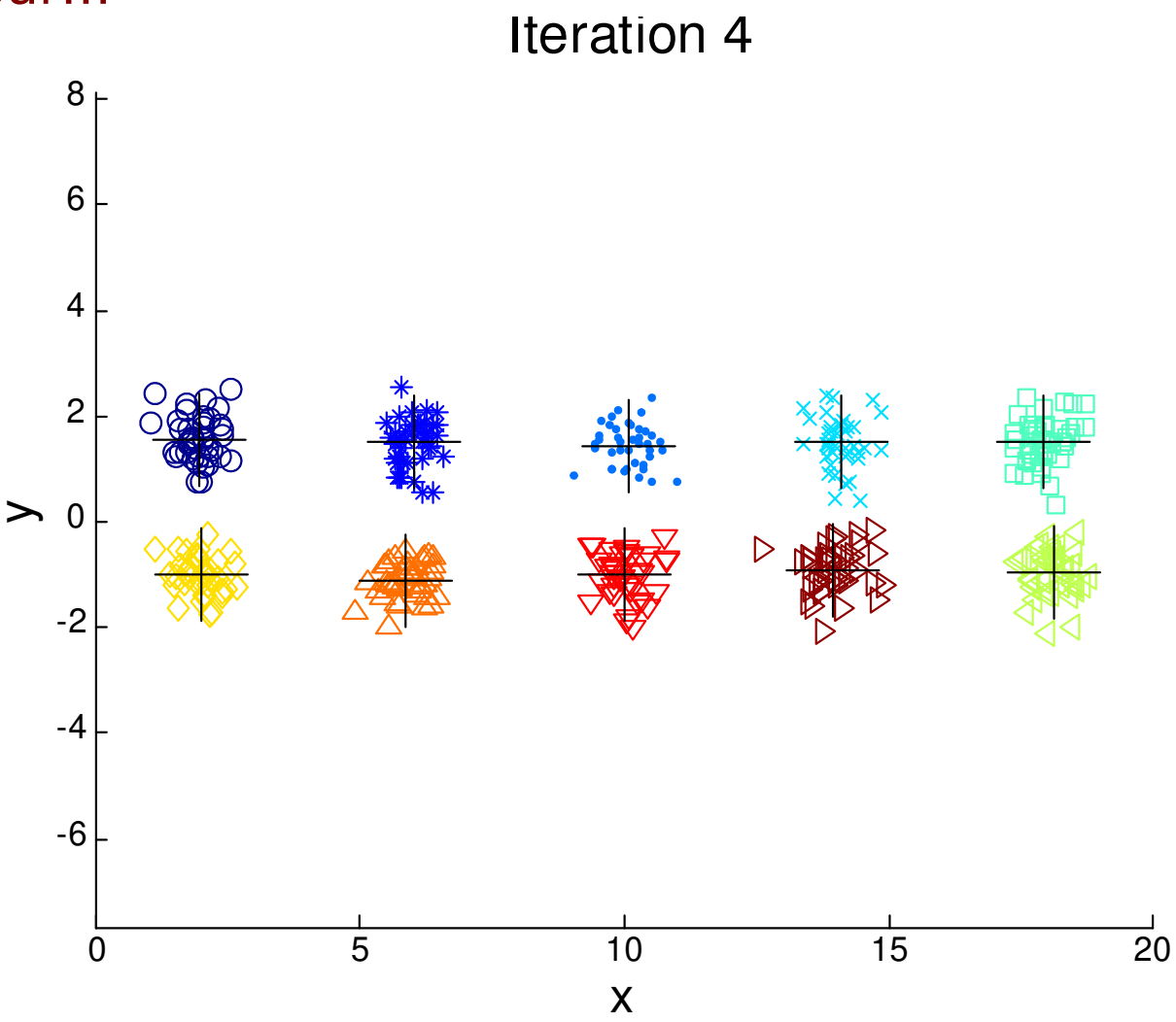
- **Premissa**: Uma boa seleção de k protótipos iniciais em uma base de dados com k grupos naturais é tal que cada protótipo é um objeto de um grupo diferente
- No entanto, a chance de se selecionar um protótipo de cada grupo é pequena, especialmente para k grande.
- Assumamos grupos balanceados, com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

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

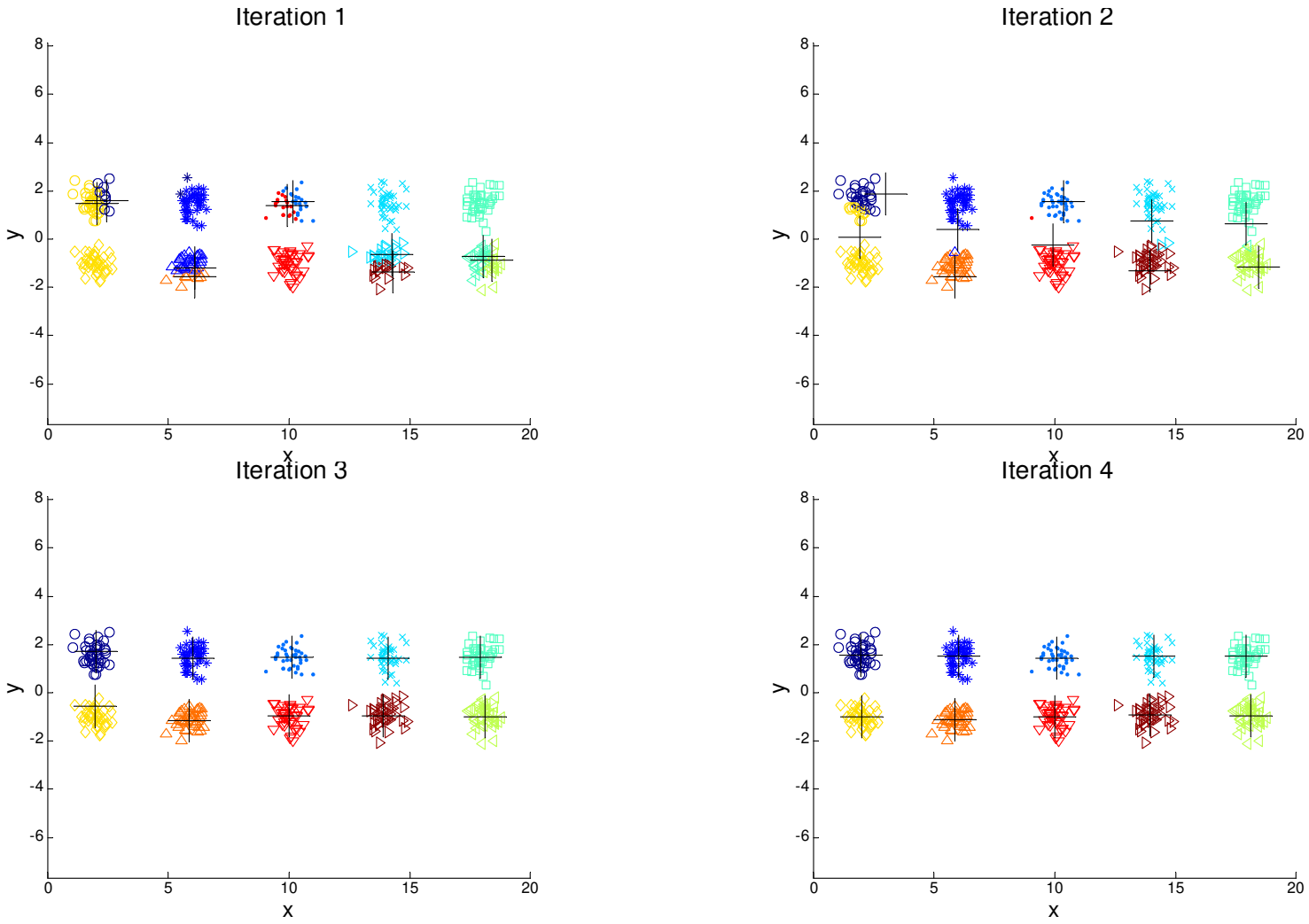
# Análise da Seleção dos Protótipos Iniciais

- Nº 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
- Nº 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 $k \cdot g \times (k - 1) \cdot g \times \ldots \times g = k! g^k$
- Portanto, tem-se $P = k! g^k / k^k g^k \rightarrow P = k! / k^k$
- Exemplo: se k = 10, P = 0.00036

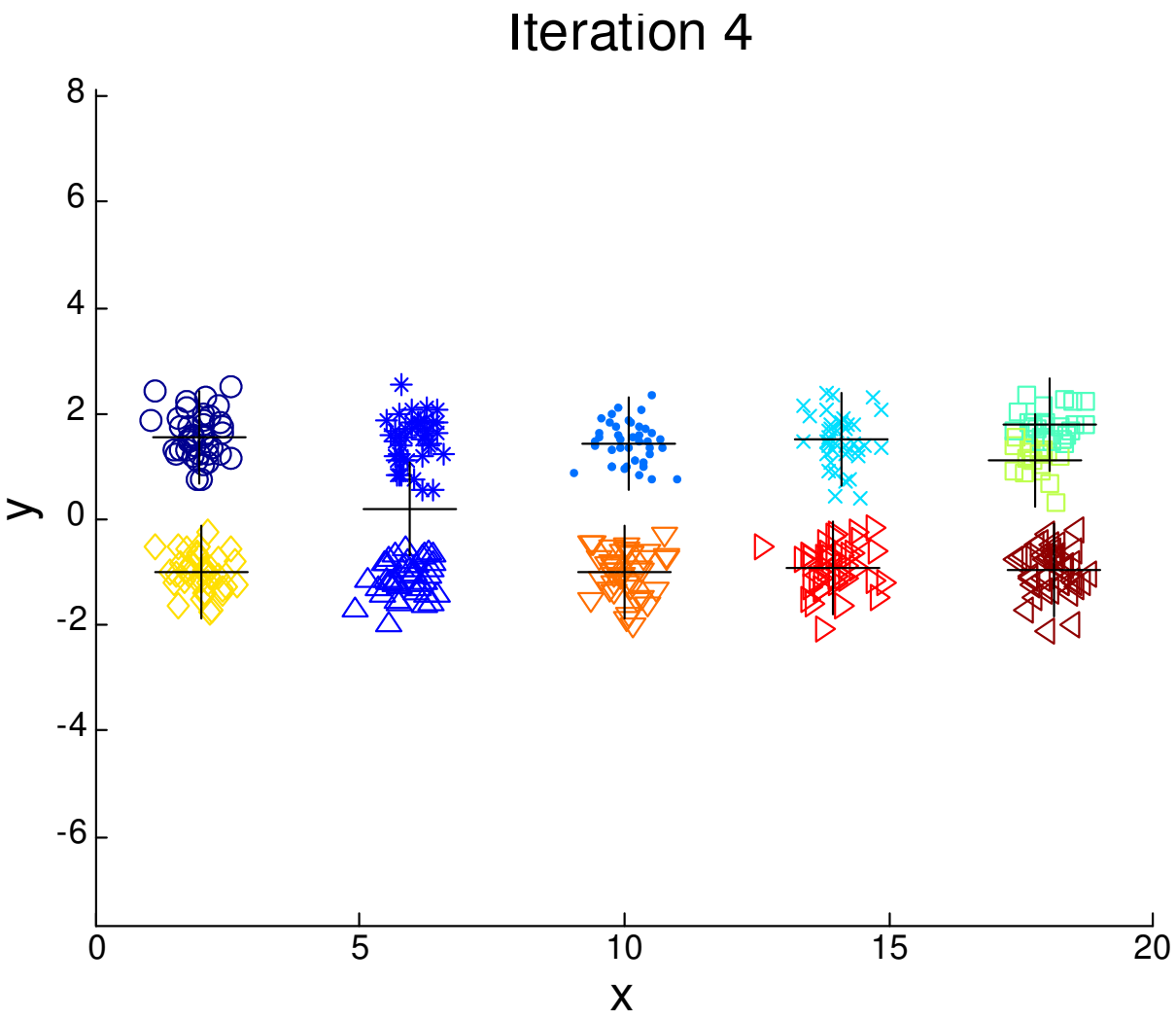
# Exemplo: Iniciando com 2 centróides iniciais em um grupo de cada par...



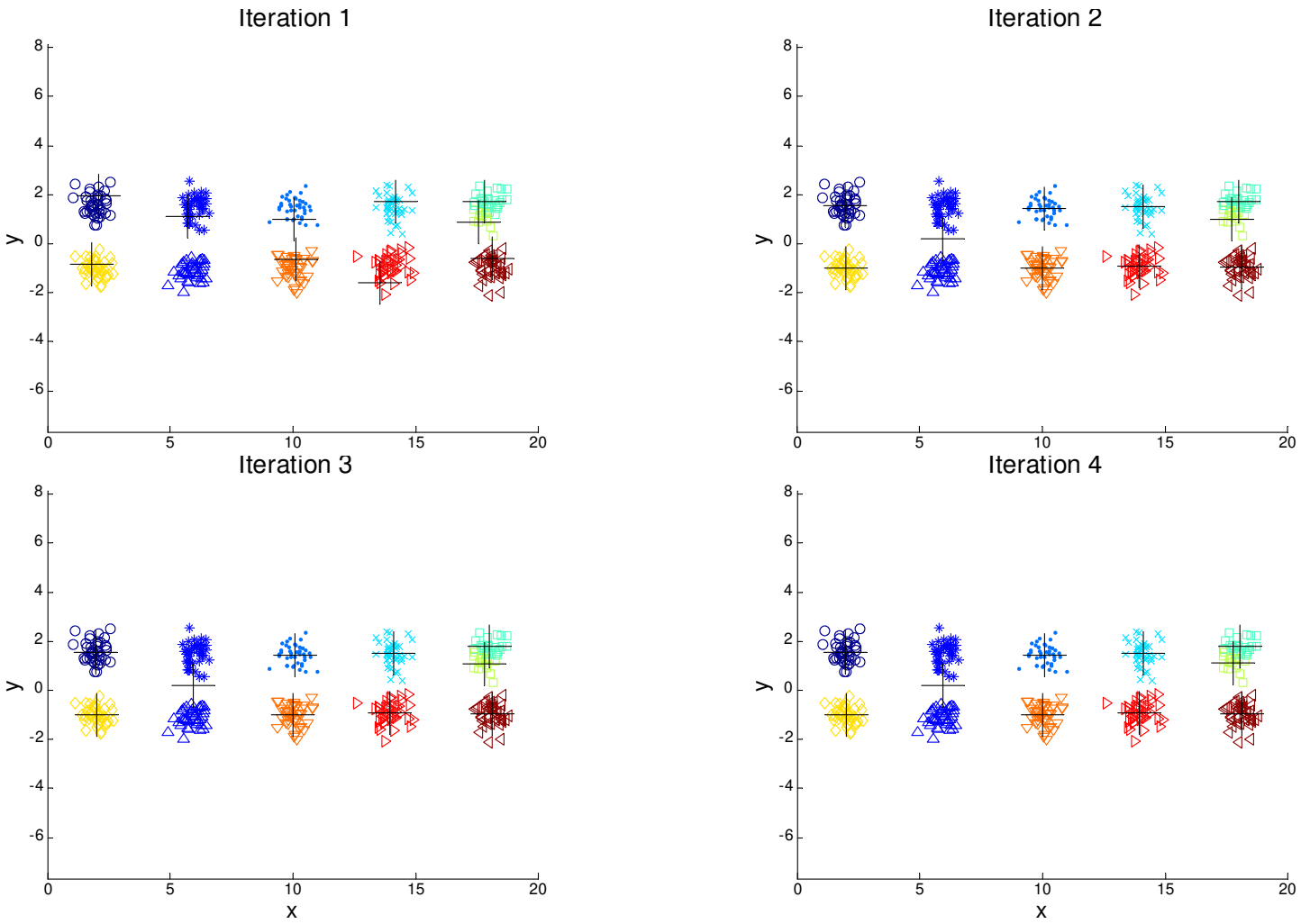
# Ilustrando Todas as Iterações:



# Agora Vejamos Outra Inicialização:



# Ilustrando Todas as Iterações:



# Alternativas para Inicialização

- Múltiplas Execuções (inicializações aleatórias):
  - funciona bem em muitos problemas.
  - mas em bases de dados complexas, pode demandar um no. enorme de execuções.
  - em particular para no. de grupos grande.
    - especialmente porque k é, em geral, desconhecido
- Agrupamento Hierárquico:
  - agrupa-se uma amostra dos dados
  - tomam-se os centros da partição com k grupos

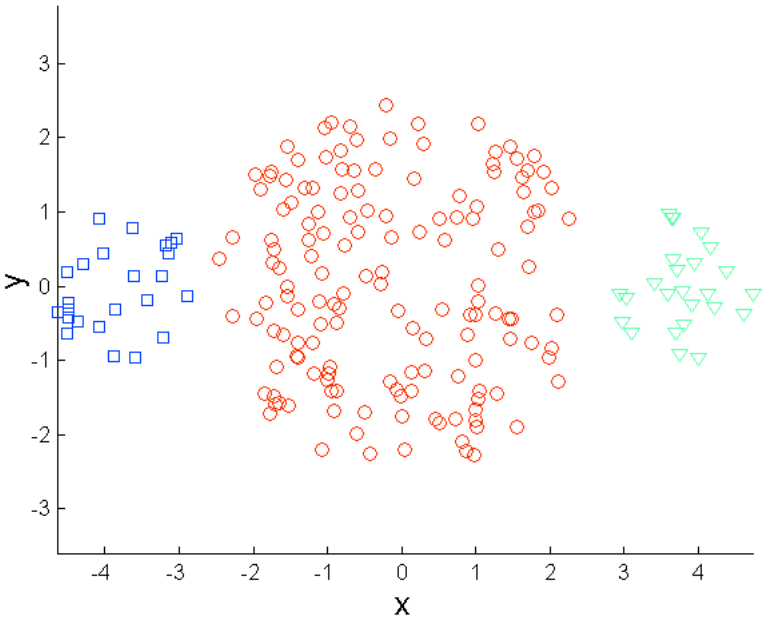
# Alternativas para Inicialização

- 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**, ...

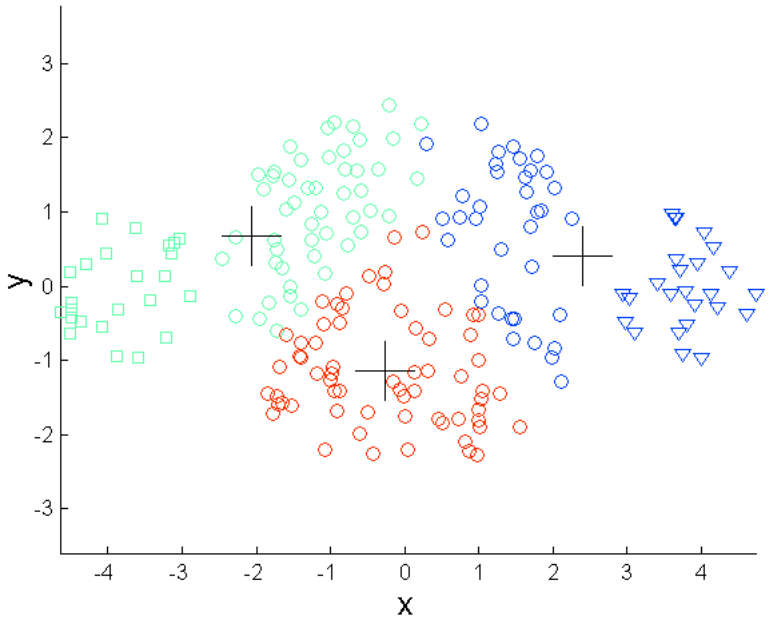
# Discussão

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

# Differing Sizes

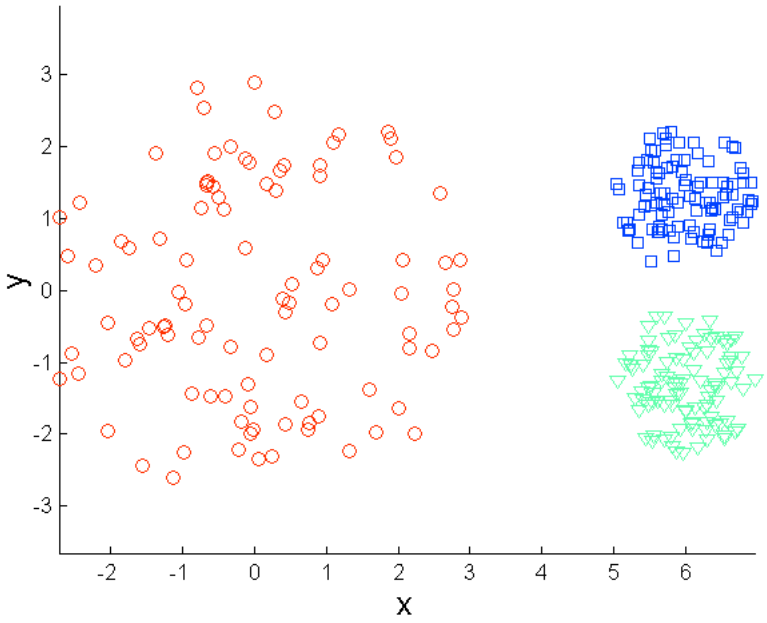



Original Points

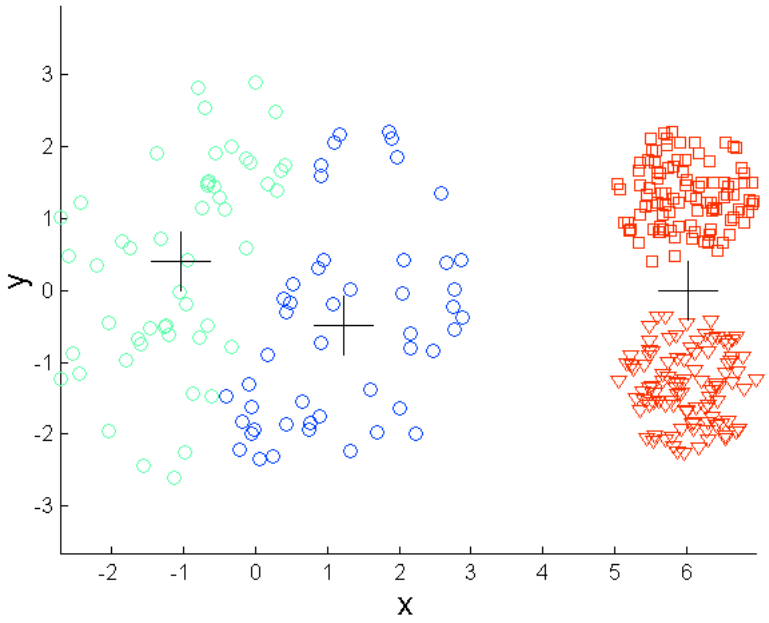



K-means (3 Clusters)

# Differing Density



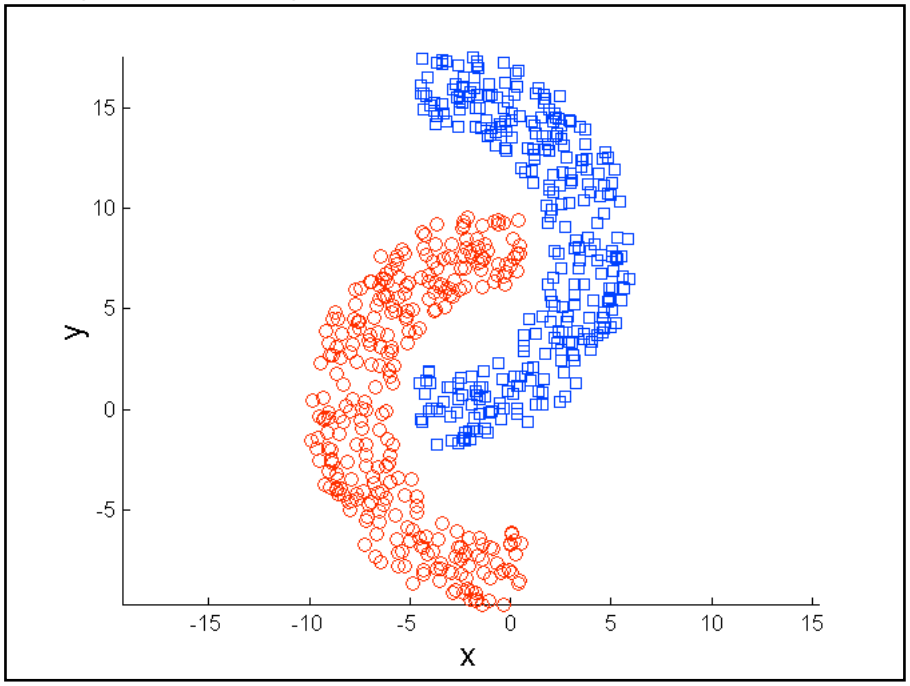

Original Points

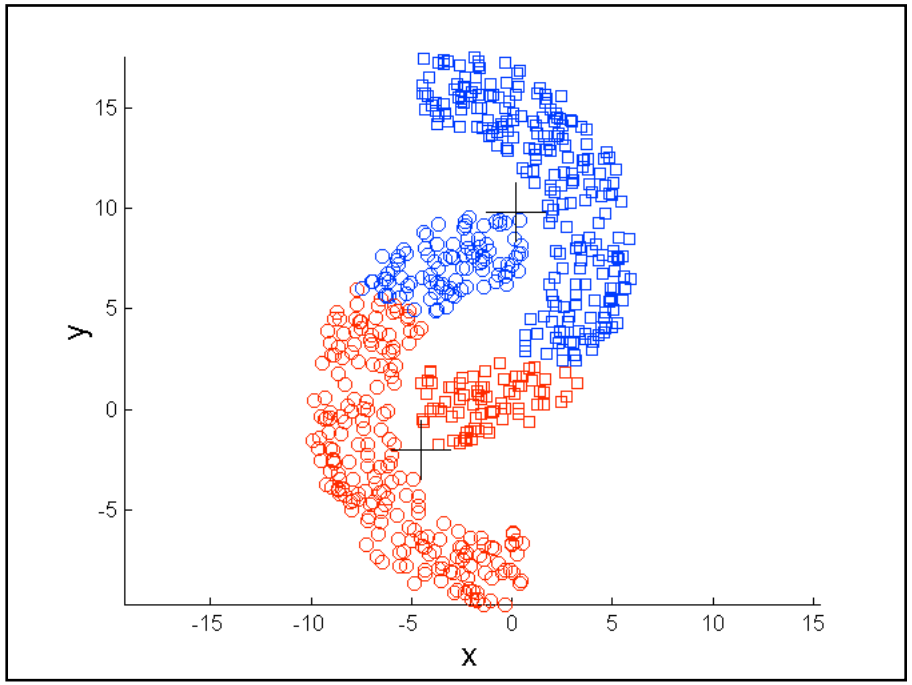



K-means (3 Clusters)

# Formas Não-Globulares

Tan, Steinbach, Kumar



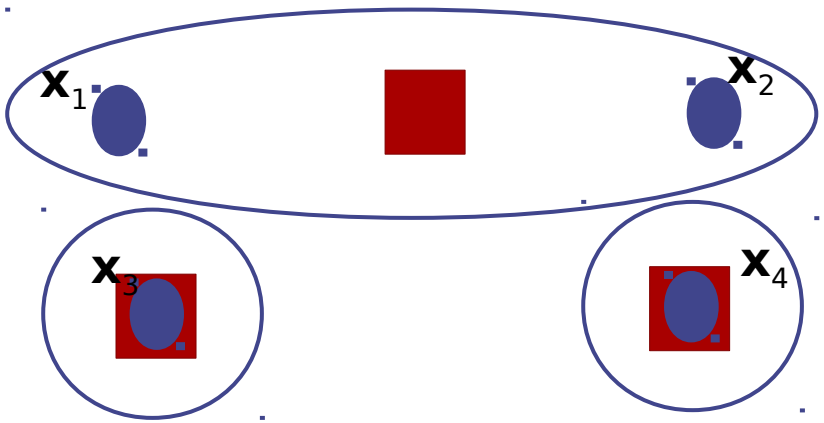




- **Nota**: na prática, esse problema em geral não é crítico, i.e., há pouco interesse na maioria das aplicações de mundo real

# Como tratar esses casos?

- O k-means identifica bem grupos que possuem o mesmo tamanho/densidade ou que estão bem separados
- Quando isso não ocorre, existe solução?
  - Podemos dividir os grupos em subgrupos menores
  - O conjunto desses subgrupos permitem amenizar as dificuldades

- O que acontecerá na próxima iteração?

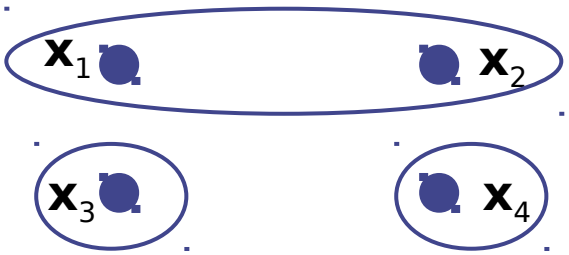



**Grupos iniciais**

***k* =3**

# Manipulando Grupos Vazios

- k-means pode gerar **grupos vazios**
  - Por inicialização em pontos “dominados” do espaço
    - protótipos não representativos: nenhum objeto mais próximo
    - inicialização como objetos ao invés de pontos aleatórios resolve
  - Pela inicialização de grupos
    - cujos protótipos são não representativos; por exemplo:




**Grupos iniciais**

***k* =3**

  - Ao longo das iterações

# Manipulando Grupos Vazios

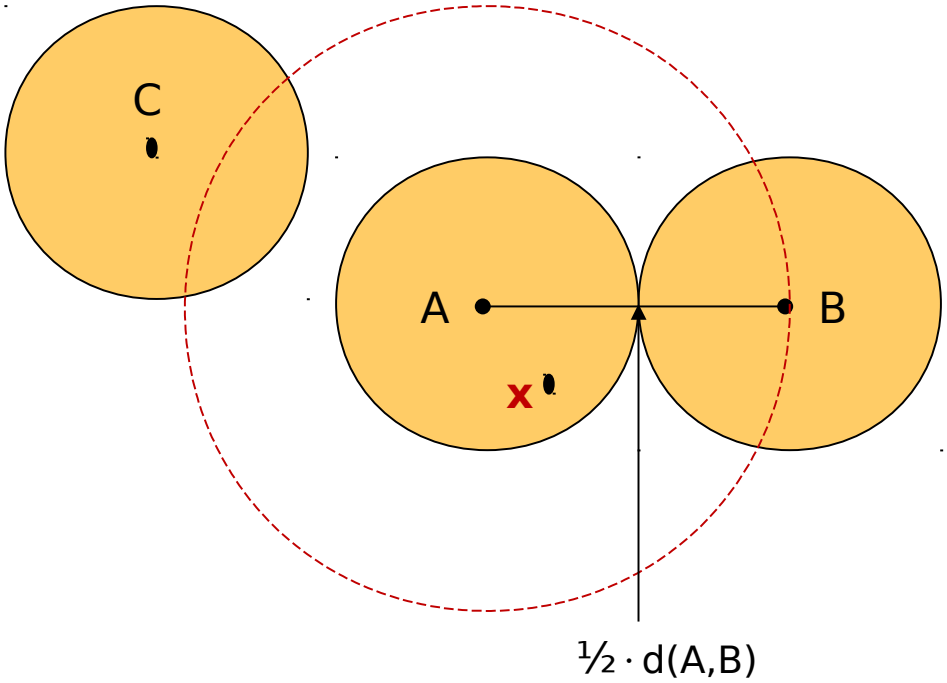
- Estratégias para contornar o problema:
  - Eliminar os protótipos não representativos (reduz k)
    - viável se o número inicial de grupos, k, puder ser reduzido
    - pode ser útil para ajustar valores superestimados de k
  - Substituir cada protótipo não representativo (mantém k)
    - pelo objeto que mais contribui para o SSE da partição
    - por um dos objetos do grupo com maior MSE
      - visa dividir o grupo com maior erro quadrático médio
    - **Nota**: a execução do algoritmo prossegue após a substituição

# Implementações Eficientes

- Desempenho computacional pode ser melhorado...
  - **Estruturas de Dados**, e.g.
    - **kd-trees** (seminários...)
  - **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** (vide discussão a seguir)
    - **Paralelização** (vide discussão a seguir)

# Uso da Desigualdade Triangular

- **Lema:** Se d(x,A)≤d(A,B)/2 então d(x,B)≥d(x,A).
  - **Prova:** d(A,B)≤d(A,x)+d(x,B); ...
- **Interpretação Geométrica**:




- G. Hamerly, Making k-means even faster, SIAM DM 2010.
- C. Elkan, Using the Triangle Inequality to Accelerate k-Means, ICML 2003.

# 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.

# Resumo do k-means

## Vantagens

- Simples e intuitivo
- Complexidade computacional **linear** em todas as variáveis críticas: O(N D k)
  - quadrático se D ≈ N ...
- 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 globulares
- Cada item deve pertencer a um único cluster (**partição rígida**, ou seja, sem sobreposição)
- Limitado a atributos numéricos
- Sensível a *outliers*

# Algumas Variantes do k-means

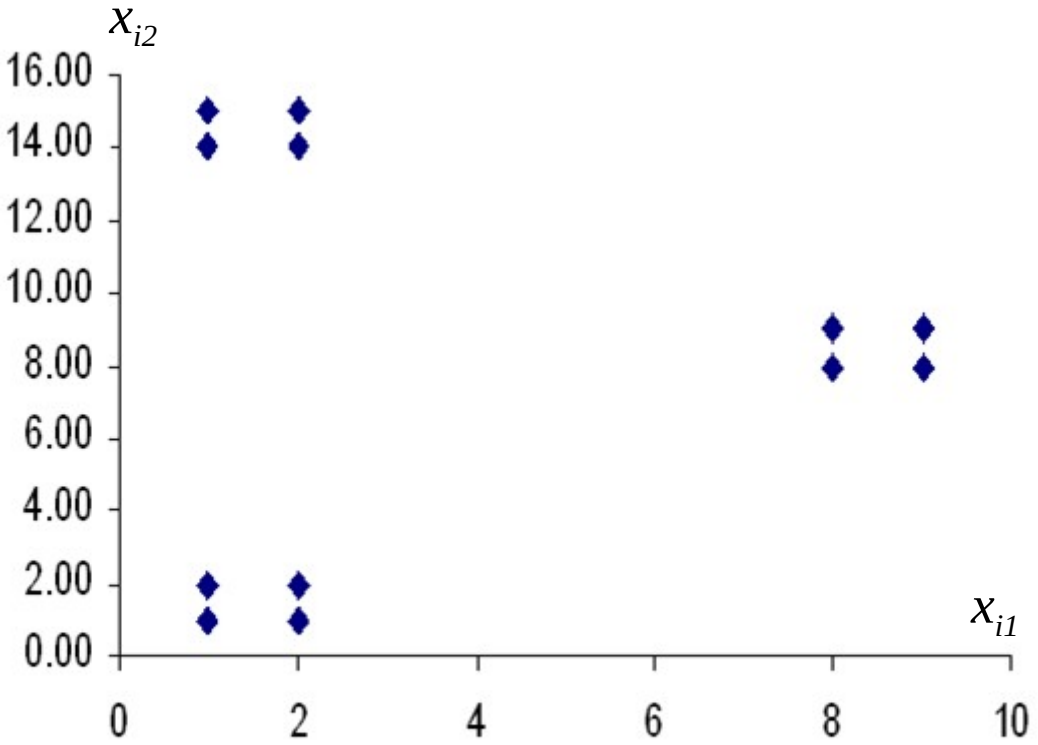
- **K-medianas:** Substituir as médias pelas medianas
  - Média de 1, 3, 5, 7, 9 é **5**
  - Média de 1, 3, 5, 7, 1009 é **205**
  - Mediana de 1, 3, 5, 7, 1009 é **5**
  - **Vantagem:** menos sensível a outliers
  - **Desvantagem:** implementação mais complexa
    - cálculo da mediana em cada atributo...
- 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 $\mathbf{x}_i$ | $x_{i1}$ | $x_{i2}$ |
|---|---|---|
| 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-medóides com k=3 nos dados acima, com medóides iniciais dados pelos objetos 5, 6 e 8

# 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 $k \in [k_{min}, k_{max}]$

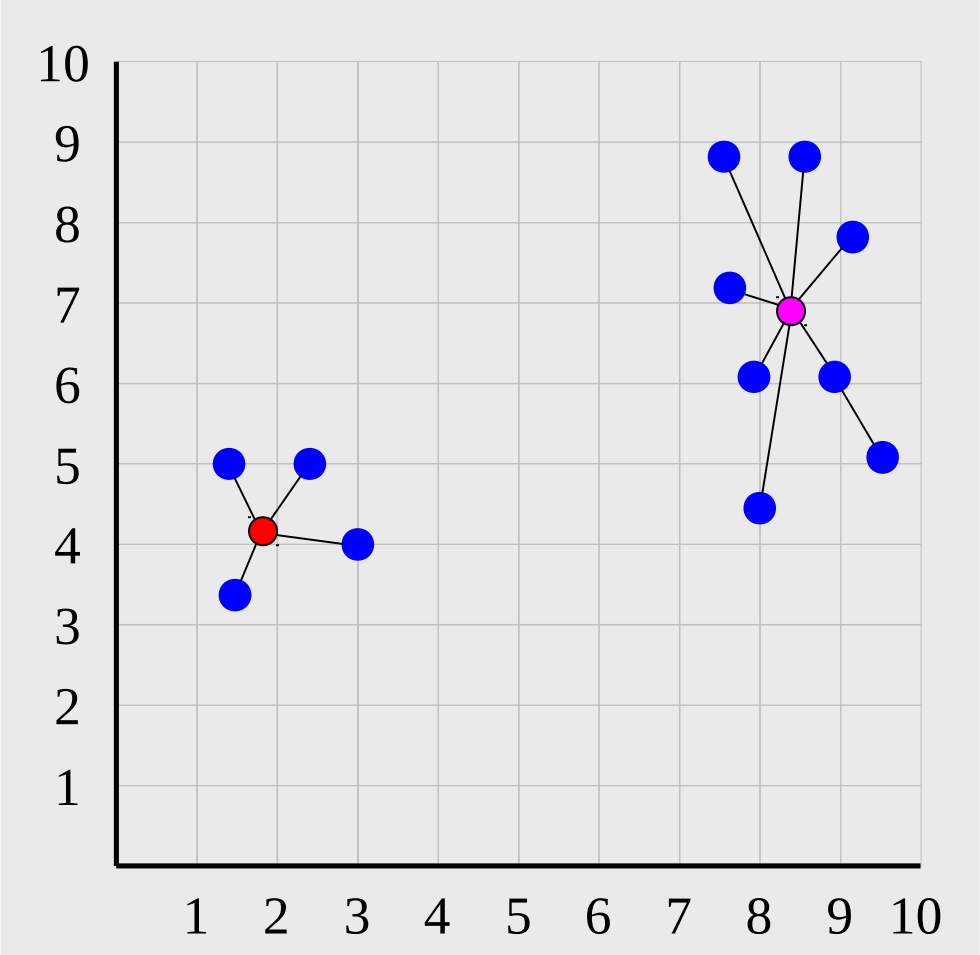
# Questão...

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

## Erro Quadrático:

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

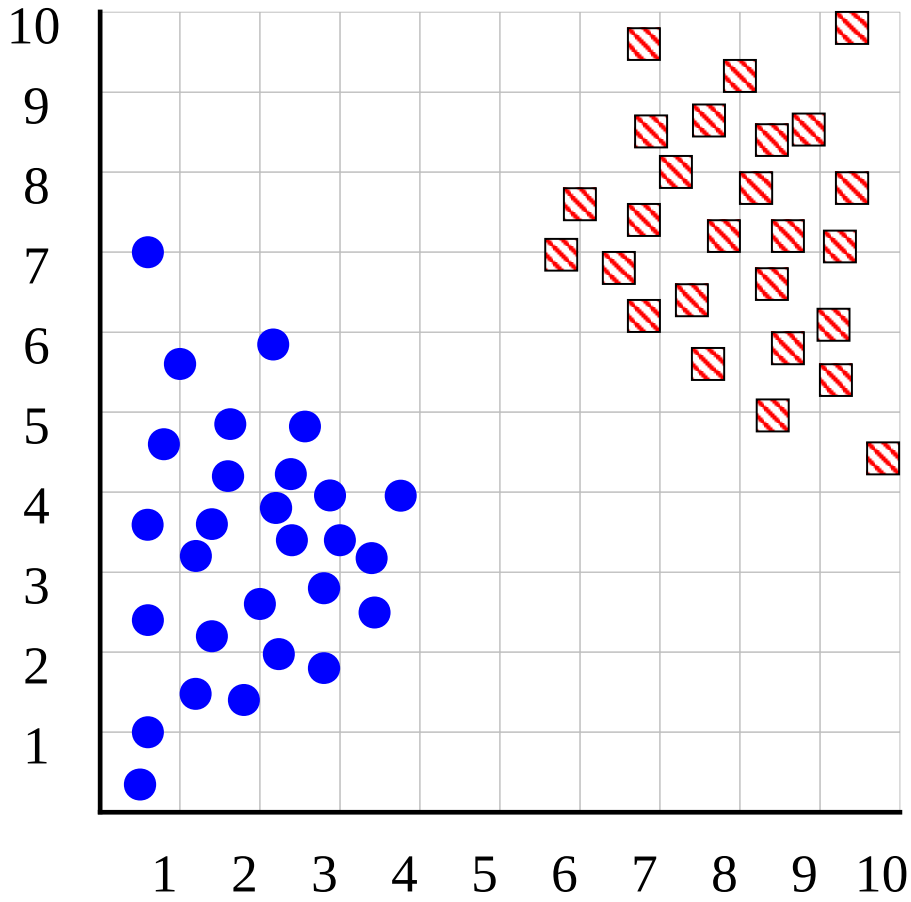
Função Objetivo




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

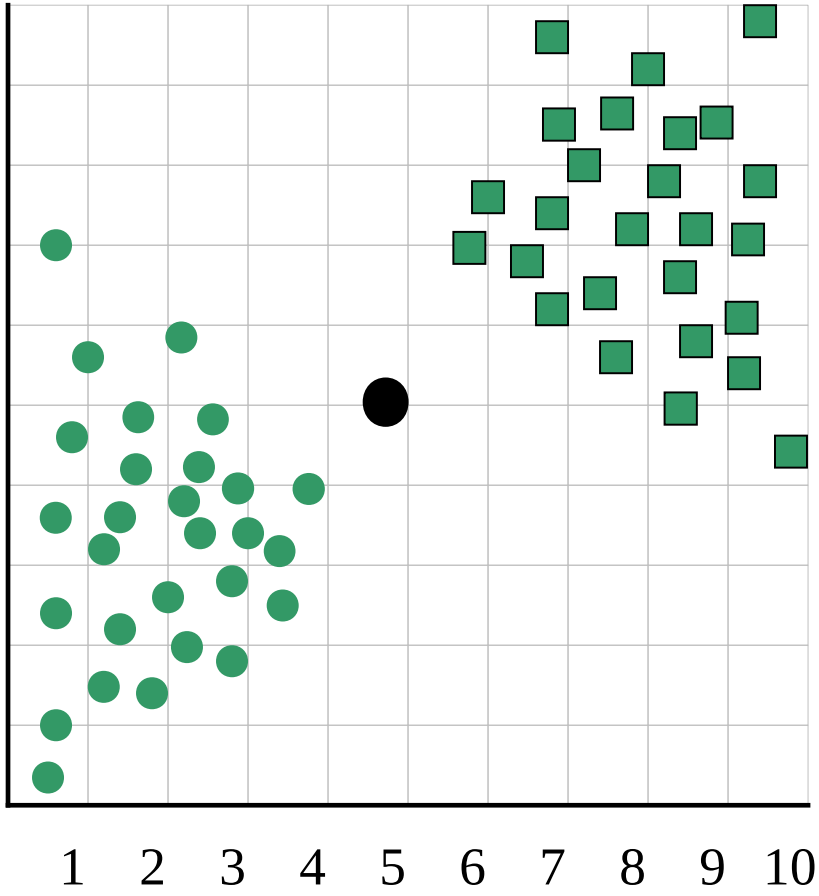
# Questão...

- Considere o seguinte exemplo:

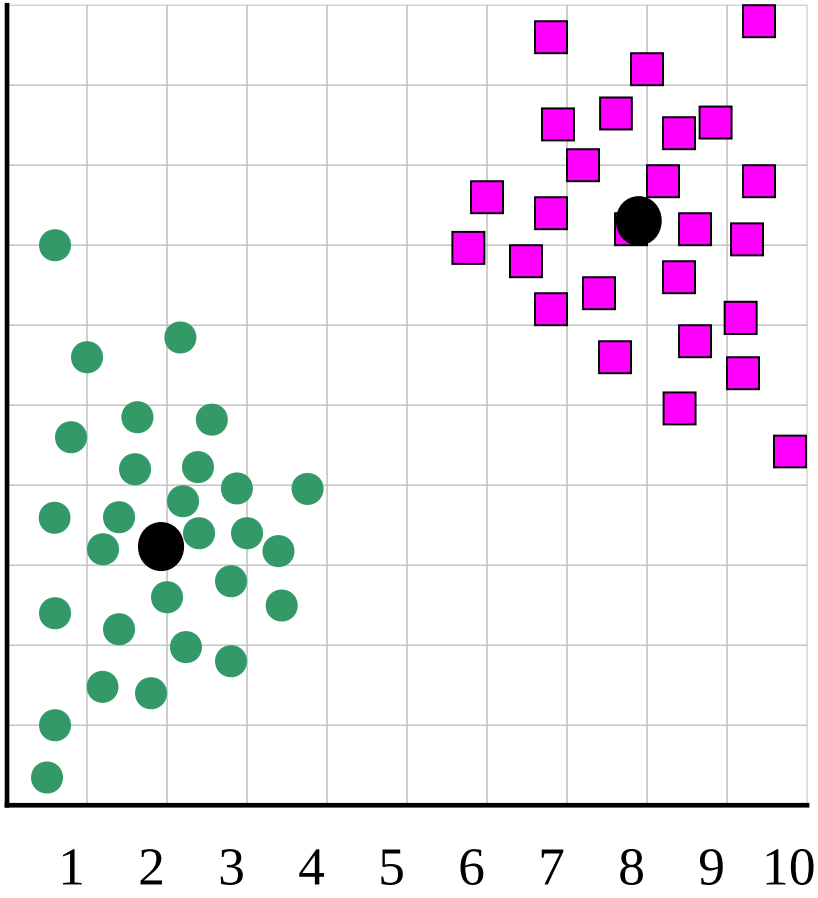



Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Manaus.

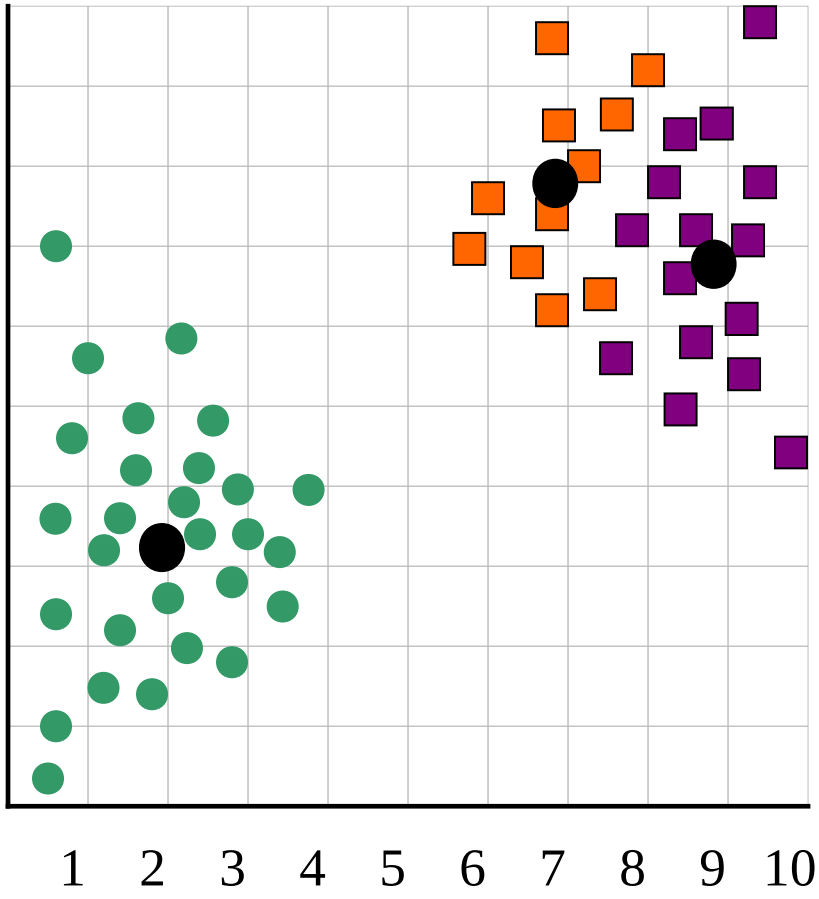
Para $k$ = 1, o valor da função objetivo é 873,0



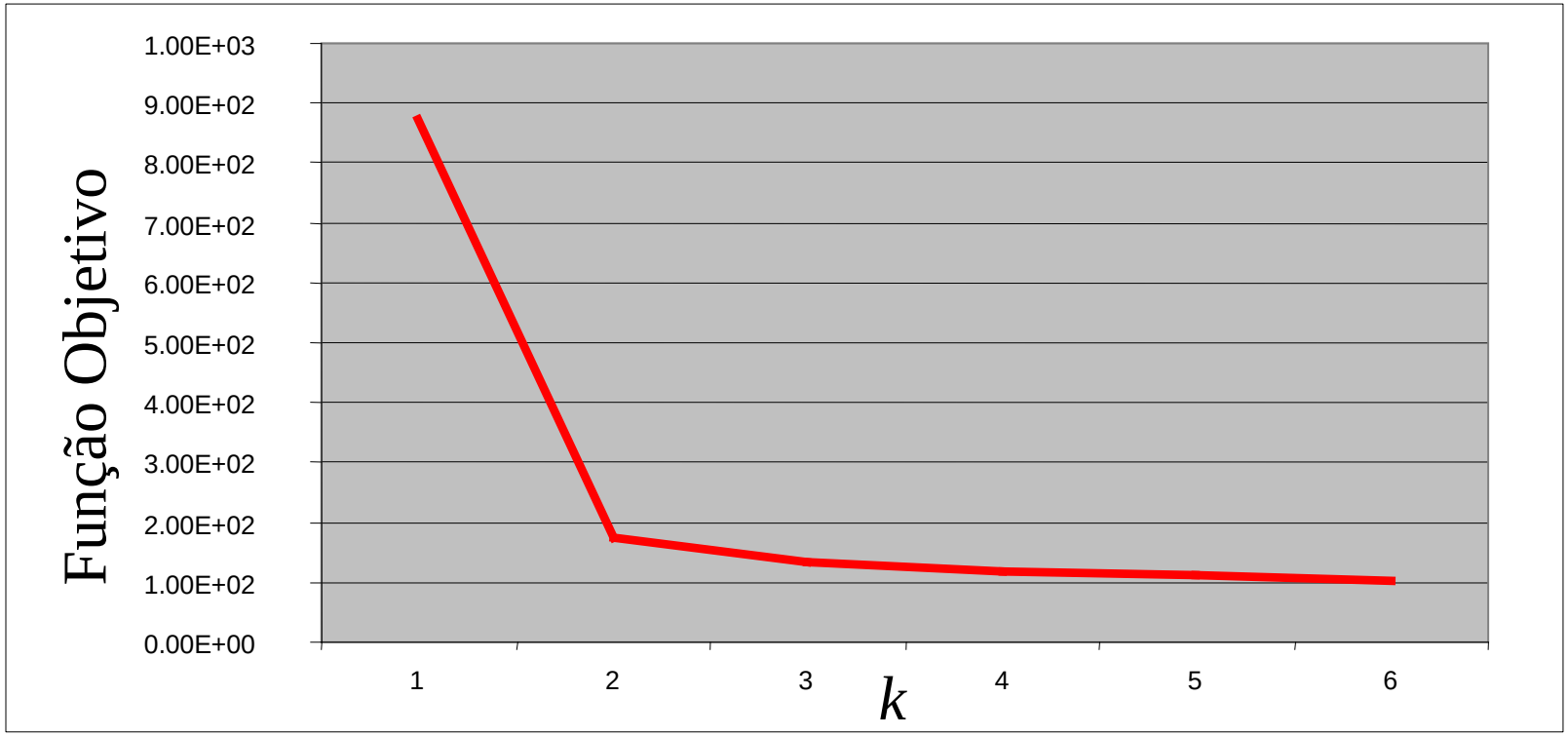
Para $k$ = 2, o valor da função objetivo é 173,1



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,…,6, ... e tentar identificar um “joelho” :

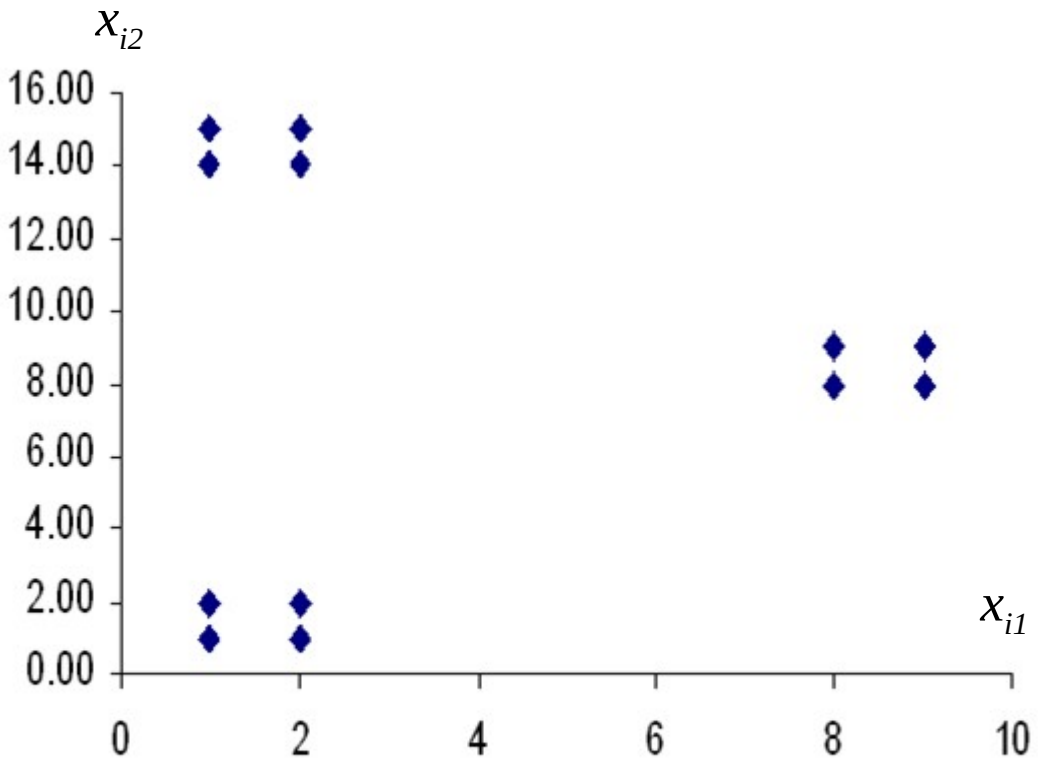





Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Manaus.

# Exercício

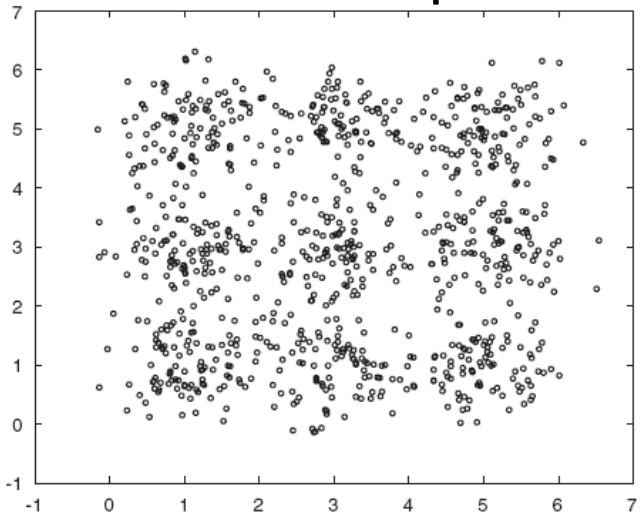
| Objeto $\mathbf{x}_i$ | $x_{i1}$ | $x_{i2}$ |
|---|---|---|
| 1 | 1 | 2 |
| 2 | 2 | 1 |
| 3 | 1 | 1 |
| 4 | 2 | 2 |
| **5** | **8** | **9** |
| **6** | **9** | **8** |
| **7** | **9** | **9** |
| **8** | **8** | **8** |
| 9 | 1 | 15 |
| 10 | 2 | 15 |
| 11 | 1 | 14 |
| 12 | 2 | 14 |

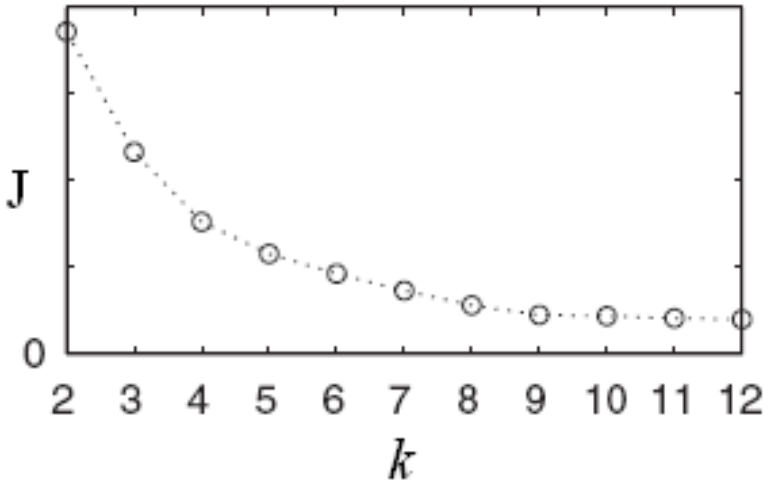



- Executar k-means com k=2 até k=5 nos dados acima e representar graficamente a f. objetivo J em função de k

# Questão...

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

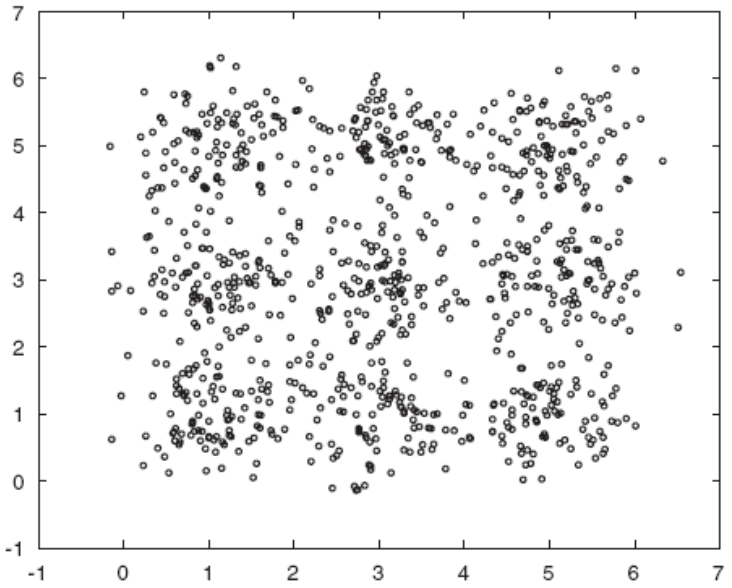



- Além disso, como utilizar essa metodologia em variantes baseadas em busca guiada, que otimizam k ?
  - X-means, k-means evolutivo, ...
- Solução: **critérios de validação de agrupamento**.

# Algoritmos de Partição Com Sobreposição

- Algoritmos particionais como o k-means, k-medóides e diversos outros produzem uma **partição rígida** da base de dados:
  - Cada objeto pertence a um único grupo, de forma integral
  - Usualmente refere-se a esse tipo de partição como **Hard** ou **Crisp**
- No entanto, muitos problemas envolvem grupos mal delineados, que não podem ser separados adequadamente dessa maneira
- Em outras palavras, existem situações nas quais os dados compreendem categorias que se sobrepõem umas às outras em diferentes níveis
- Por exemplo:



# Partições Probabilísticas

**Matriz de Partição Probabilística**: elementos assumem valores contínuos de pertinência que representam probabilidades, ao invés de binários, i.e., $\mu_{ij} \in [0,1]$, $\sum_i(\mu_{ij}) = 1 \ \forall j$

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

# Expectation Maximization (EM)

- 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**

# EM – Mistura de Gaussianas

- Representada pela seguinte *p.d.f* :

$$p(\mathbf{x}) = \sum_{i=1}^{k} \pi_i \mathrm{N}\left(\mathbf{x} \middle| \mathbf{v}_i, \boldsymbol{\Sigma}_i\right)$$

  - **x** é um objeto
  - N é uma Gaussiana (da mesma dimensão dos objetos)
    - $\mathbf{v}_i$ é o centro da i-ésima Gaussiana (vetor da mesma dimensão de **x**)
    - $\boldsymbol{\Sigma}_i$ é a matriz de covariância da i-ésima Gaussiana
  - $\pi_i$ são os coeficientes da mistura
  - *k* é o número de Gaussianas/componentes da mistura

# EM – Mistura de Gaussianas

- Para compreender $p(\mathbf{x})$, seja uma var. aleatória binária k-dimensional **z**, tal que:
  - **z** = $[z_1 \ldots z_k]^T$ assume apenas valores em representação 1-de-k:
    - $z_i = 1$ para um dado i $\in$ {1, ..., k}; todos os demais são nulos, ou seja: $z_i \in \{0, 1\}$ e $\sum_i z_i = 1$
  - Define-se $\pi_i = p(z_i = 1)$ como a **probabilidade a priori** de que $z_i = 1$, $0 \leq \pi_i \leq 1$, $\sum_i \pi_i = 1$.
  - A distribuição de probabilidades $p(\mathbf{z})$ é tal que:

$$p(\mathbf{z}) = \prod_{i=1}^{k} \pi_i^{z_i}$$

# EM – Mistura de Gaussianas

- Note que a i-ésima Gaussiana corresponde à distribuição condicional de **x** dado um valor particular de **z**, i.e.:
  - $p(\mathbf{x} \mid z_i = 1) = \mathrm{N}(\mathbf{x} \mid \mathbf{v}_i, \boldsymbol{\Sigma}_i)$, ou (equivalentemente)
  - $p(\mathbf{x} \mid \mathbf{z}) = \mathrm{N}(\mathbf{x} \mid \mathbf{v}_i, \boldsymbol{\Sigma}_i)$ para a realização de $\mathbf{z}$ tal que $z_i = 1$
- Das distribuições $p(\mathbf{z})$ e $p(\mathbf{x} \mid \mathbf{z})$ temos:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})\, p(\mathbf{x} \mid \mathbf{z})$$

- A distribuição $p(\mathbf{x})$ é obtida então como:

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{k} \pi_i \mathrm{N}\left(\mathbf{x} \middle| \mathbf{v}_i, \boldsymbol{\Sigma}_i\right)$$

# EM – Mistura de Gaussianas

- Responsabilidade do componente *i* para explicar $\mathbf{x}_j$ :

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

$( p(\mathbf{z}_j \mid \mathbf{x}_j) = p(\mathbf{x}_j, \mathbf{z}_j) / p(\mathbf{x}_j) \rightarrow p(\mathbf{z}_j \mid \mathbf{x}_j) = p(\mathbf{z}_j)\ p(\mathbf{x}_j \mid \mathbf{z}_j) / p(\mathbf{x}_j) \rightarrow p(z_{ij} = 1 \mid \mathbf{x}_j) = p(z_{ij} = 1)\ p(\mathbf{x}_j \mid z_{ij} = 1) \ / \ p(\mathbf{x}_j)\ )$

- É a probabilidade **a posteriori** de $z_i = 1$ dado que se observou $\mathbf{x}_j$
  - Compare com $\pi_i$ (probabilidade a priori)
- Probabilidade a posteriori de $\mathbf{x}_j$ ter sido produzido pela i-ésima Gaussiana

# EM – Mistura de Gaussianas

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

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

  - $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_k\}$, $\mathbf{v} = \{\mathbf{v}_1, ..., \mathbf{v}_k\}$ e $\boldsymbol{\pi} = \{\pi_1, ..., \pi_k\}$
  - Refere-se a esta distribuição por $p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\Sigma}, \mathbf{v})$

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

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

# EM – Mistura de Gaussianas

- 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 $\mu_{ij}$ (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 $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_k\}$, $\mathbf{v} = \{\mathbf{v}_1, ..., \mathbf{v}_k\}$ e $\boldsymbol{\pi} = \{\pi_1, ..., \pi_k\}$
  - **Passo M** (Maximization)
    - Ajusta os parâmetros do modelo visando maximizar a log-verossimilhança

# EM – Mistura de Gaussianas

- **Passo E** (Expectation):
  - Avalia as probabilidades a posteriori $\mu_{ij}$ (i = 1, ..., k; j = 1, ..., N)

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

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

# EM – Mistura de Gaussianas

- **Passo M** (Maximization):
  - Ajusta o modelo visando maximizar a verossimilhança

$$\begin{cases} \mathbf{v}_i = \dfrac{1}{N_i} \displaystyle\sum_{j=1}^{N} \mu_{ij} \mathbf{x}_j \\ \boldsymbol{\Sigma}_i = \dfrac{1}{N_i} \displaystyle\sum_{j=1}^{N} \mu_{ij} (\mathbf{x}_j - \mathbf{v}_i)(\mathbf{x}_j - \mathbf{v}_i)^T \\ \pi_i = \dfrac{N_i}{N} \end{cases}$$

$$N_i = \sum_{j=1}^{N} \mu_{ij}$$

centróide ponderado

covariância ponderada

Coeficientes da mistura / prob. a priori do i-ésimo componente

Número efetivo de pontos atribuídos ao i-ésimo grupo

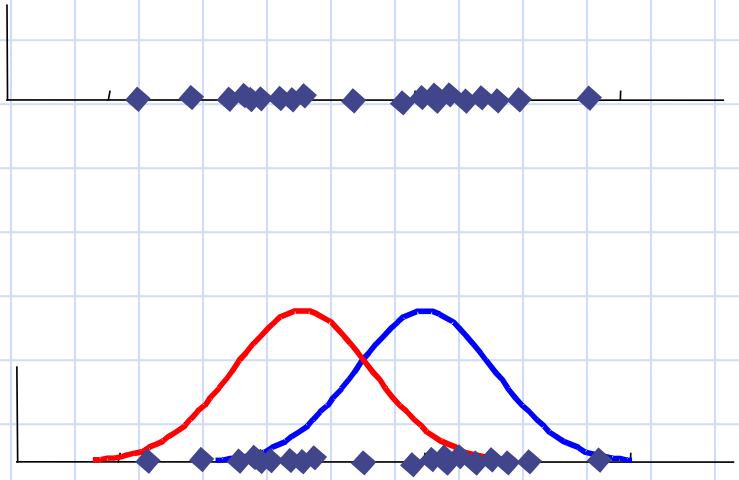
# EM – Mistura de Gaussianas

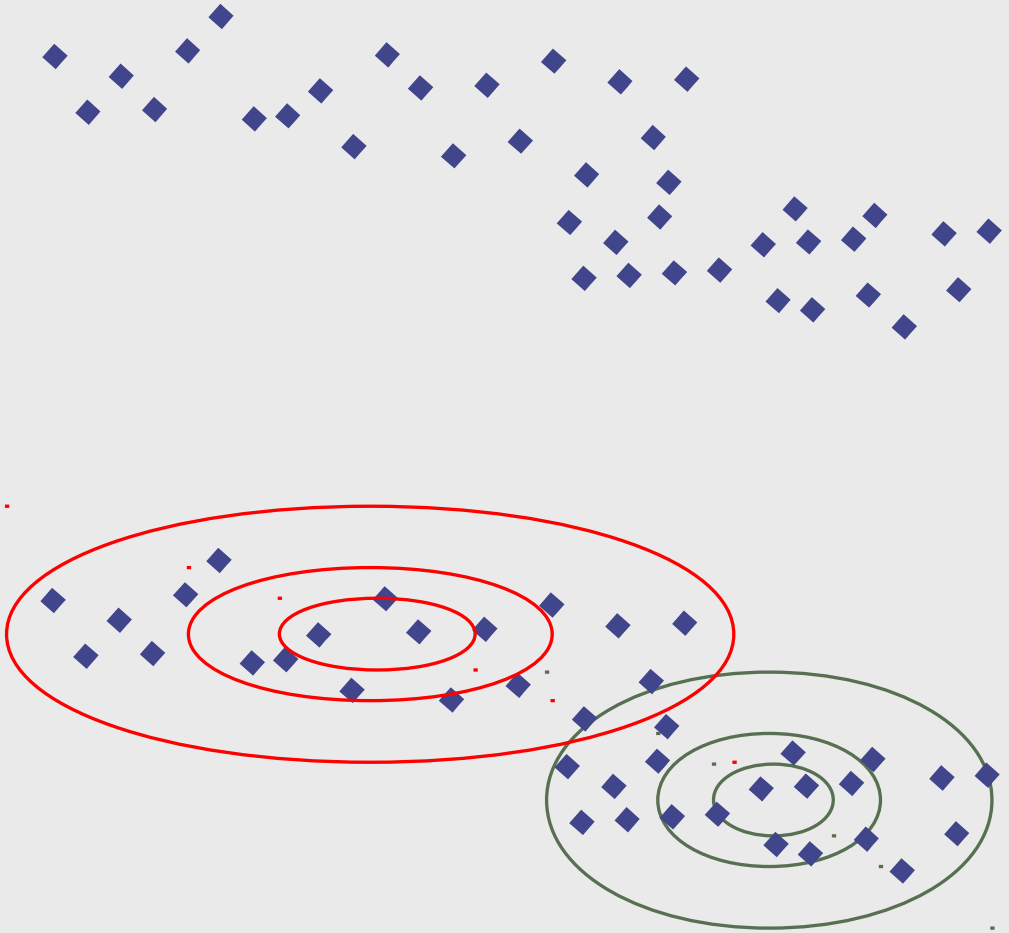
- **Algoritmo**:
  1. **Inicialização** (via k-means)
     - protótipos $\mathbf{v}_i$ = centróides finais do k-means
     - covariâncias $\Sigma_i$ = matrizes de covariância finais (dos grupos)
     - probabilidades $\mu_{ij}$ (para $N_i$ e $\pi_i$) = matriz de partição rígida final
  2. **Passo E**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

# EM × k-Means

- EM produz informação muito mais rica sobre os dados
  - Probabilidades associadas a cada padrão / cluster
- Probabilidades produzidas por EM podem facilmente ser convertidas em uma partição rígida, caso desejado
  - Via projeção do maior valor
  - 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 N demanda as inversas das matrizes de covariância $\Sigma_i$, que é $O(n^3)$
- k-means é um caso particular de EM. Ambos estão sujeitos a mínimos locais

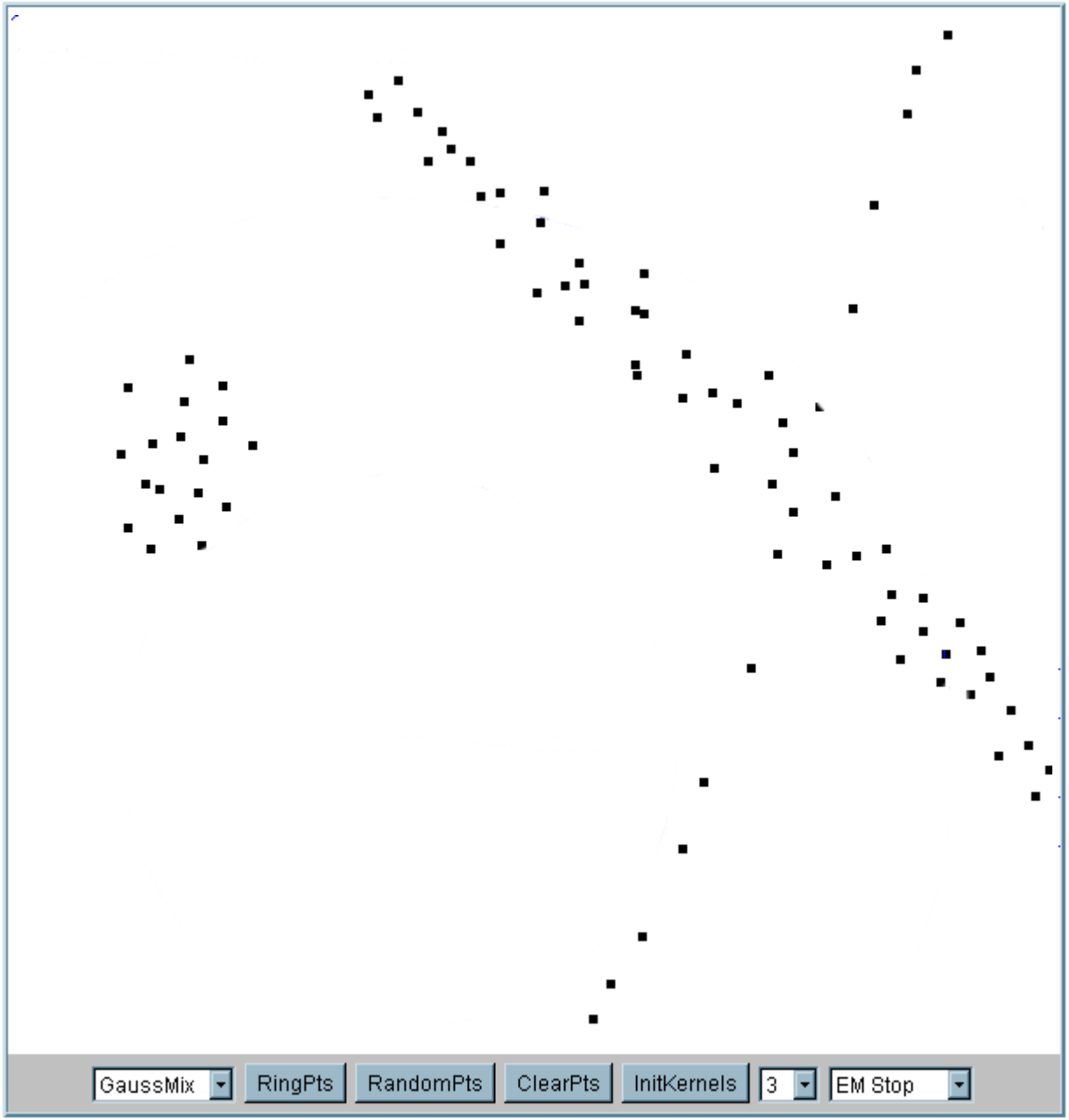
# **EM** (Mistura de Gaussianas – Exemplos)





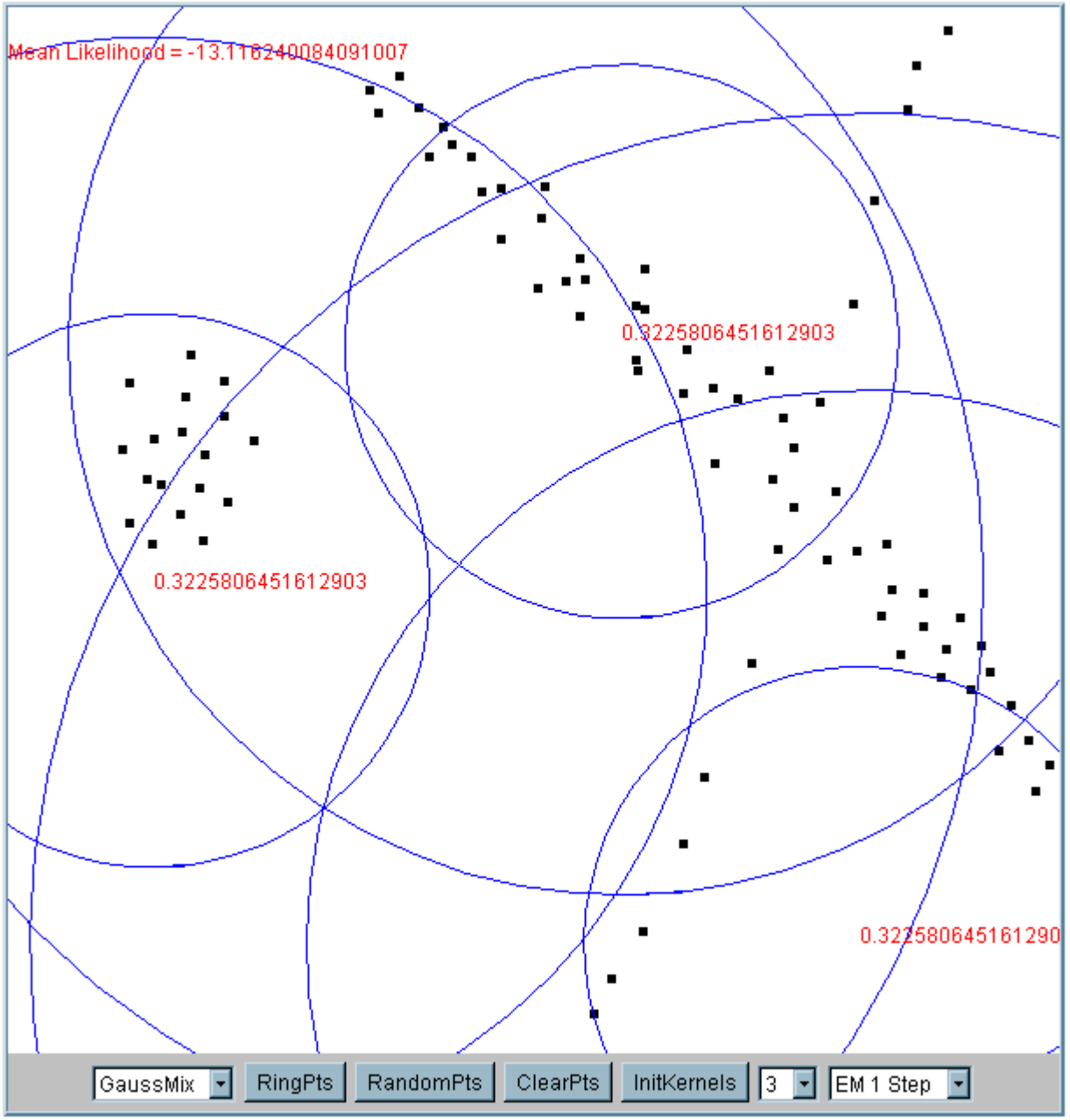
Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003,

# Exemplo (passo-a-passo)



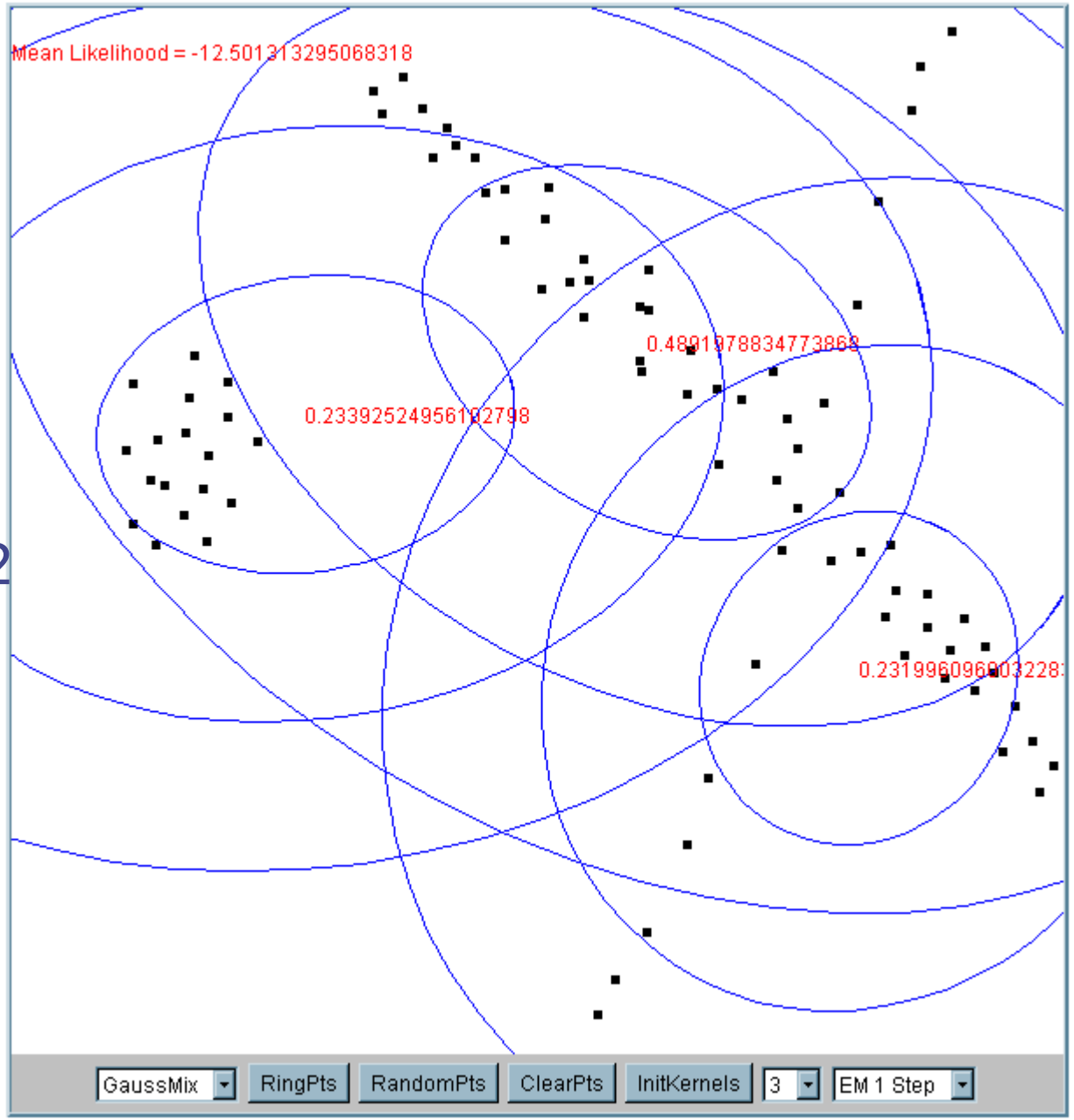

Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003,

Iteração 1

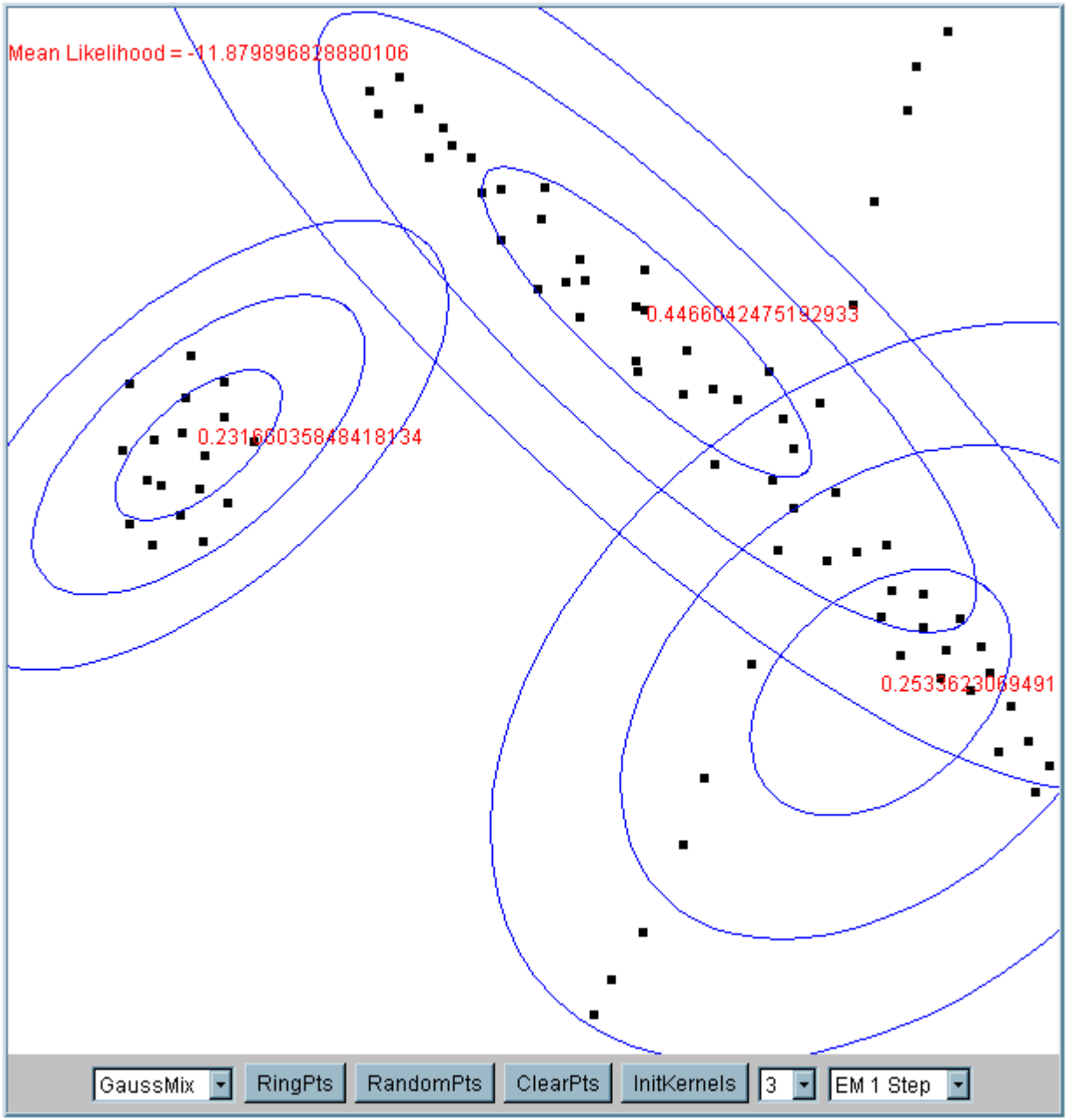



Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003,

# Iteração 2

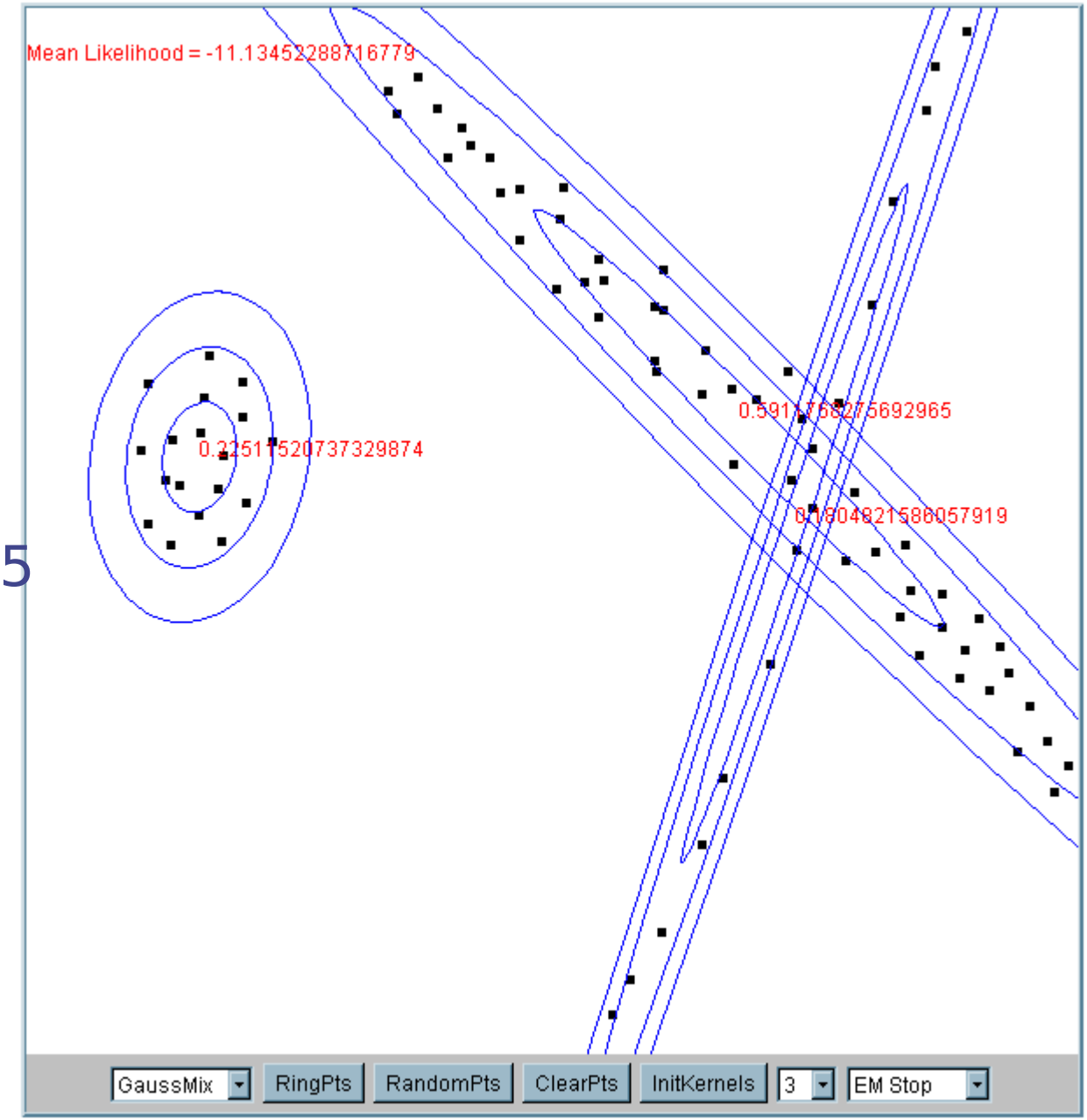

Mean Likelihood = -12.501313295068318
0.48918788347738
0.23392524956102798
0.23199609680322
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM 1 Step

Iteração 5




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

Iteração 25


Mean Likelihood = -11.13452288716779
0.22517520737329874
0.5911768275692965
0.18048215880057919
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 (D = 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

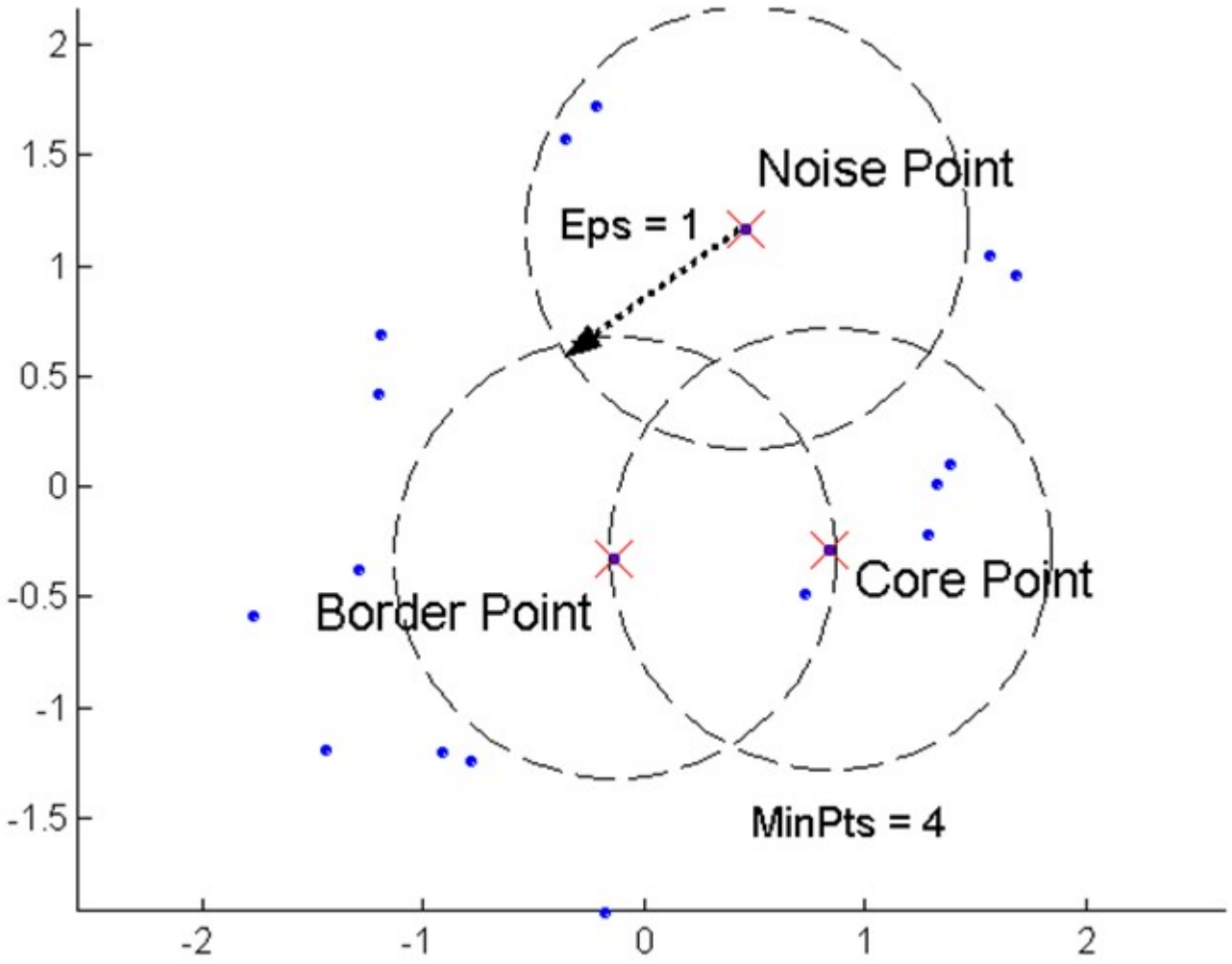
# Algoritmos Baseados em Densidade

- **Paradigma de Agrupamento por Densidade**
  - Clusters como regiões de alta concentração de objetos
    - separadas por regiões de baixa concentração de objetos
  - Paradigma alternativo àquele baseado em protótipos
    - K-means e variantes, EM, etc
- Existem vários algoritmos
- Veremos a seguir um dos mais conhecidos: **DBSCAN**

# DBSCAN

- DBSCAN é um algoritmo baseado em densidade
  - Utiliza o conceito de **Center-Based Density**
    - Número de pontos dentro de um raio (**Eps**)
- Definições:
  - Um ponto é um **core point** sem tem pelo menos um número pré-definido de pontos (**MinPts**) dentro de seu raio **Eps** (incluindo ele mesmo)
    - Estes pontos são o interior do grupo
  - Um **border point** tem menos que MinPts dentro de seu raio Eps, mas está na vizinhança (dentro do raio) de pelo menos 1 core point
  - Um **noise point** não é nem core nem border point**.**

# DBSCAN: Core, Border, and Noise Points



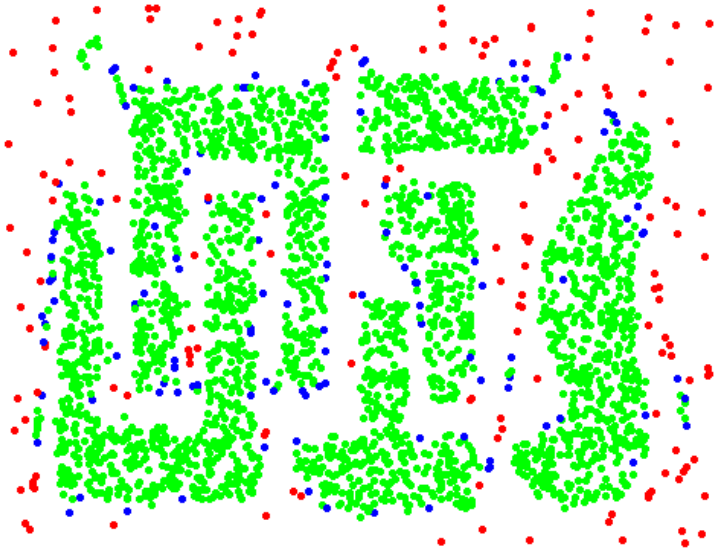
# DBSCAN

- **Algoritmo Conceitual:**

1. Percorra a BD e rotule os objetos como core, border ou noise
2. Elimine aqueles objetos rotulados como **noise**
3. Insira uma aresta entre cada par de objetos **core** vizinhos
   - 2 objetos são vizinhos se um estiver dentro do raio Eps do outro
4. Faça cada componente conexo resultante ser um cluster
5. Atribua cada **border** ao cluster de um de seus core associados
   - Resolva empates se houver objetos core associados de diferentes clusters

# DBSCAN: Core, Border and Noise Points

Pontos originais

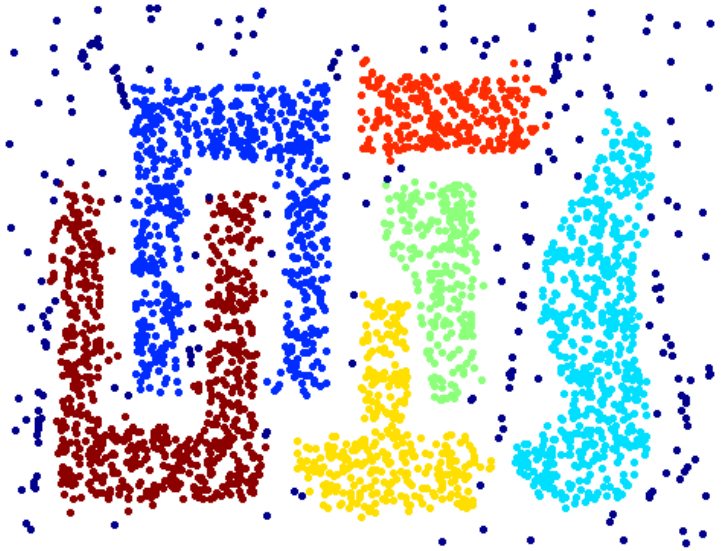


Tipos de pontos:
core, border e noise

Eps = 10, MinPts = 4

# Quando DBSCAN funciona bem

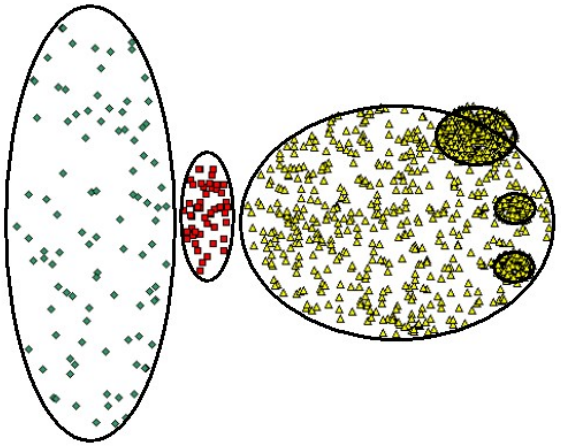
Pontos Originais



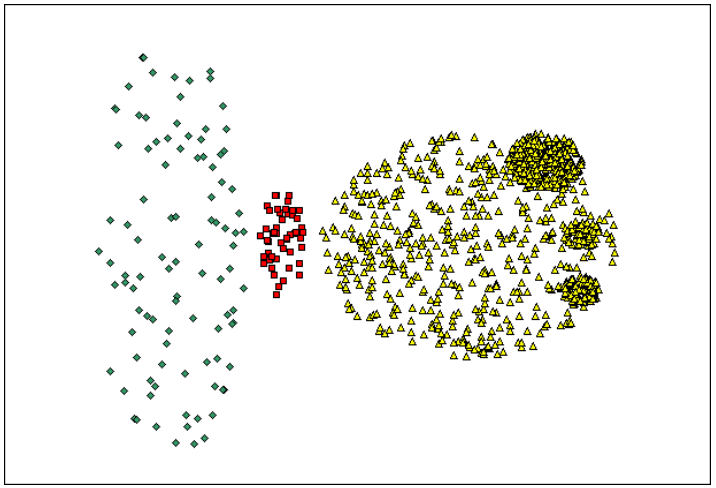
Grupos

- Resistentes a ruído
- Capaz de encontrar grupos de diferentes formatos e tamanhos

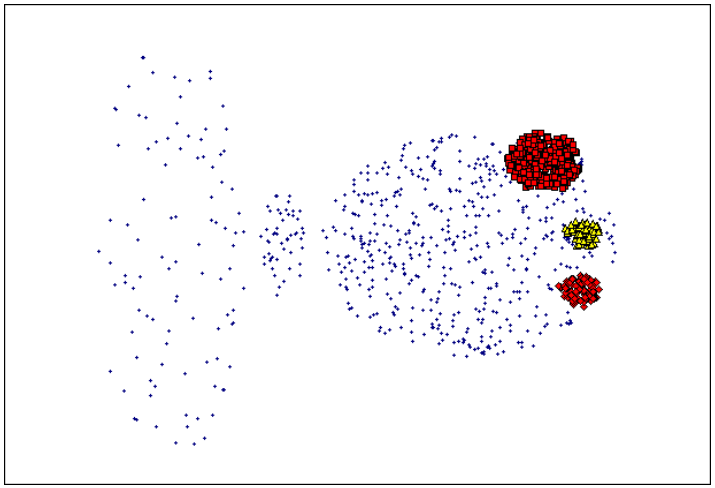
# Quando DBSCAN NÃO funciona bem



**Pontos originais**



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

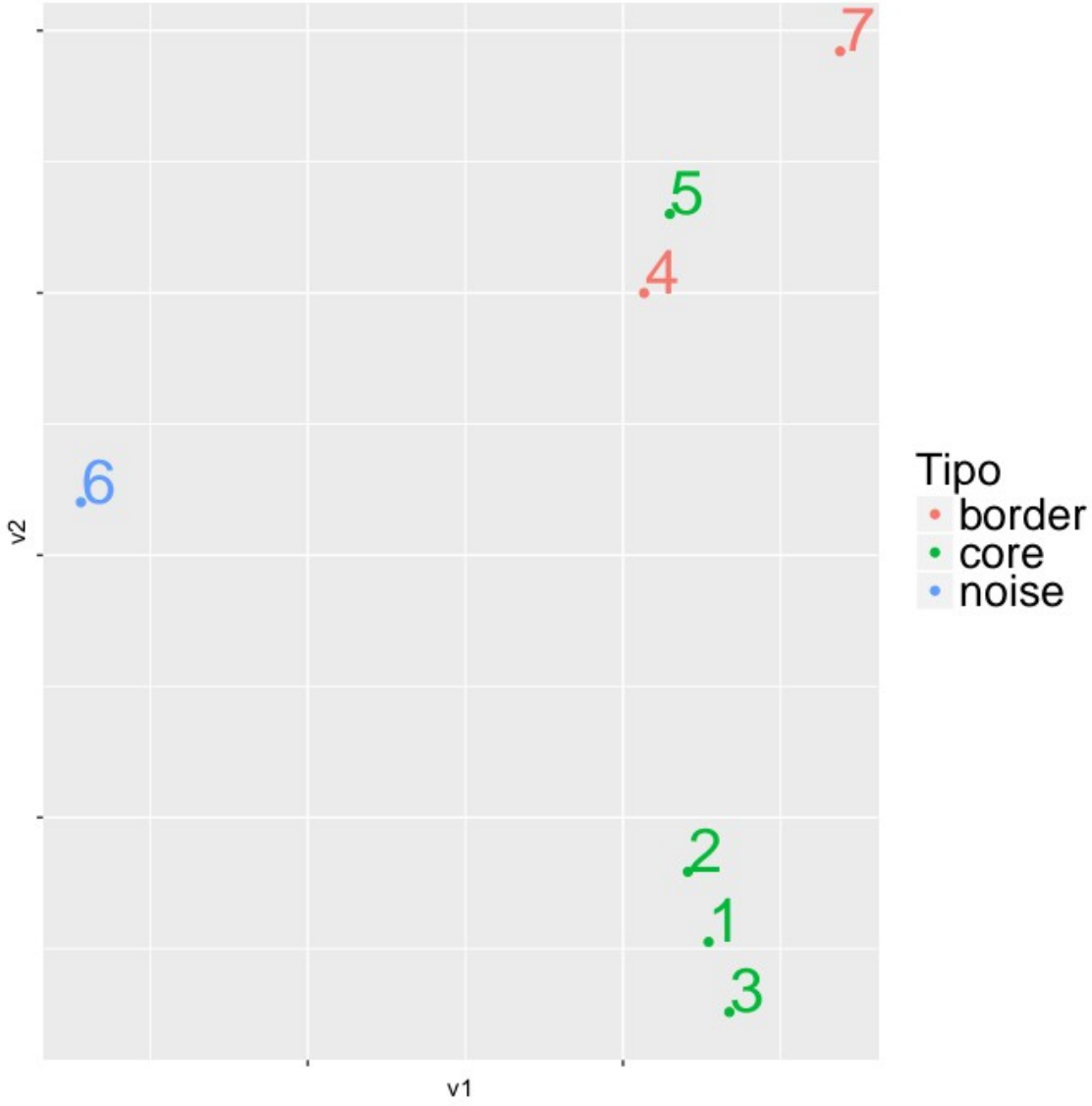
- **Grupos com diferentes densidades**
- **Dados em alta-dimensão**

# DBSCAN

- **Exercício:**
  - Aplique o algoritmo DBSCAN na BD abaixo, com Eps = 5 e MinPts = 3 (que inclui o objeto em questão), indicando os rótulos de cada objeto (core, border ou noise) e os grupos

$$\mathbf{D} = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{bmatrix} 0 & & & & & & \\ 1 & 0 & & & & & \\ 1 & 2 & 0 & & & & \\ 10 & 9 & 11 & 0 & & & \\ 11 & 10 & 12 & 2 & 0 & & \\ 21 & 20 & 22 & 18 & 19 & 0 & \\ 14 & 13 & 15 & 6 & 4 & 25 & 0 \end{bmatrix}$$

# DBSCAN



# 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