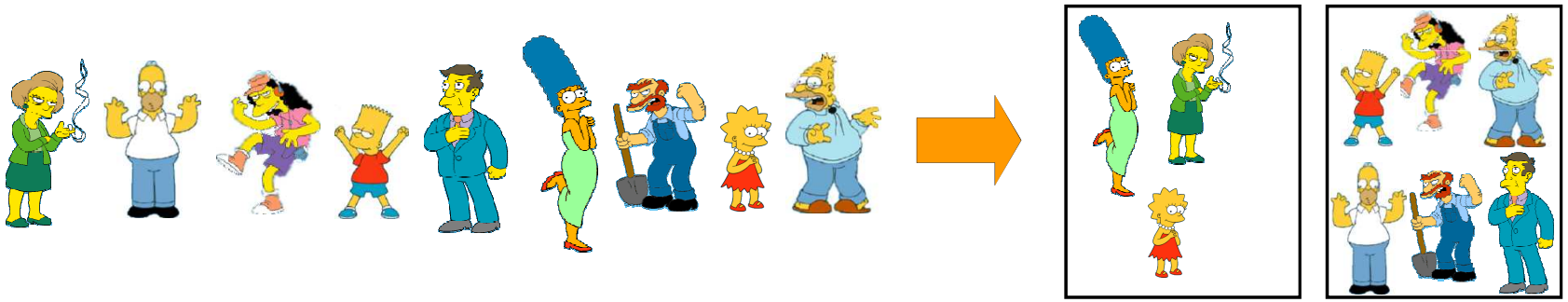
Organização

1. Introdução
2. Medidas de Similaridade
3. Métodos de Agrupamento (métodos hierárquicos, **de partição**)
4. Critérios numéricos para definir o número de *clusters*

Métodos de Partição

- Cada exemplo pertence a um *cluster* dentre *k clusters* possíveis;
- Usuário normalmente deve fornecer o número de *clusters (k)*;
- Normalmente envolvem a otimização de algum índice (critério numérico) que reflete a qualidade de determinada partição;



→ Vamos iniciar por um algoritmo amplamente utilizado (*k*-means), o qual fornecerá uma noção mais intuitiva do problema a ser resolvido.

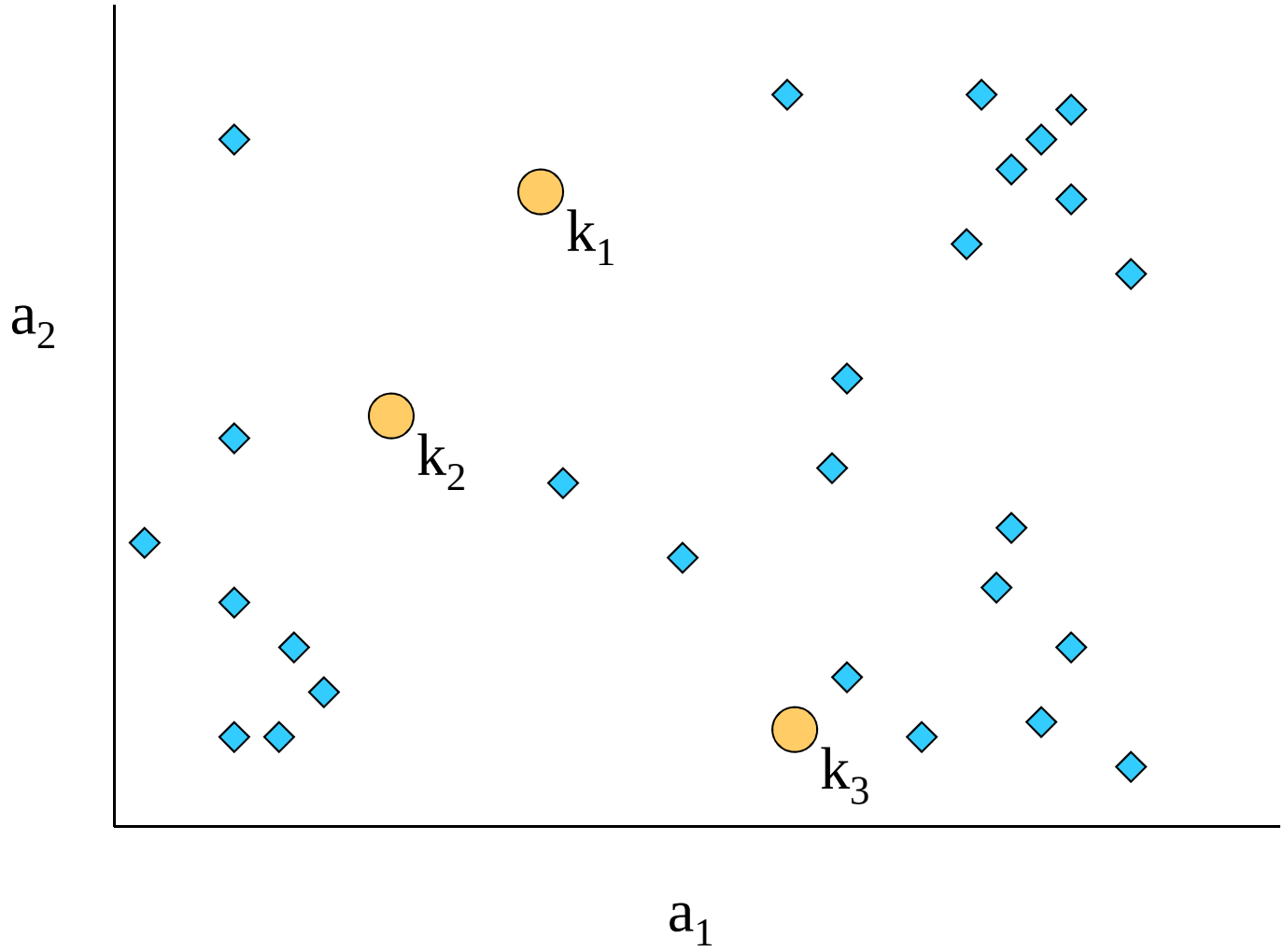
Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Manaus.

k-means :

1) Escolher aleatoriamente um número *k* de centros para os clusters;
2) Atribuir cada objeto para o cluster de centro mais próximo (e.g. usando a distância Euclidiana)
3) Mover cada centro para a média dos objetos atribuídos;
4) Repetir os passos 2 e 3 até que algum critério de convergência seja obtido (número de iterações, tolerância em relação às mudanças nos centróides).

k-means - passo 1:

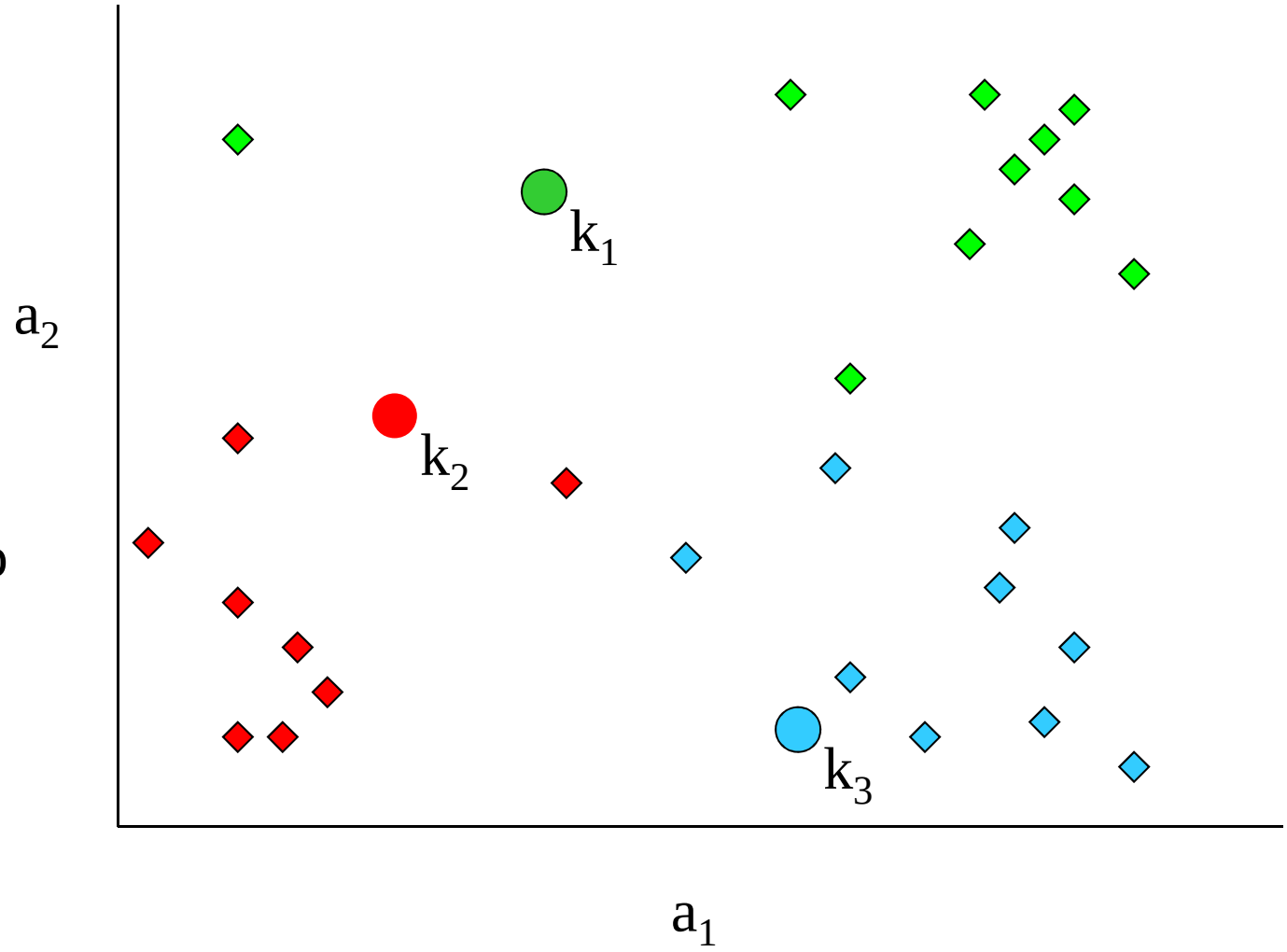
Escolher 3 centros iniciais



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

k-means - passo 2:

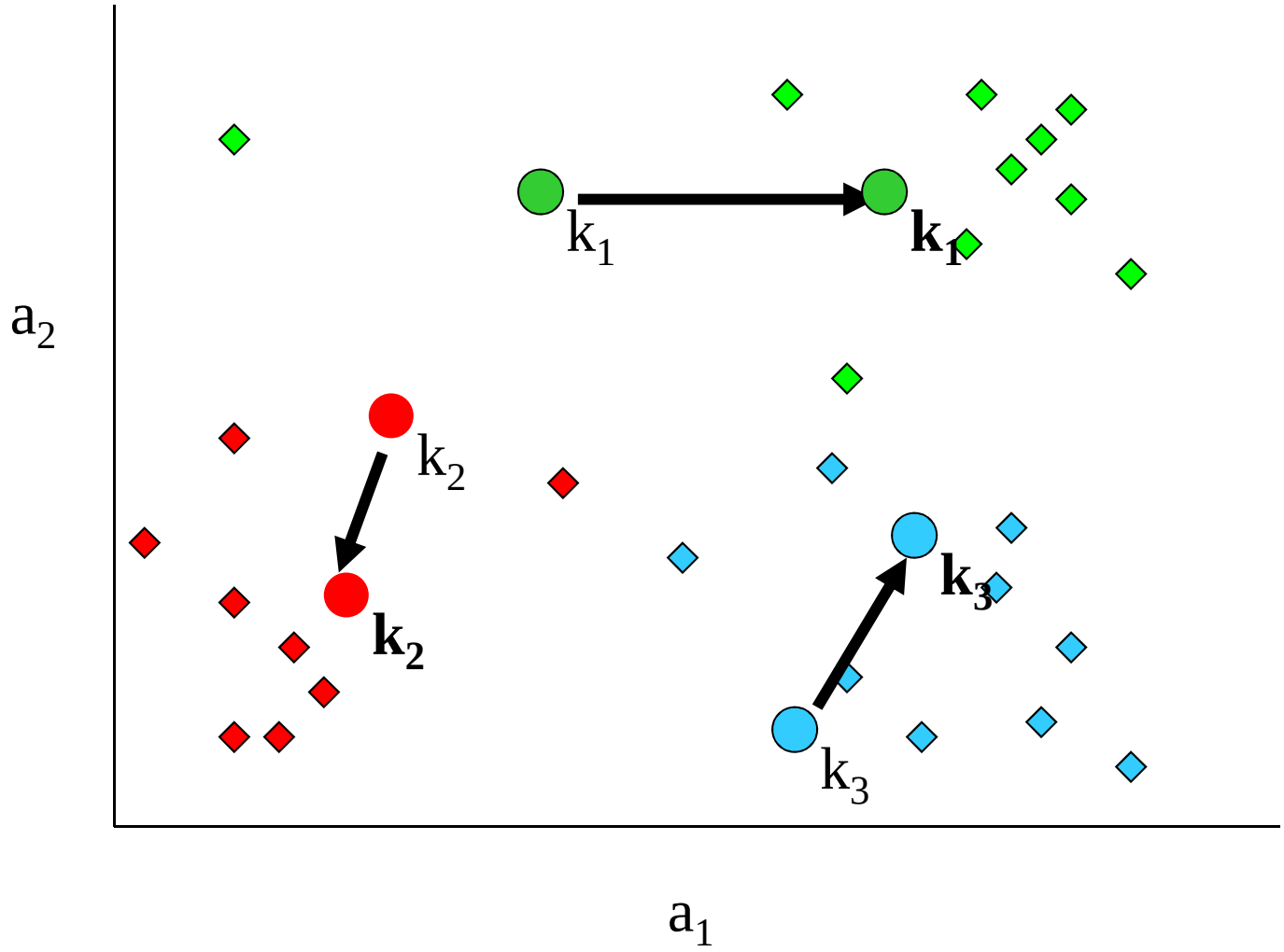
Atribuir cada ponto ao cluster de centro + próximo



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

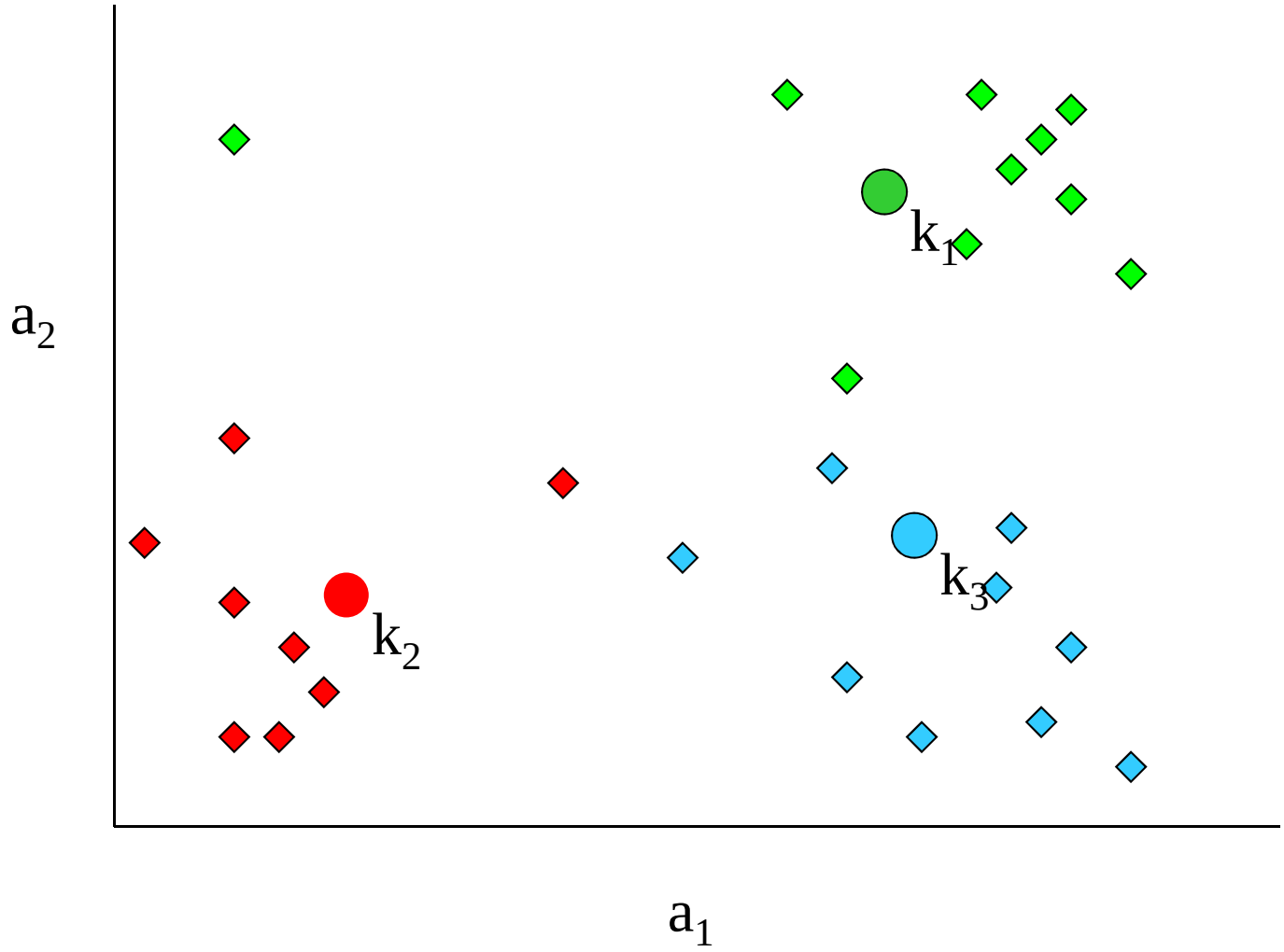


Slide baseado no curso de Gregory Piatetsky-Shapiro, disponível em http://www.kdnuggets.com

k-means - passo 4:

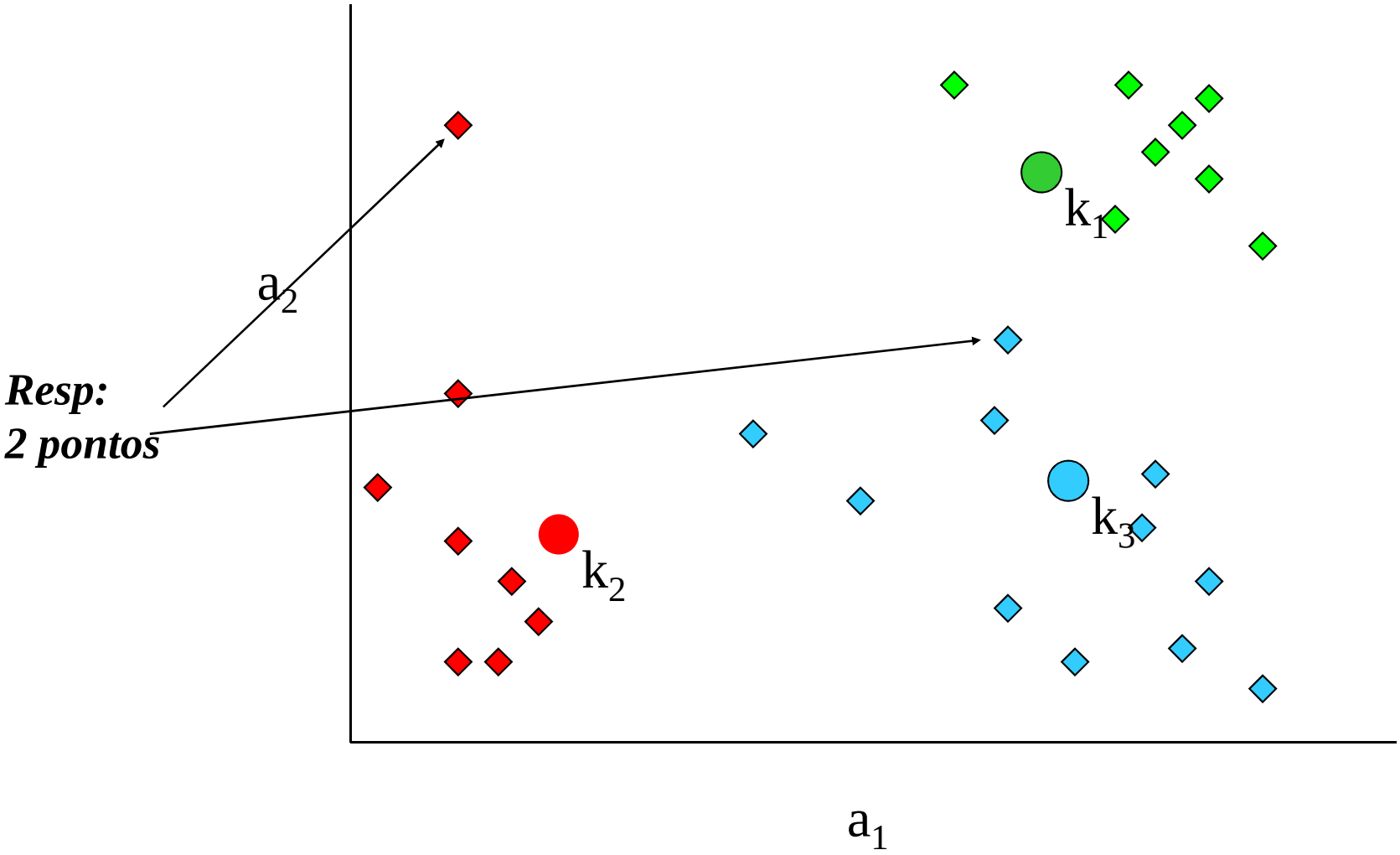
Re-atribuir objetos aos clusters de centróides mais próximos

Quais objetos mudarão de cluster?



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

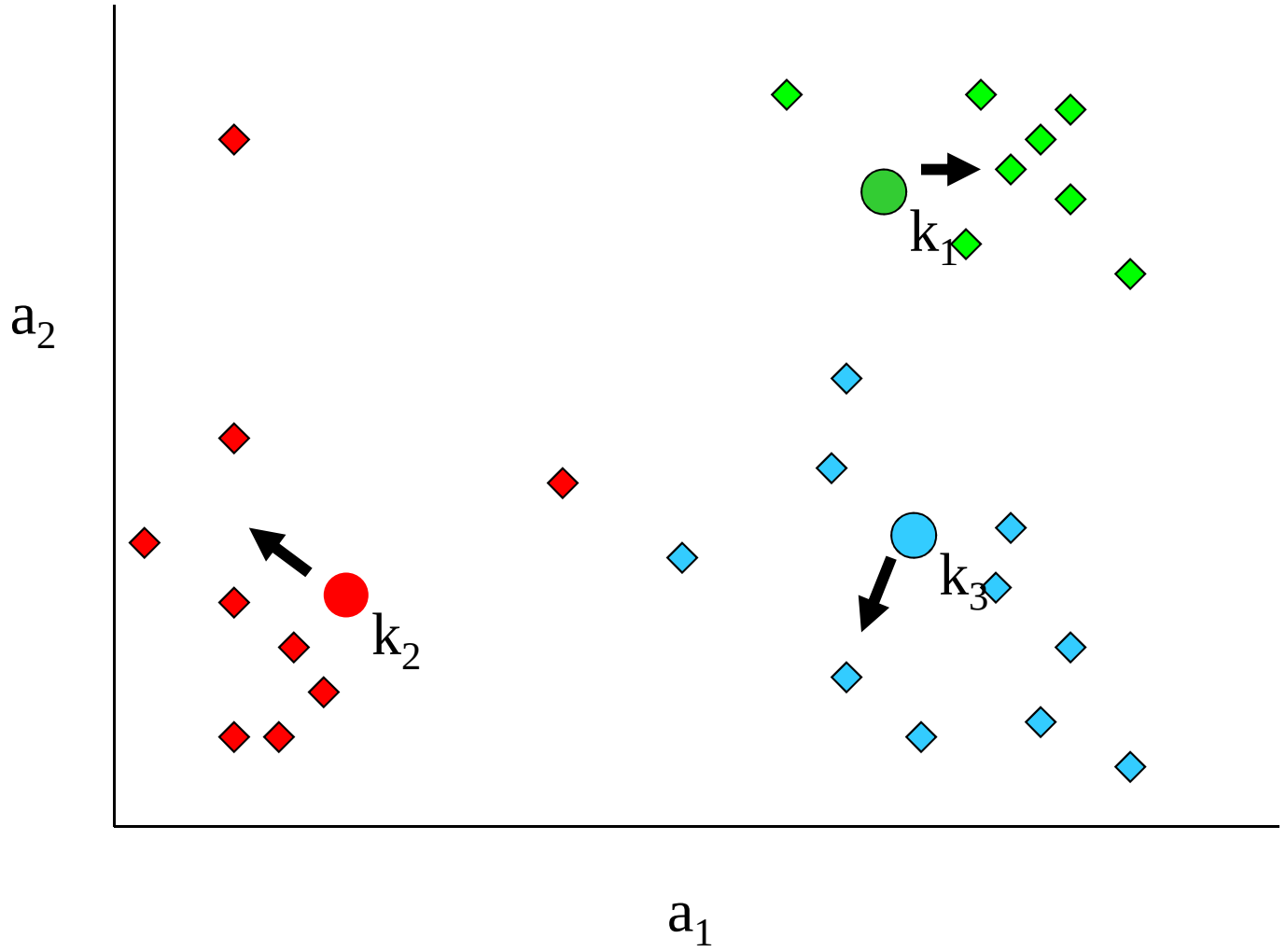
k-means - passo 4 ...



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

k-means - passo 4:

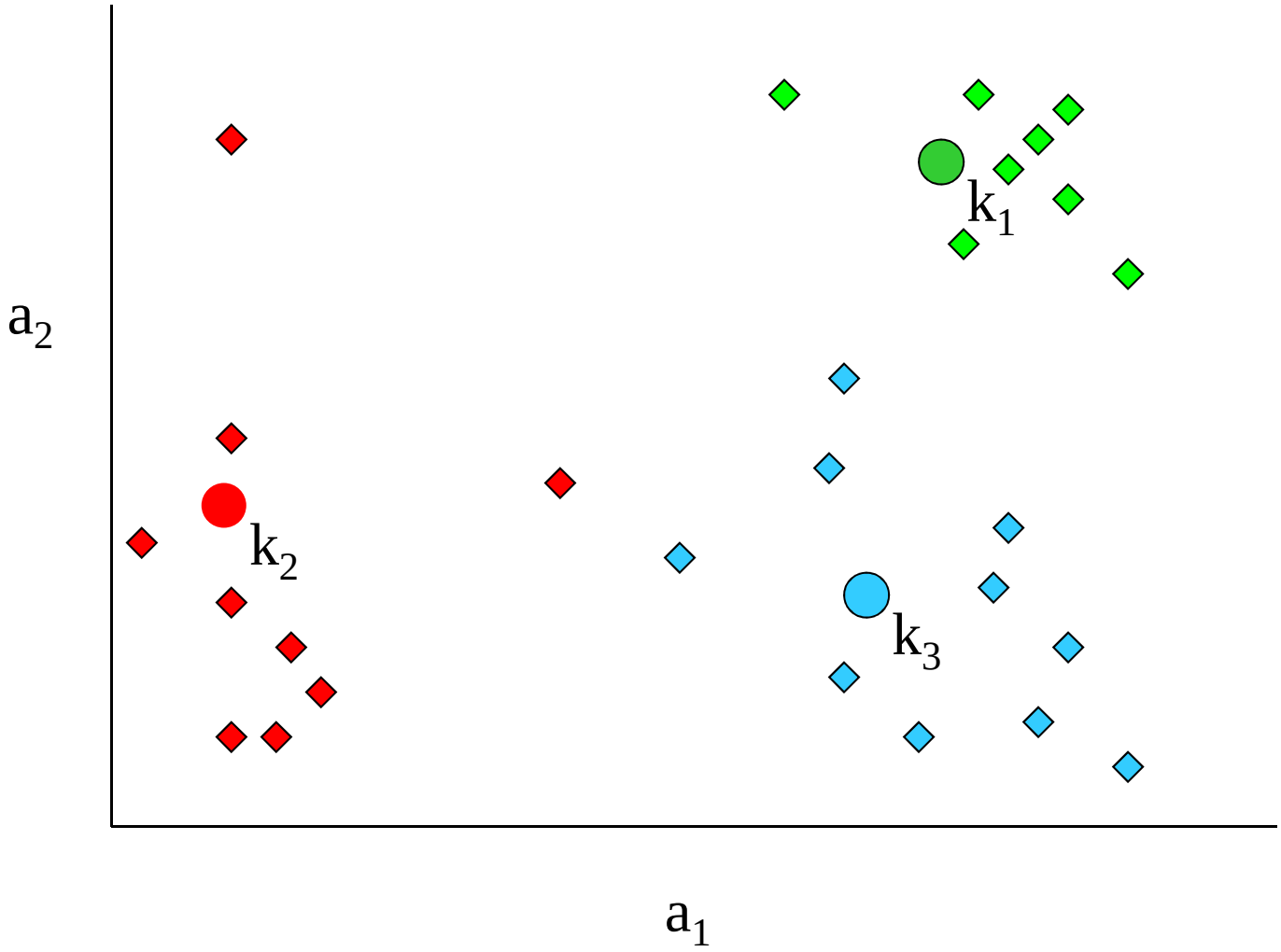
re-calcular
vetores
médios



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

k-means - passo 5:

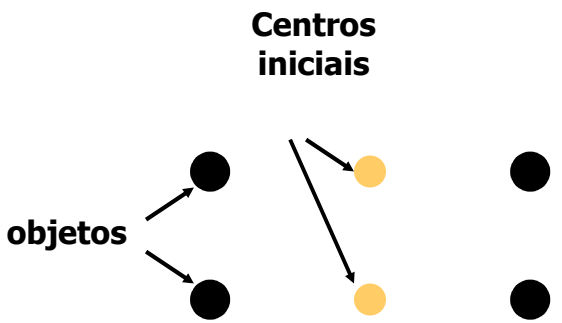
mover
centros dos
clusters…



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

Discussão

- Resultado pode variar significativamente dependendo da escolha das sementes iniciais;
- *k*-means pode ficar preso em ótimos locais;
  - Exemplo:



- Para aumentar a chance de encontrar ótimos globais: várias inicializações diferentes.

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

Observações sobre o *k*-means:

Vantagens

- Simples
- Itens são automaticamente atribuídos aos clusters

Desvantagens

- *k* = ?
- Cada item deve pertencer a um único cluster
- Sensível a *outliers*

Como o *k-means* poderia ser adaptado pra usar os medóides em vez dos centróides?

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

Algumas variações do *k*-means:

- Substituir as médias pelas medianas de cada cluster.
  - 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 da Mediana: menos afetada por valores extremos.
- Para grandes bases de dados pode-se usar técnicas de amostragem: e.g. CLARA e CLARANS, ambos baseados no PAM (partitioning around medoids - Kaufman & Rousseeuw, Finding groups in Data, 1990).

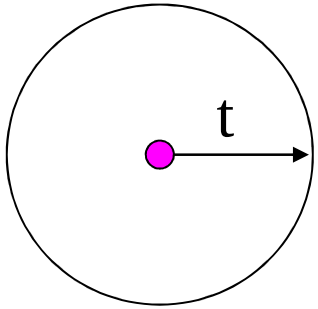
Slide baseado no curso de Gregory Piatetsky-Shapiro, disponível em http://www.kdnuggets.com

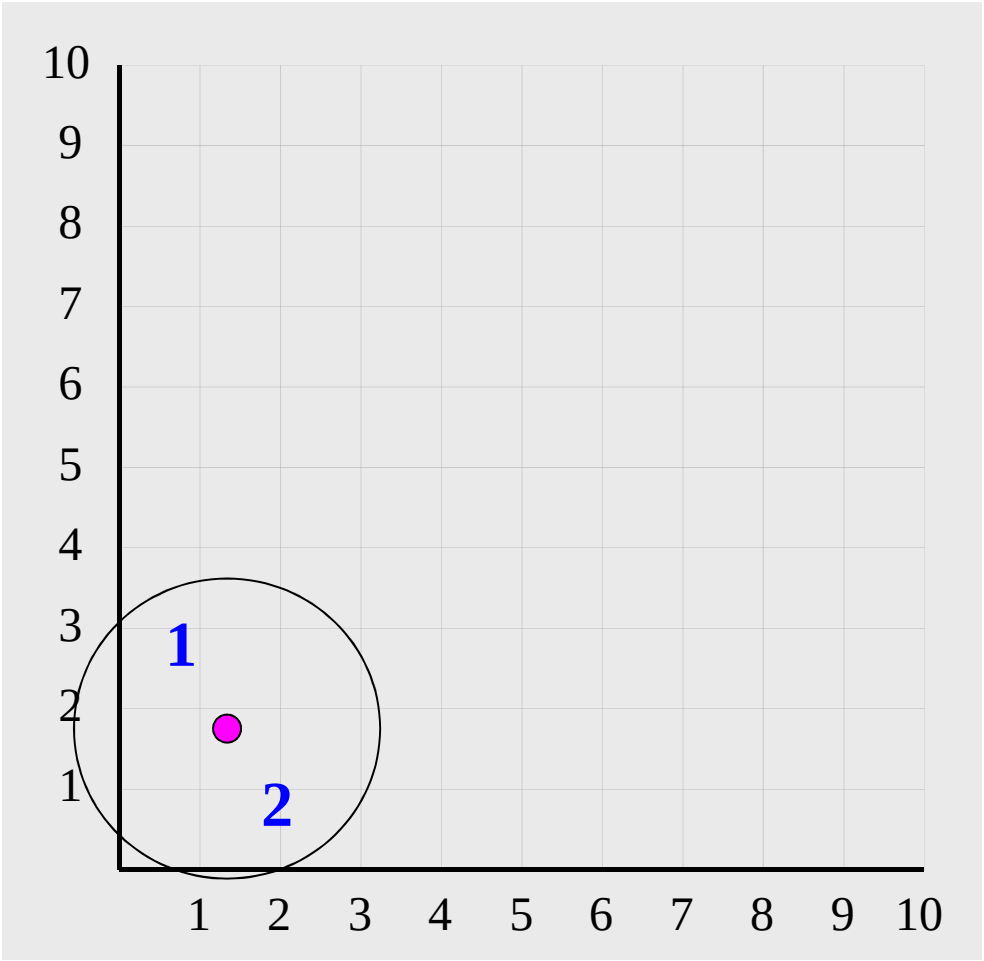
O que podemos fazer quando se tem uma atualização contínua da base de dados?

→ **Uma alternativa envolve usar o conceito de vizinho mais próximo (*Nearest Neighbor Clustering*):**

- Objetos são dinamicamente incorporados aos *clusters* mais próximos;
- Incremental;
- Depende de um limiar (*threshold* - *t*), usado para determinar se os objetos são incorporados aos grupos existentes ou se um novo grupo deve ser criado.

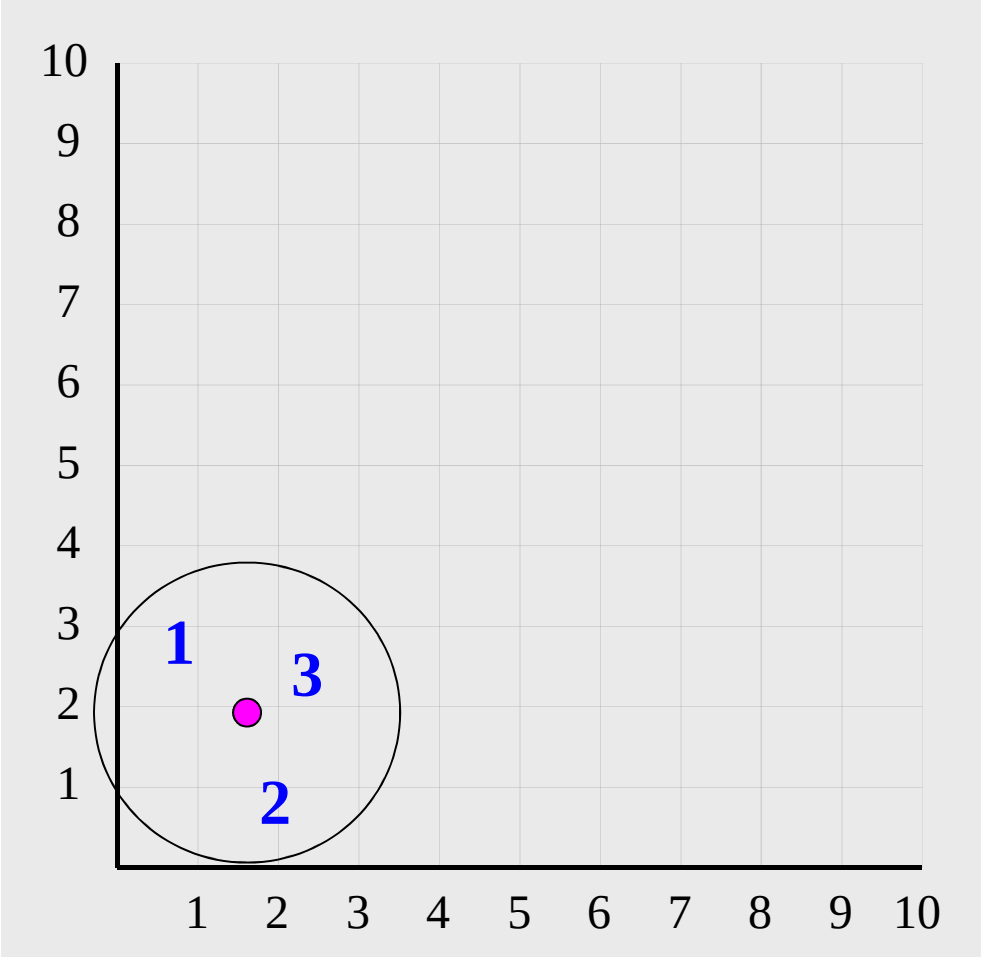
Threshold t





Novo objeto (3) é inserido na base de dados…

Este objeto está dentro do limiar para pertencer ao *cluster 1*. Neste caso, adicionamo-lo a este *cluster* e atualizamos o seu centróide.



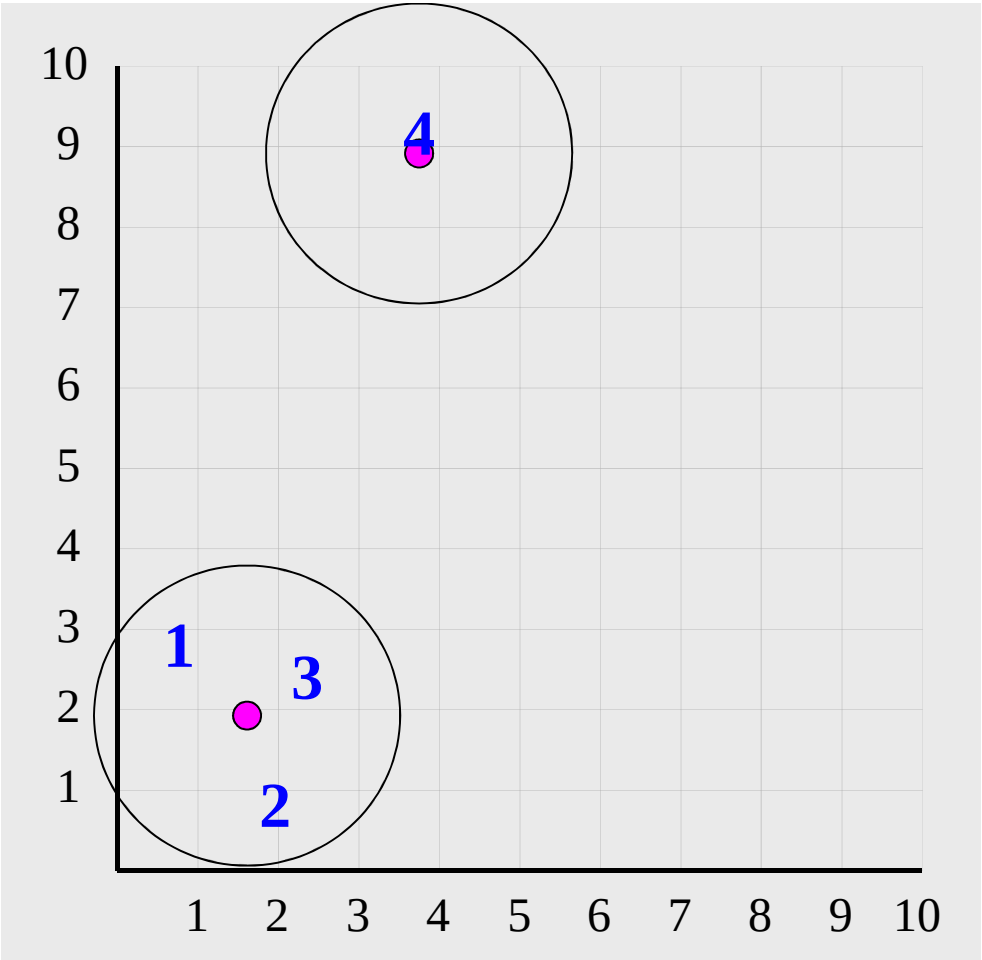
Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Manaus.

Outro objeto (4) é inserido na base.

Este não está dentro do limiar para o cluster 1. Neste caso, criamos um novo cluster e assim por diante...

Resultado obtido é altamente dependente da ordem de apresentação dos objetos…

Como determinar *t* ?



Algoritmos para grandes bases de dados: BIRCH, DBSCAN, CURE – ver, por exemplo Dunham, M.H., Data Mining – Introductory and Advanced Topics, Prentice Hall, 2003.

Formalização do Problema de Agrupamento de Dados *(clustering):*

- Denotemos por $\mathbf{X}_{nxp}$ uma matriz formada por *n* linhas (correspondentes aos objetos da base de dados) e *p* colunas (atributos):

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

Particionar um conjunto **X** de objetos em uma coleção de sub-conjuntos mutuamente disjuntos $\mathbf{C}_i$ de **X**. Formalmente, consideremos um conjunto de *n* objetos $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_n\}$ a ser agrupado, onde cada $\mathbf{x}_i \in \Re^\rho$ é um vetor de atributos consistindo de ρ medições reais. Os objetos devem ser agrupados em grupos não sobrepostos $\mathbf{C}=\{\mathbf{C}_1,\mathbf{C}_2,...,\mathbf{C}_k\}$, onde *k* é o número de *clusters*, 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 $i \neq j$.

- Uma definição para o problema de agrupamento de dados (Dunham, 2003):

  - Dada uma base de dados $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_n\}$, $\mathbf{x}_i \in \Re^\rho$, o problema de agrupamento envolve definir um mapeamento $f:\mathbf{X}\rightarrow\{1,2,...,k\}$ onde cada $\mathbf{x}_i$ é atribuído a um *cluster* $\mathbf{C}_j$, $1\leq j\leq k$. Um *grupo de dados* $\mathbf{C}_j$ contém precisamente as tuplas mapeadas para o mesmo, ou seja, $\mathbf{C}_j = \{ \mathbf{x}_i \mid f(\mathbf{x}_i)=\mathbf{C}_j, 1\leq i\leq n$ e $\mathbf{x}_i \in \mathbf{X} \}$.

- Assumindo-se que o número de grupos (k) seja conhecido, o número de maneiras (*NM*) de se 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)≈56.6x10^{67}. Supondo que um computador tenha capacidade de processar 10^9 avaliações de partições por segundo (por exemplo do *k-means)*, precisaríamos de aproximadamente 1.8x10^{50} séculos para processar todas as avaliações!
- Necessitamos de heurísticas;
- Índices normalmente refletem dois conceitos: homogeneidade (compactação) e separabilidade (dispersão).

Alguns Critérios (Everitt et al., Cluster Analysis, 2001):

Assumindo que n_m é o número de objetos do cluster m, d_{ij} é a distância entre objetos i e j pertencentes a um mesmo *cluster* m e $r \in \{1,2\}$:

a) Falta de homogeneidade do *cluster m*:

$$h_1(m) = \sum_{l=1}^{n_m} \sum_{v=1, v \neq l}^{n_m} d_{lv}{}^r$$

$$h_2(m) = \max_{\substack{l=1,\ldots,n_m \\ v=1,\ldots,n_m \\ v \neq l}} d_{lv}^r$$

$$h_3(m) = \min_{v=1,\ldots,n_m} \left[\sum_{l=1}^{n_m} d_{lv}{}^r \right]$$

b) Separação dos *clusters:*

$$s_1(m) = \sum_{l=1}^{n_m} \sum_{k \neq m} \sum_{v=1}^{n_k} d^r_{ml,kv}$$

$$s_2(m) = \min_{\substack{l=1,\ldots,n_m \\ v=1,\ldots,n_k \\ k \neq m}} d^r_{ml,kv}$$

Agora que já dispomos de algumas alternativas para avaliar a homogeneidade e a separação de cada *cluster* individualmente, como avaliar uma partição *(clustering)*, isto é, um conjunto de *clusters* obtidos?

c) Escolhido um índice para medir a homogeneidade/separação de cada *cluster* individualmente,podemos usar os seguintes critérios para avaliar a *qualidade geral* da partição, como por exemplo:

$$c_1(n,k) = \sum_{m=1}^{k} h(m)$$

$$c_2(n,k) = \max_{m=1,\ldots,k} h(m)$$

$$c_3(n,k) = \min_{m=1,\ldots,k} h(m)$$

É importante observar que alguns critérios são maximizados quando a *melhor* partição é obtida, enquanto que outros são minimizados.

Os critérios vistos até aqui podem ser usados para avaliar partições formadas por diferentes números de clusters ?

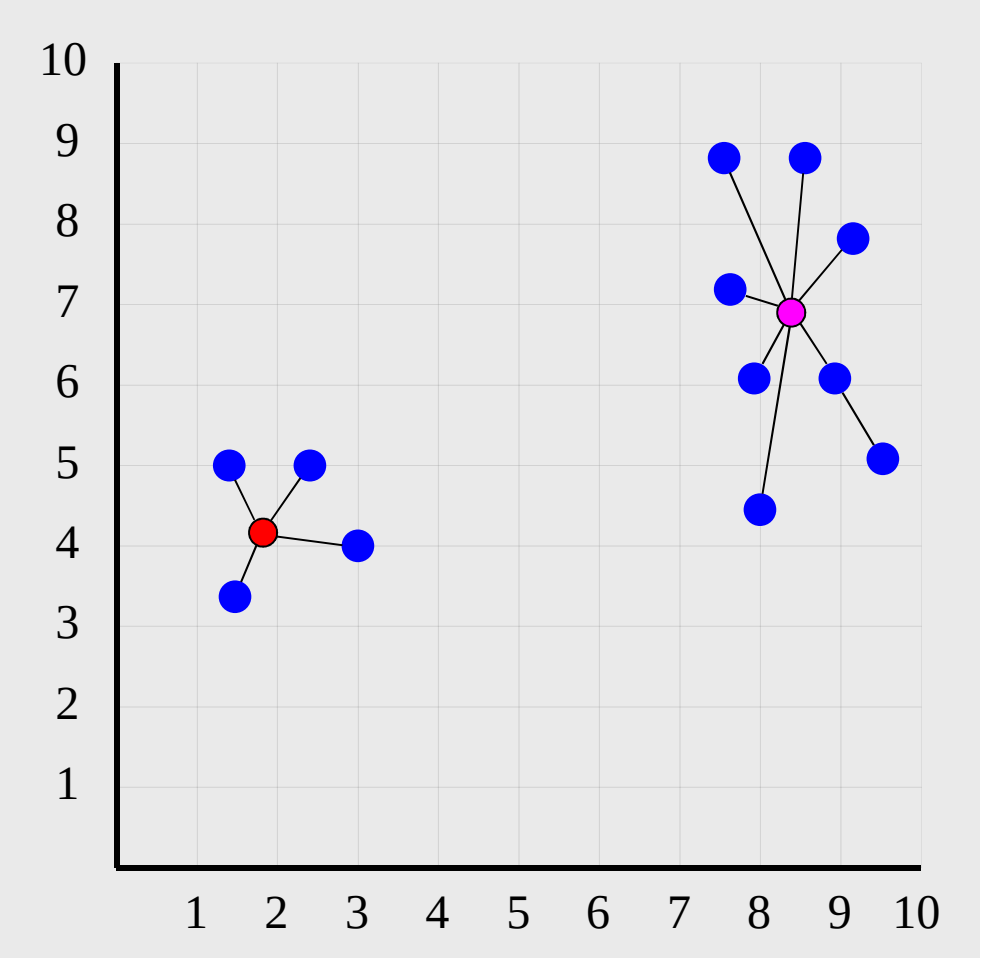
Critérios numéricos para avaliar partições para *k* desconhecido *a priori:*

- Como estimar o número *correto* de *clusters*? Em geral, este é um problema não resolvido. Entretanto, podemos usar alguns critérios numéricos *razoáveis*.

Erro Quadrático (EQ):

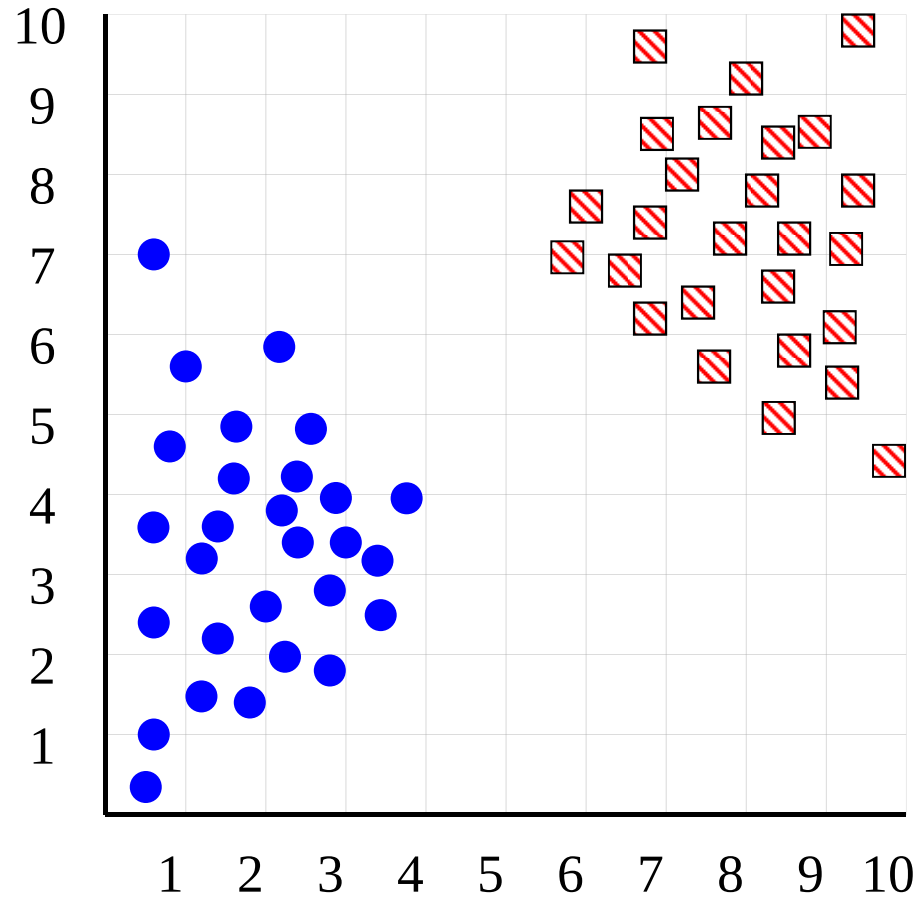
$$EQ = \sum_{m=1}^{k} \sum_{l=1}^{n_m} \left\| \mathbf{x}_{ml} - \bar{\mathbf{x}}_m \right\|^2$$

Função Objetivo



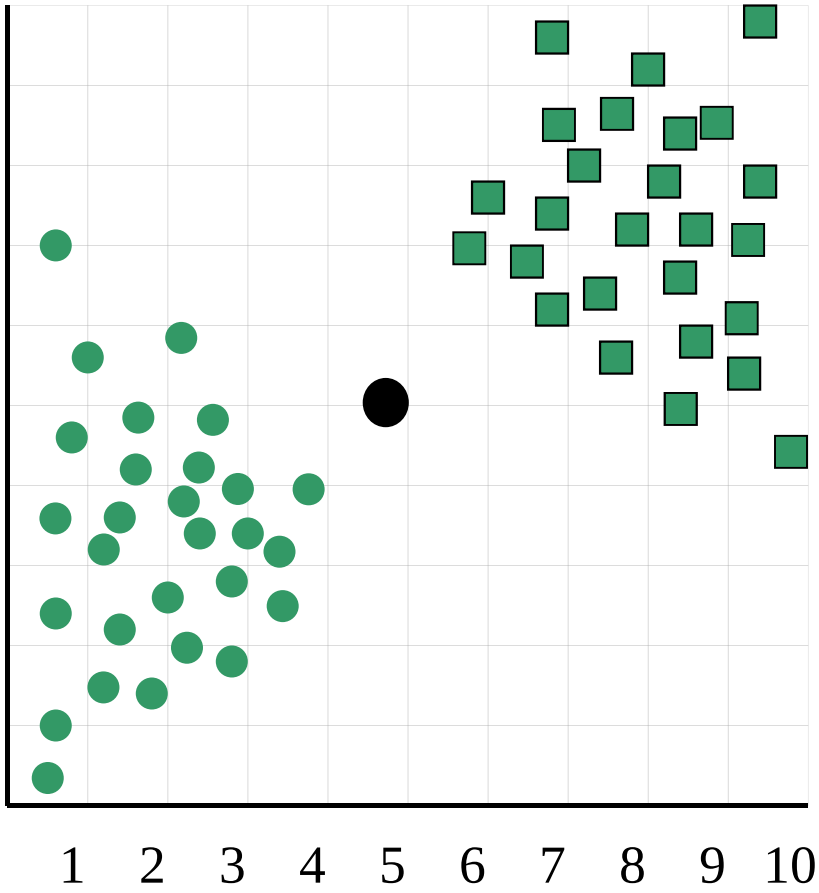
Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Manaus.

Exemplo:

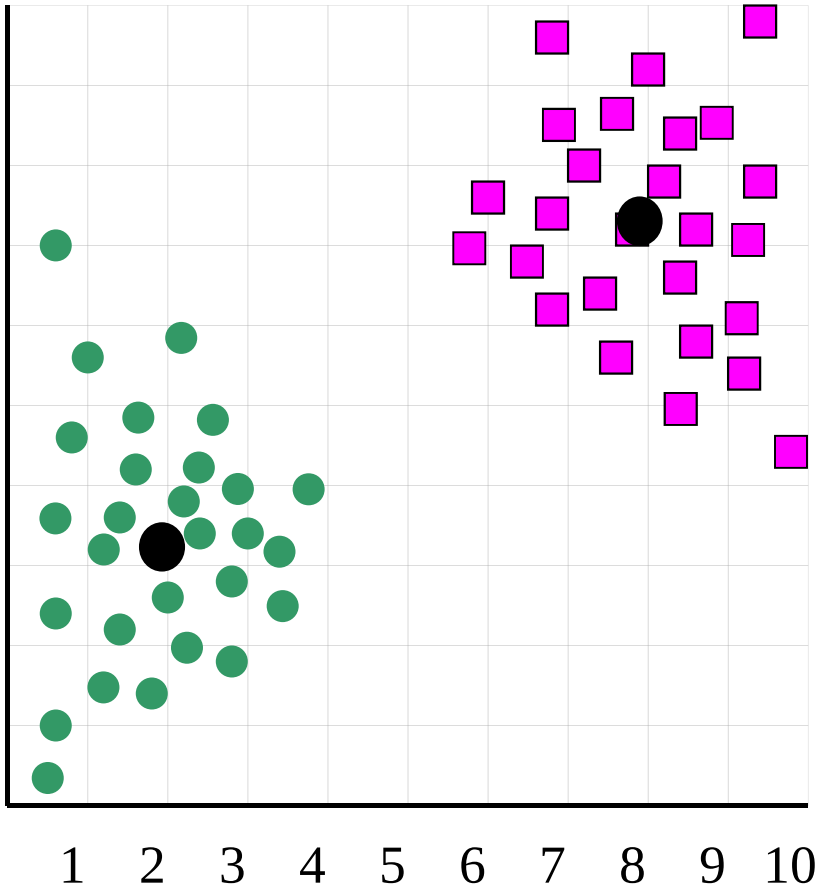


Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Manaus.

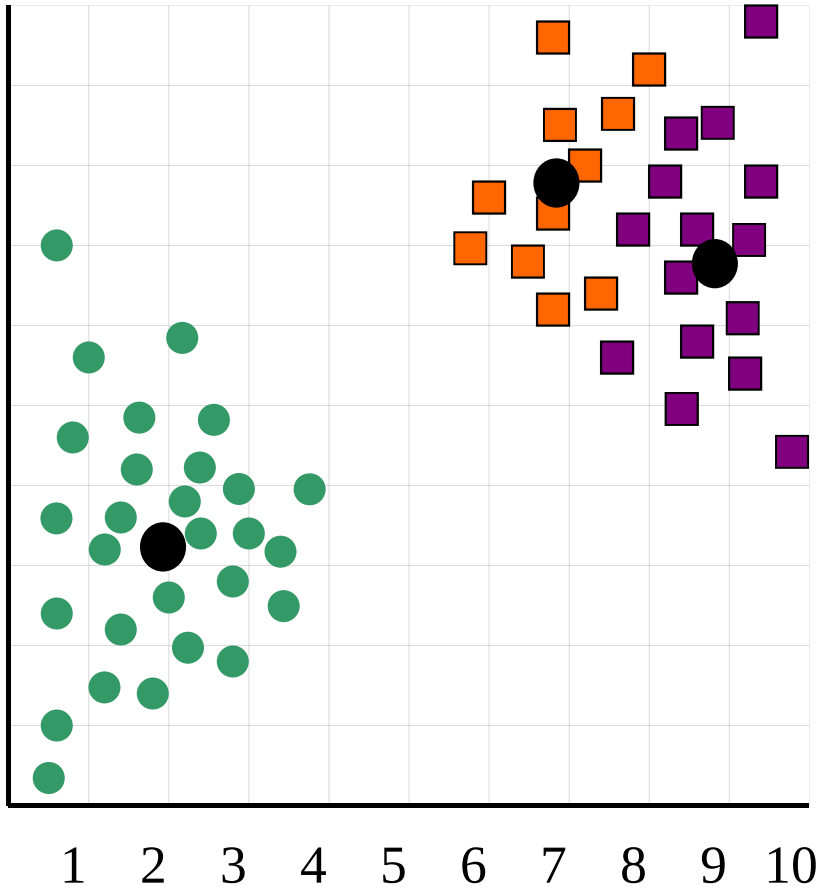
Para $k = 1$, o valor da função objetivo é 873,0.



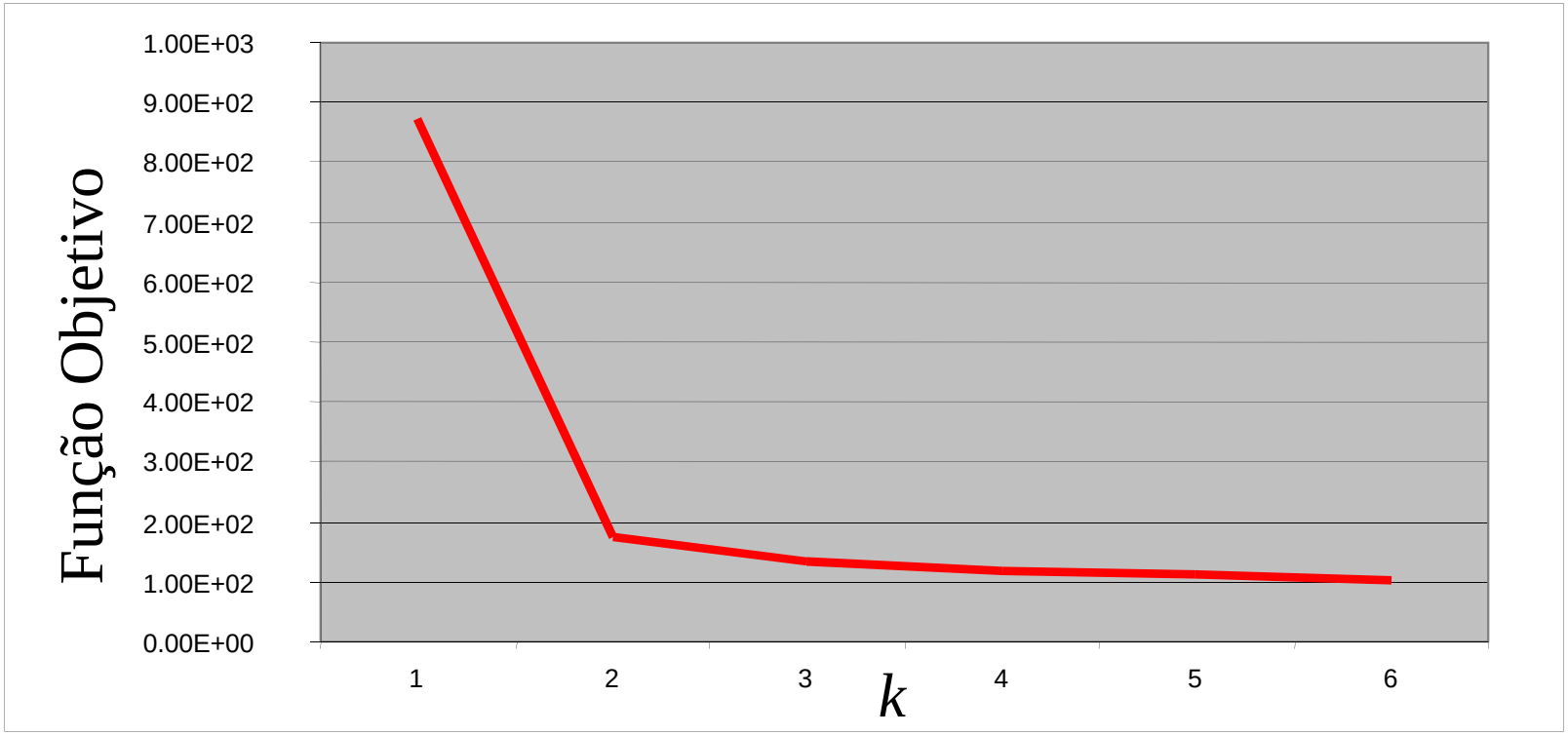
Para $k = 2$, o valor da função objetivo é 173,1.



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



Podemos então repetir este procedimento e plotar os valores da função objetivo para k =1,…6 e tentar identificar um joelho (ou cotovelo):



Infelizmente os resultados não são sempre tão claros quanto neste exemplo…Além disso, como automatizar?

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

Outro critério que pode ser usado envolve o cálculo da silhueta (Kaufman & Rousseeuw, Finding Groups in Data, An Introduction to Cluster Analysis, 1990):

Consideremos um objeto *i* que pertença ao cluster **A**. A dissimilaridade média de *i* em relação a todos os outros objetos de **A** é denotada por $a(i)$. Consideremos agora um cluster **B**. A dissimilaridade média de *i* em relação a todos os objetos de **B** será chamada de $d(i,\mathbf{B})$. Após computar $d(i,\mathbf{B})$ para todos os clusters $\mathbf{B} \neq \mathbf{A}$, a menor destas é selecionada e chamada de *b(i)*, isto é $b(i) = \min d(i,\mathbf{B})$, $\mathbf{B} \neq \mathbf{A}$. Este valor representa a dissimilaridade de *i* em relação ao seu *cluster vizinho* e $s(i)$ é então dada por:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \in [-1, +1]$$

Após calcular *s(i)* para cada objeto $i=1,...N$ calculamos o índice de validade (função objetivo) *f:*

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

Outros tópicos:

- Algoritmos de agrupamento probabilísticos, técnicas para otimizar o número de clusters e as partições correspondentes, extração de regras usando partições, aplicações, etc.
  - Estudar a vasta literatura disponível na biblioteca!
- Como comparar algoritmos de agrupamento?
  - Qualidade (eficácia);
  - Eficiência.