O que é Agrupamento?

Conhecido também por *aprendizado não-supervisionado* e, às vezes, chamado de *classificação* por estatísticos e de *segmentação* por pessoas de *marketing*

- O que é agrupamento?
- Algumas aplicações
- Definição formal e complexidade computacional
- Objetivos e características
- Grupos naturais
- Medidas de (dis)similaridade
- A tarefa de agrupamento e desafios
- Agrupamento particional
  - K-Médias
  - C-Médias Nebuloso
- Agrupamento hierárquico
  - Single-link
- Medidas de avaliação

O que é Agrupamento?

- Agrupar é organizar coisas similares em categorias; é a capacidade de identificar características similares, como forma, tamanho, cor...

- Assim, é possível organizar grandes bases de dados com o objetivo de facilitar seu entendimento

- Intuitivamente, objetos pertencentes a um mesmo grupo são similares entre si e dissimilares a grupos distintos

O que é Agrupamento?

Algumas aplicações

- Medicina: identificação de categorias de diagnósticos
- Marketing: identificar grupos de clientes, produtos, serviços
- Arqueologia: identificar relações entre diferentes tipos de objetos
- Finanças: identificar o perfil de clientes fraudadores, ou transações fraudulentas

O que é Agrupamento?

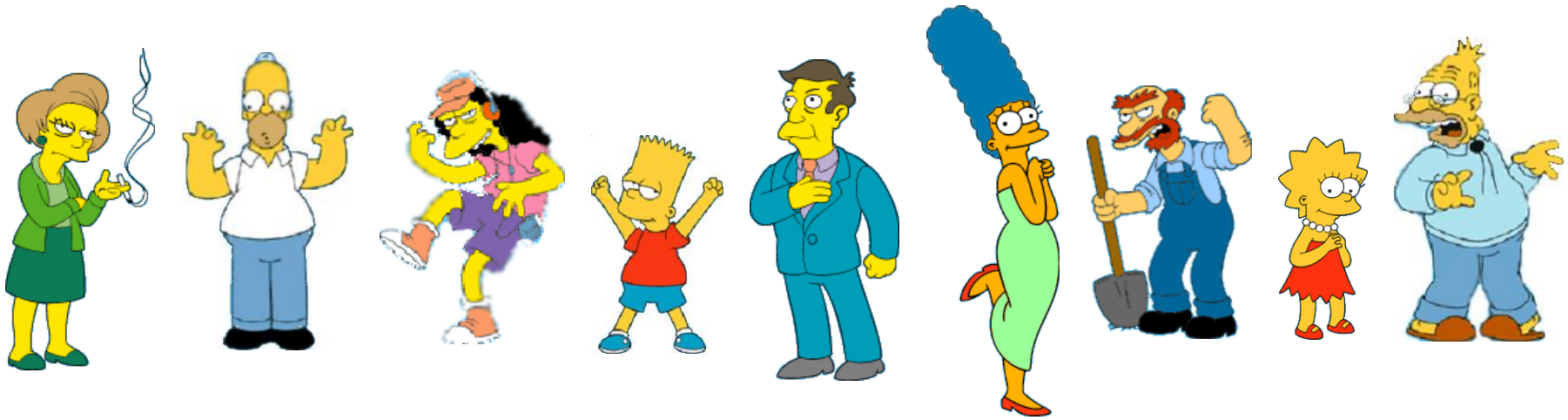
- O processo de agrupamento se refere à organização de objetos em grupos usando alguma medida de similaridade ou dissimilaridade entre eles, tal que objetos do mesmo grupo tenham características comuns entre si

- Diferentemente dos processos de classificação, a análise de *clusters* considera dados de entrada não-rotulados, ou seja, a classe a qual cada padrão de entrada (objeto) pertence não é conhecida *a priori*

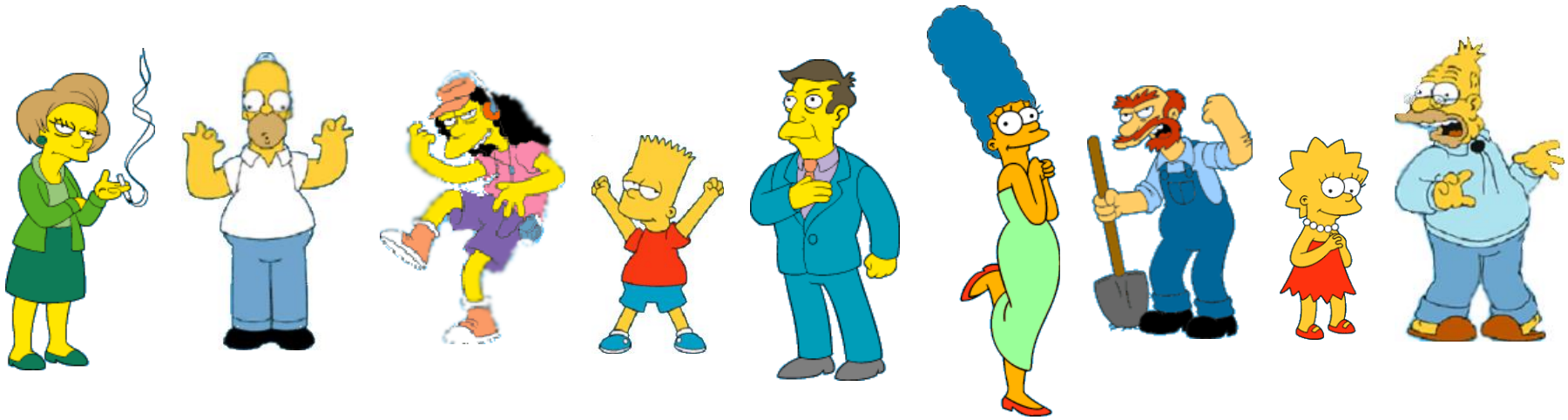
- Alguma medida deve ser aplicada para avaliar a similaridade entre os objetos. Geralmente é usada alguma medida de distância

$$d\left(x_a^d, x_b^d\right) = \sqrt{\sum_{i=1}^{d}\left(x_a^i - x_b^i\right)^2}$$ (distância Euclidiana)

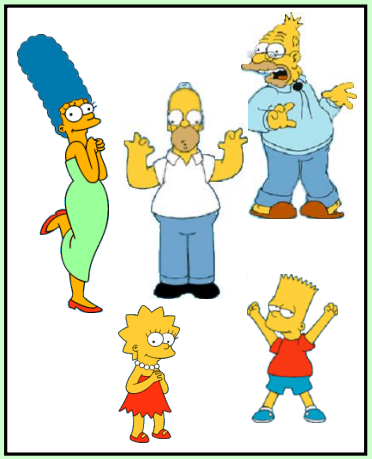
Qual é o agrupamento natural entre esses objetos?



Qual é o agrupamento natural entre esses objetos?

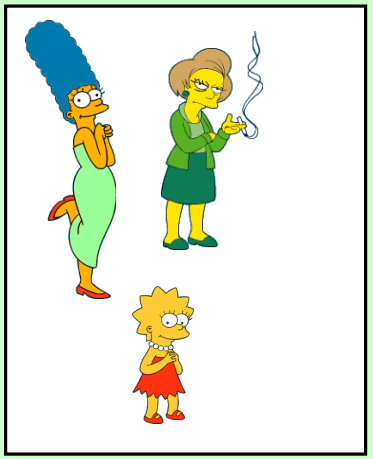


Agrupamento é subjetivo



Família Simpson

Empregados da escola



Mulheres

Homens

Definição formal e Complexidade Computacional

- Dado um conjunto de *n* objetos não rotulados $X = \{\mathbf{x}_1^d, \mathbf{x}_2^d, \ldots, \mathbf{x}_n^d\}$ definido no espaço $\Re^d$, estes objetos devem ser agrupados em *k* grupos distintos $C = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_k\}$, tal que objetos similares entre si pertençam ao mesmo grupo

- O conjunto *X* de objetos pode ser agrupado de diversas maneiras, isto é, há um número de formas distintas de agrupar *n* objetos em *k* grupos ($n \geq k$) de modo a obter o agrupamento ótimo

Definição formal e Complexidade Computacional

- Qual o número de formas distintas de agrupar *n* objetos em *k* grupos ($n \geq k$) de modo a obter o agrupamento ótimo?

$$P(n,k) = \frac{1}{k!}\sum_{j=1}^{k}(-1)^{k-j}\binom{k}{j}j^n$$

Sendo P o número de possibilidades de agrupar *n* objetos em *k* grupos

- Ex.:

Para *n* = 10 objetos e *k* = 2 grupos, há 511 possibilidades

Para *n* = 15 objetos e *k* = 2 grupos, há 16383 possibilidades

Objetivos e características

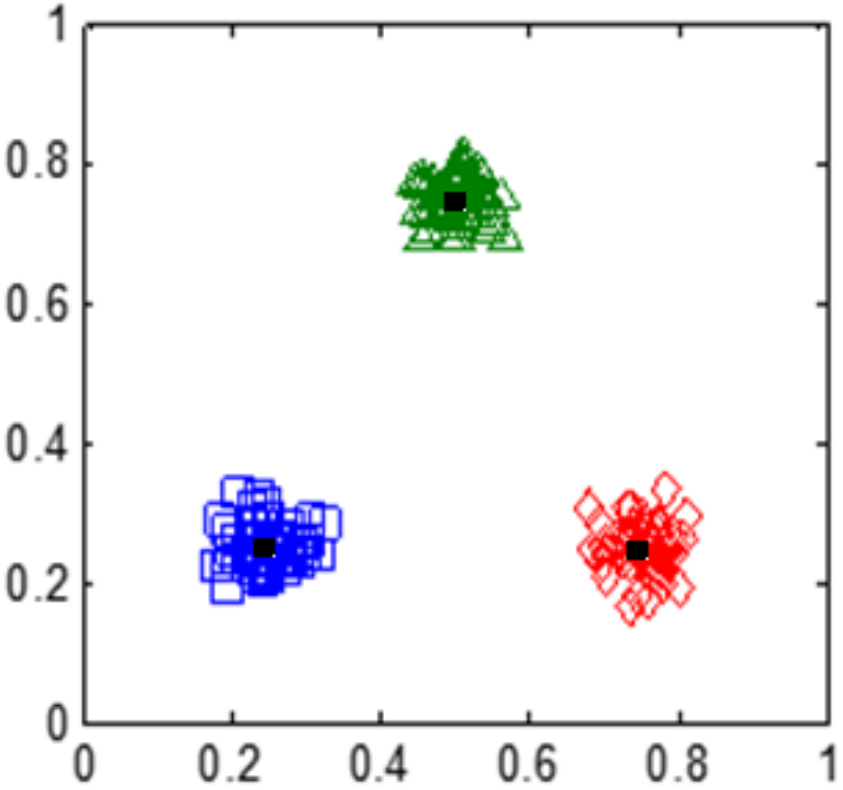
- A tarefa de agrupamento opera sobre *objetos não rotulados*, ou seja, a classe (ou grupo) a qual cada objeto de entrada pertence não é conhecida *a priori*

- Algoritmos de agrupamento normalmente são usados para identificar tais classes e descrever suas características

- Cada objeto da base de dados é representado por um ponto no espaço vetorial, cuja dimensão é dada pelo número de atributos do objeto

Objetivos e características

- O agrupamento pode ser baseado em *protótipos*, ou seja, vetores específicos que representam grupos (formados por grandes quantidades) de objetos, o que possibilita reduzir o tamanho da base de dados, minimizar o custo de processamento e facilitar a análise das características da base de dados



Objetivos e características

- Ao término do processo de agrupamento, os protótipos normalmente estão posicionados no centro dos respectivos grupos ou em regiões de maior densidade de objetos, de modo a maximizar a representatividade dos objetos nos grupos encontrados

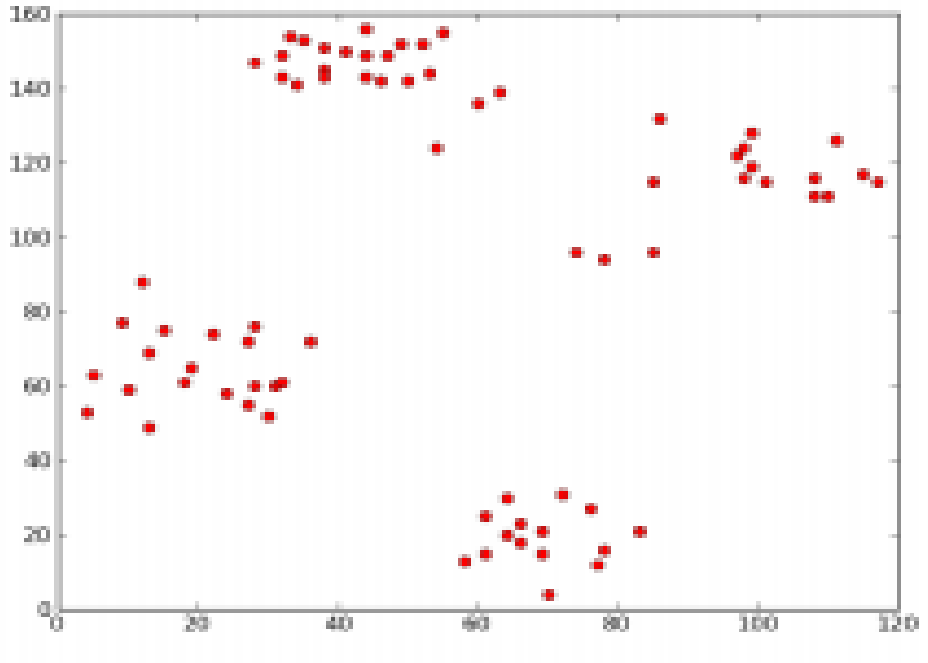
- A ideia é encontrar *grupos naturais* em bases de dados

Grupos naturais

- São aqueles que satisfazem duas condições (Carmichael, 1968):

1. Existência de regiões contínuas do espaço, relativamente densamente povoadas por objetos; e

2. Essas regiões estão rodeadas por regiões relativamente vazias do espaço



Medidas de (dis)similaridade

Que medida devo usar para avaliar a similaridade?

- Ela está diretamente relacionada às características da base de dados a ser agrupada, isto é, às formas dos grupos naturais e à dimensão do espaço de soluções

- A distância Euclidiana, por exemplo, é usada para identificar grupos esféricos, sendo essas características normalmente desconhecidas *a priori*

- A escolha da medida de similaridade apropriada também é um fator que afeta a qualidade das soluções

A tarefa de agrupamento e desafios

- A tarefa de agrupamento pode ser dividida em cinco passos:

1. *Representação*:

   As características dos dados a serem avaliados devem ser representadas por estruturas manipuláveis pelo algoritmo de agrupamento

x_{11}	...	x_{1D}
.	.	.
.	.	.
.	.	.
x_{N1}	...	x_{ND}

Matriz de N objetos (X) de dimensão D
Cada linha é um objeto (registro da base de dados)
Cada coluna é uma dimensão (atributo) do objeto

A tarefa de agrupamento e desafios

• A tarefa de agrupamento pode ser dividida em cinco passos:

2. *Definição de uma medida de proximidade ou distância*:
   Usualmente é usada uma função que avalia a semelhança ou a distância entre os objetos

   –A distância Euclidiana é uma das mais usadas na literatura

A tarefa de agrupamento e desafios

3. *Agrupamento*:

   Refere-se ao processo de busca de grupos de objetos em uma base de dados

4. *Abstração do dado*:

   Esta tarefa refere-se ao processo de descrição dos grupos encontrados

5. *Avaliação da saída*:

   Avaliação da qualidade dos grupos encontrados

A tarefa de agrupamento e desafios

Alguns desafios da tarefa de agrupamento:

- Determinação automática do número de grupos
- Multidimensionalidade
- Grupos não separáveis linearmente
- Medida de similaridade apropriada

Métodos de Agrupamento

- Há três métodos principais de agrupamento de dados

1. Particional
2. Hierárquico
3. Baseado em densidade

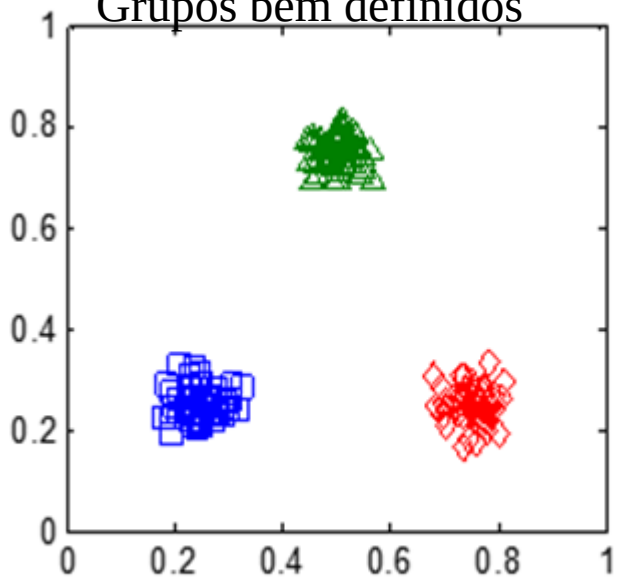
Agrupamento particional

- Um conjunto de objetos é particionado em *k* grupos (partições)

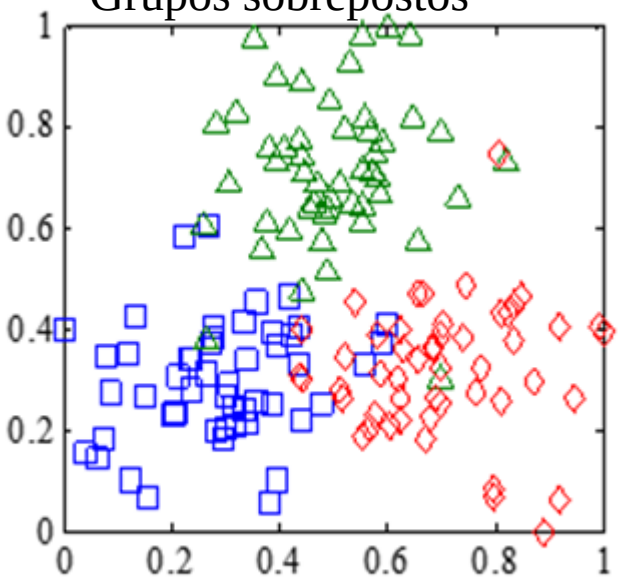
a) ***Hard*** *(Tradicional, Crisp) ou* ***Nebuloso*** (difuso):

Um algoritmo particional é classificado como *hard* quando cada objeto pertence a um único grupo, ou *nebuloso* se houver um grau de pertinência aos grupos para cada um dos objetos

Grupos bem definidos



Grupos sobrepostos



Agrupamento particional

- Um conjunto de objetos é particionado em *k* grupos (partições)

b. ***Determinístico*** *ou* ***Estocástico***:

Algoritmos determinísticos são aqueles que *apresentam o mesmo resultado* sempre que executados, ou seja, apresentam sempre o mesmo agrupamento e a mesma quantidade de iterações toda vez que são executados, considerando a mesma inicialização paramétrica

Os algoritmos estocásticos, por outro lado, *podem apresentar diferentes soluções* para o mesmo problema, dependendo de parâmetros como inicialização e ordem de apresentação dos objetos

O algoritmo k-Médias (ou k-Means)

•Ele recebe o parâmetro *k* como entrada, tal que o algoritmo encontre *k* grupos em uma dada base de dados

•O algoritmo funciona da seguinte maneira:

a. *k* centroides são inicializados, aleatoriamente ou não;

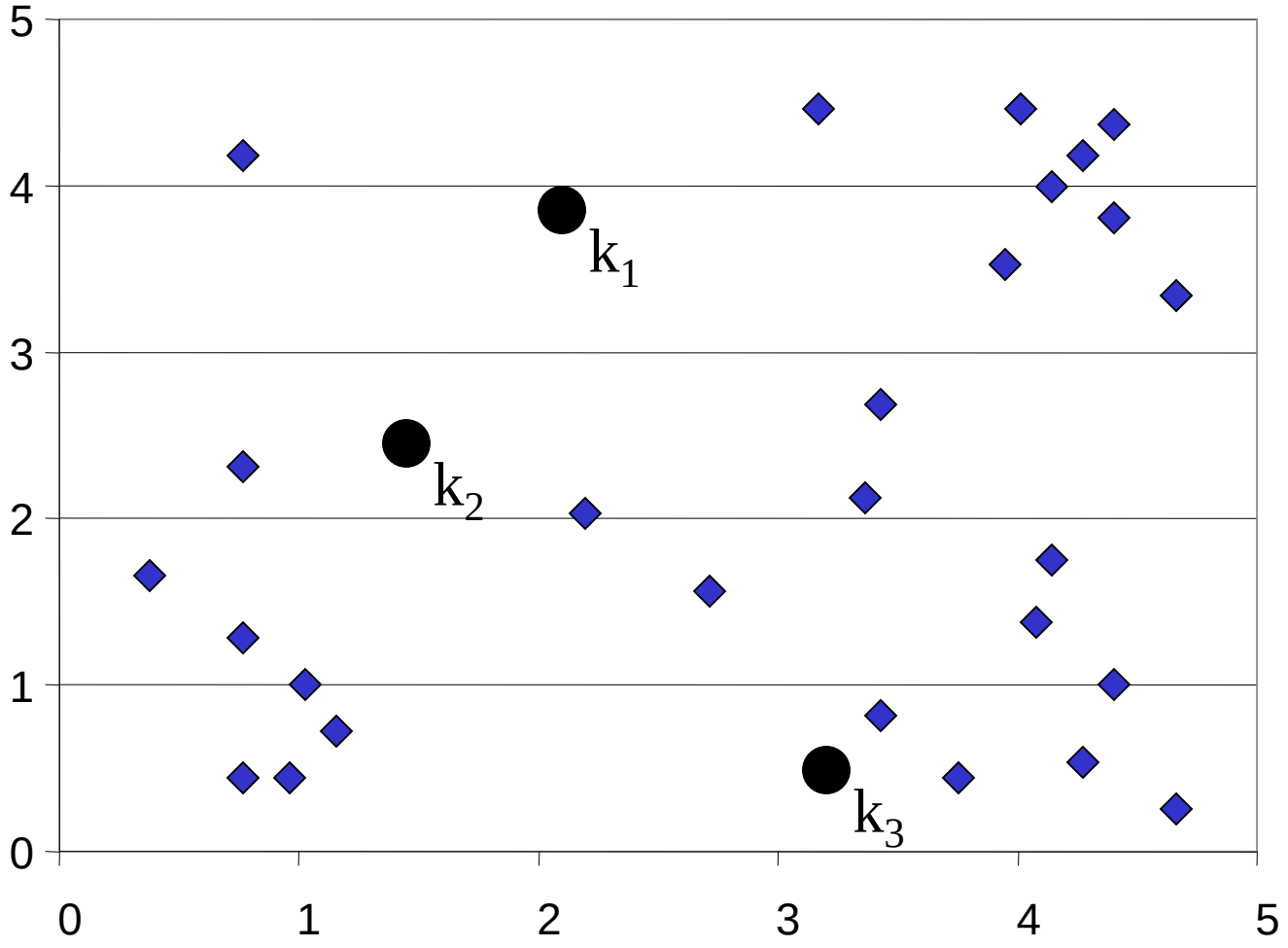
b. Cada objeto da base de dados é avaliado e associado ao grupo mais similar, baseado em alguma medida de distância entre o objeto e os centroides;

c. Um novo centroide para cada grupo é computado pela média dos objetos aos quais os respectivos centroides foram considerados similares;

d. O processo pára quando um critério de convergência é atingido

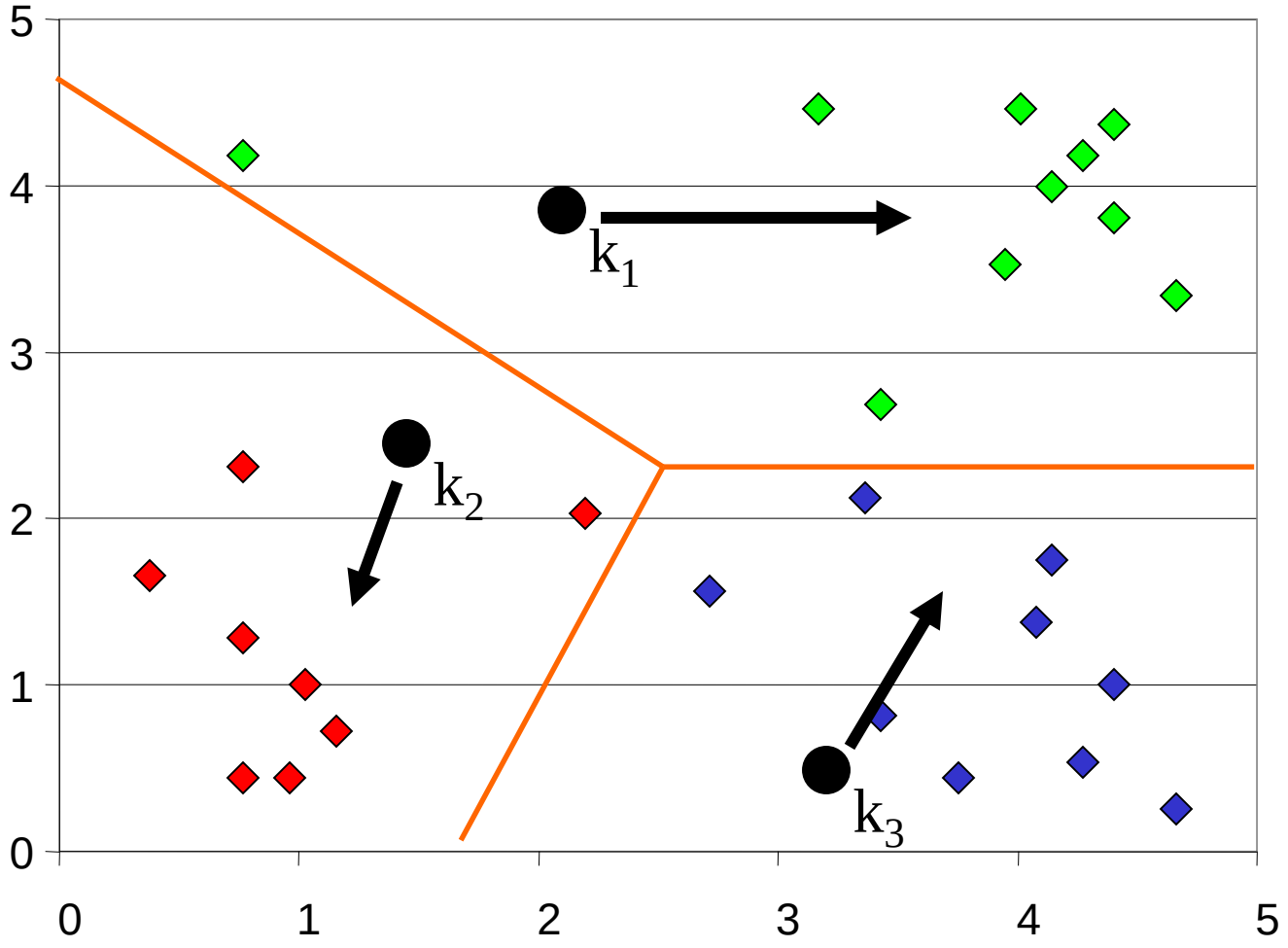
Agrupamento por K-means: Passo 1

Algoritmo: k-means, Métrica de Distância: Distância Euclidiana



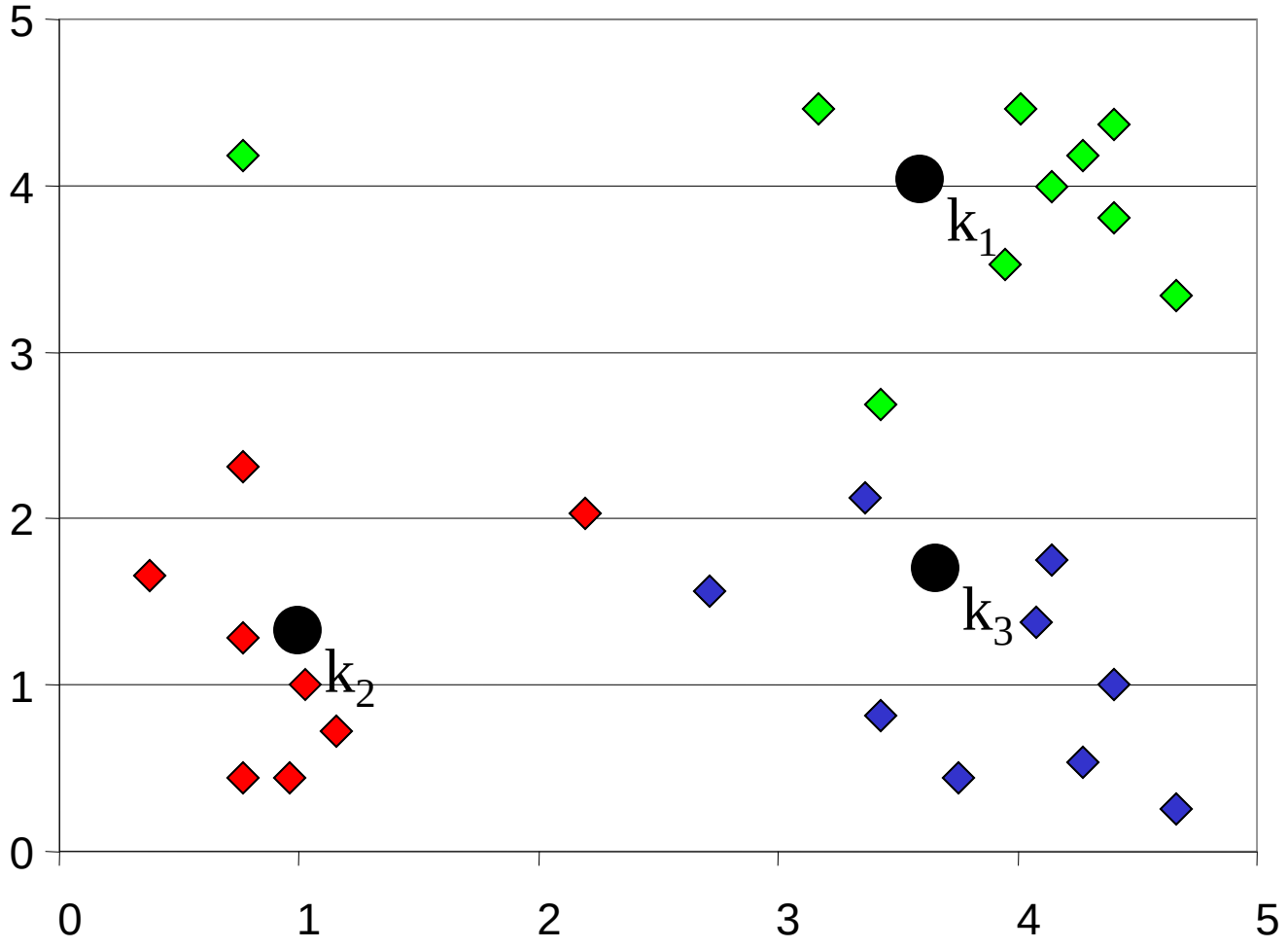
Agrupamento por K-means: Passo 2

Algoritmo: k-means, Métrica de Distância: Distância Euclidiana



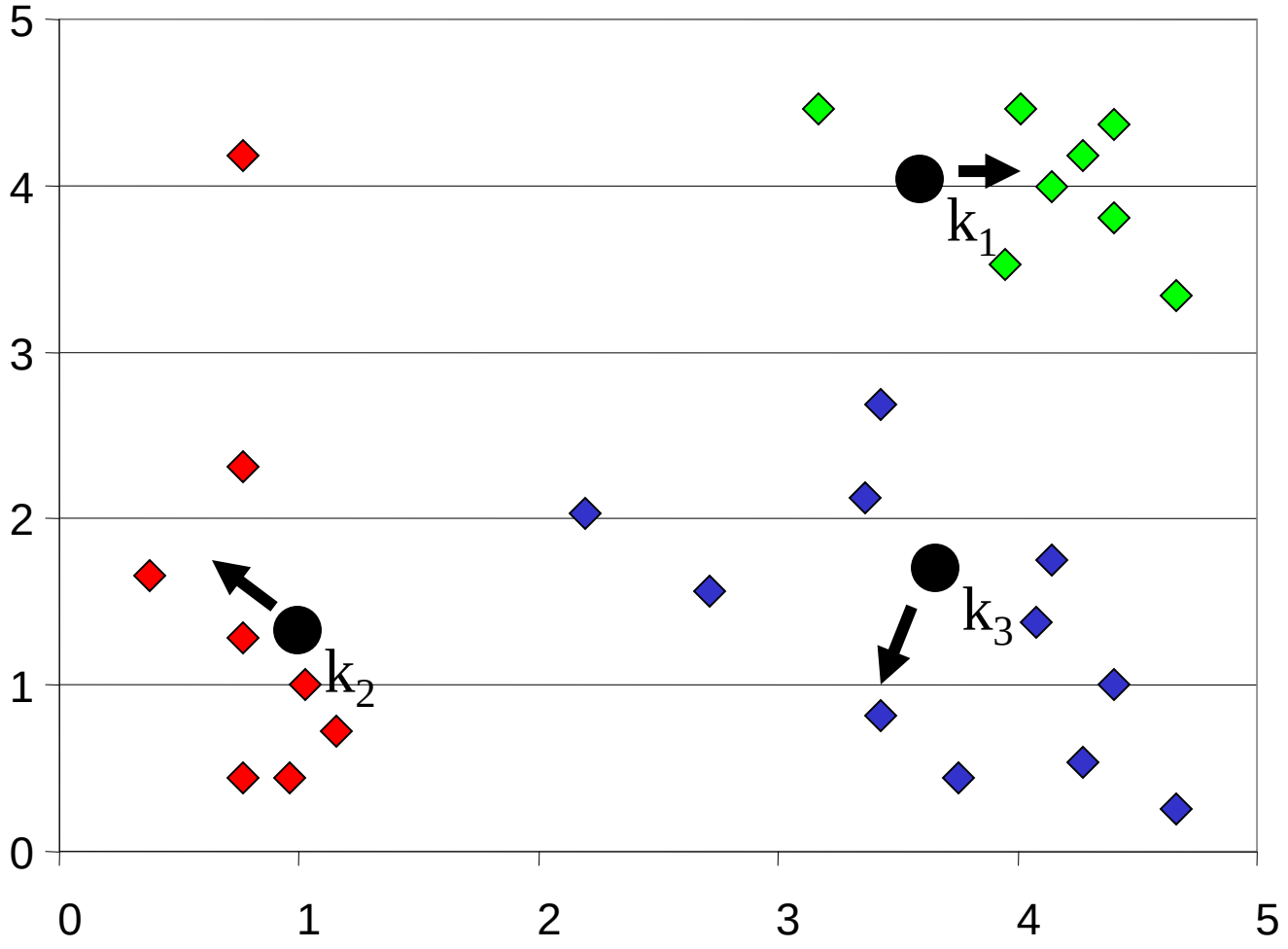
Agrupamento por K-means: Passo 3

Algoritmo: k-means, Métrica de Distância: Distância Euclidiana



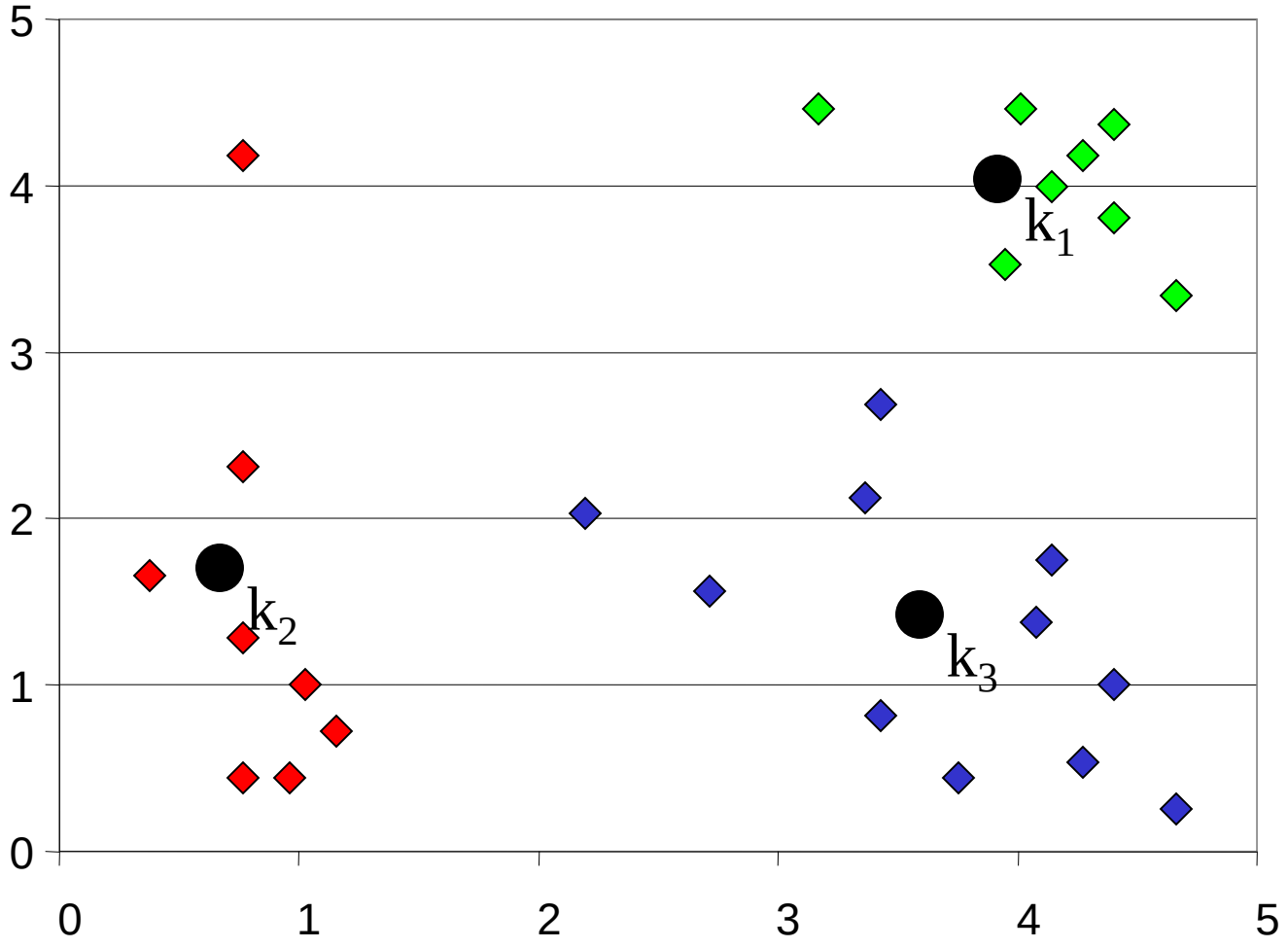
Agrupamento por K-means: Passo 4

Algoritmo: k-means, Métrica de Distância: Distância Euclidiana



Agrupamento por K-means: Passo 4

Algoritmo: k-means, Métrica de Distância: Distância Euclidiana



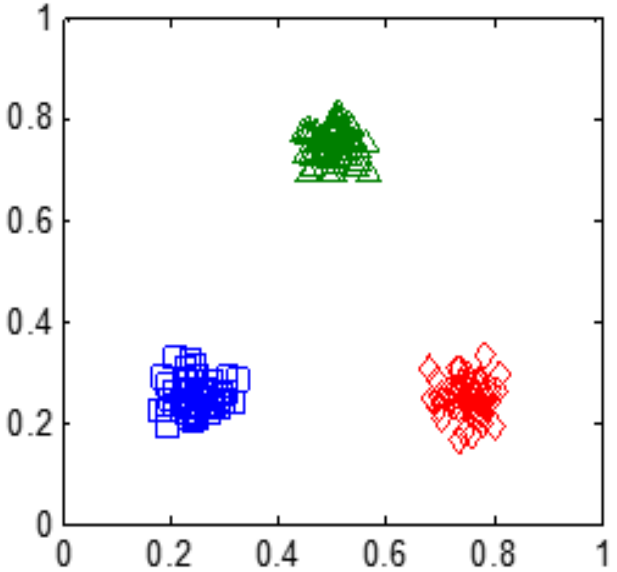
Comentários sobre o Método *K-Means*

- Pontos fortes
  - *Relativamente eficiente*: *O*(*tkn*), na qual *n* é o # de objetos, *k* é o # de *clusters*, e *t é o* # iterações
  - *Simples de entender, fácil de implementação e rápida convergência*

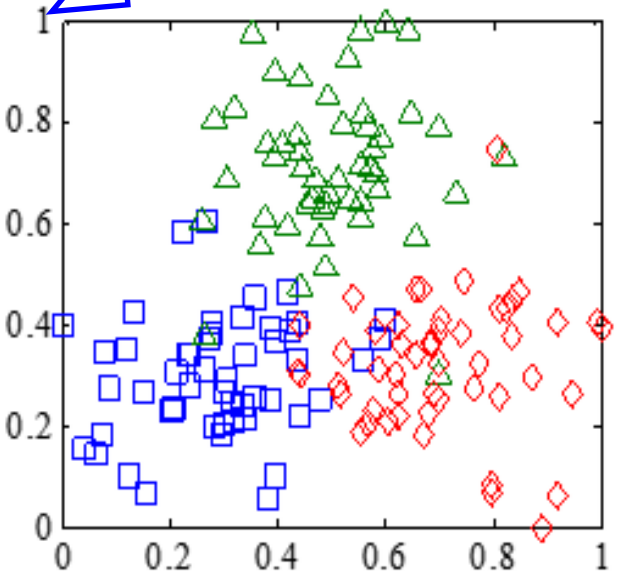
- Pontos fracos
  - *Frequentemente termina em um ótimo local (não explora o espaço de soluções)*
  - *Sensível à inicialização dos centroides*
  - *É necessário especificar k, o número de clusters, a priori*
  - *Incapaz de lidar com outliers*

Abordagem nebulosa

- **Grupos sobrepostos**; não há grupos naturais (definição de Carmichael, 1968)



(a)¤



(b)¤

Abordagem nebulosa

a. Abordagem tradicional

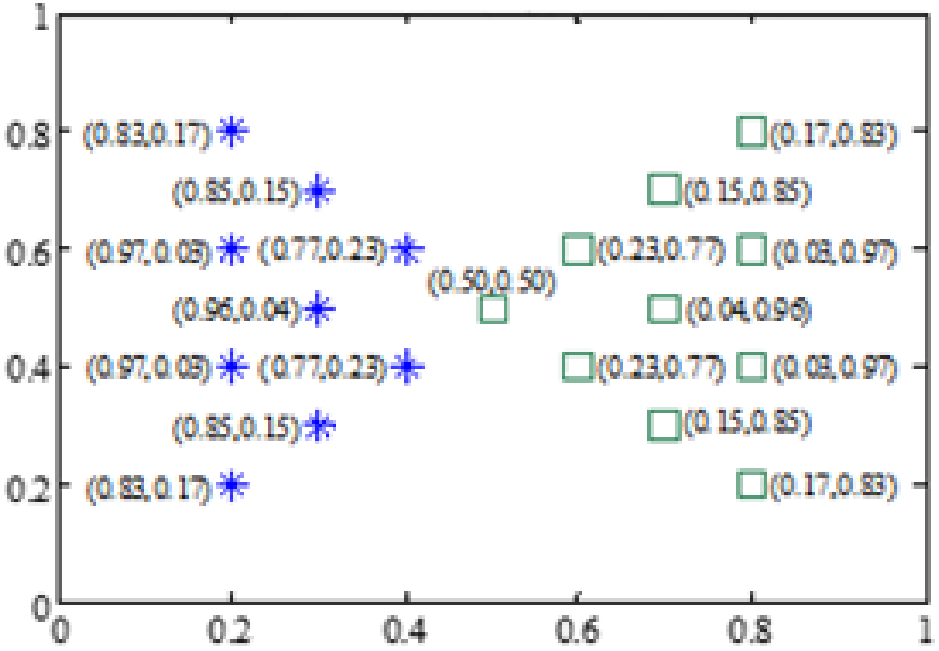
- Cada objeto (i) pertence a um único grupo (j): $\mu_{ij} = \{0,1\}$
- Os objetos contribuem, com o mesmo peso, na atualização dos protótipos

b. Abordagem nebulosa

- Todos os objetos pertencem a todos os grupos, simultaneamente, variando o grau de pertinência: $\mu_{ij} = [0,1]$
- Os objetos contribuem parcialmente na atualização dos protótipos

Abordagem nebulosa

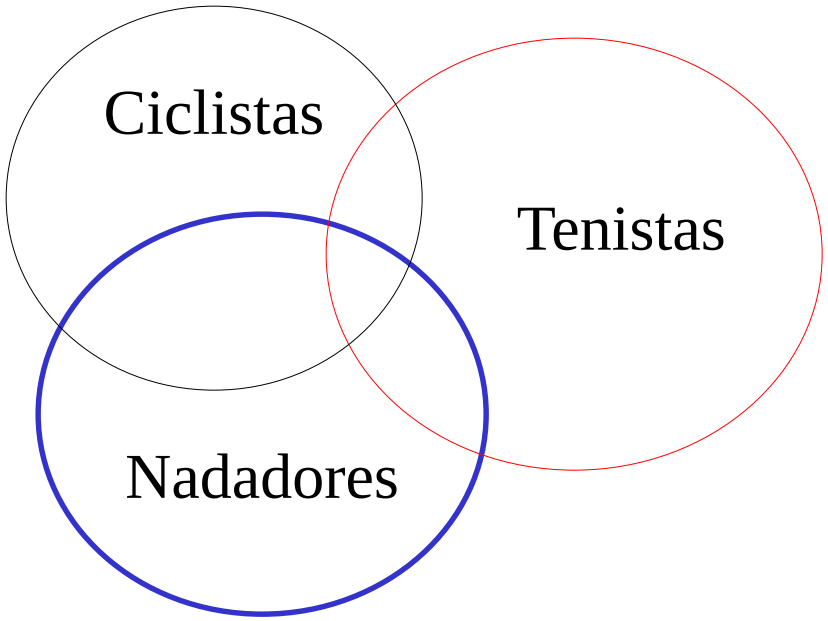
- Quanto mais próximo o objeto estiver do centroide, maior o grau de pertinência dele a esse grupo



- O grau de pertinência é uma distância normalizada; quanto mais próximo o objeto do grupo, maior é o grau de pertinência desse objeto a esse grupo

Abordagem nebulosa

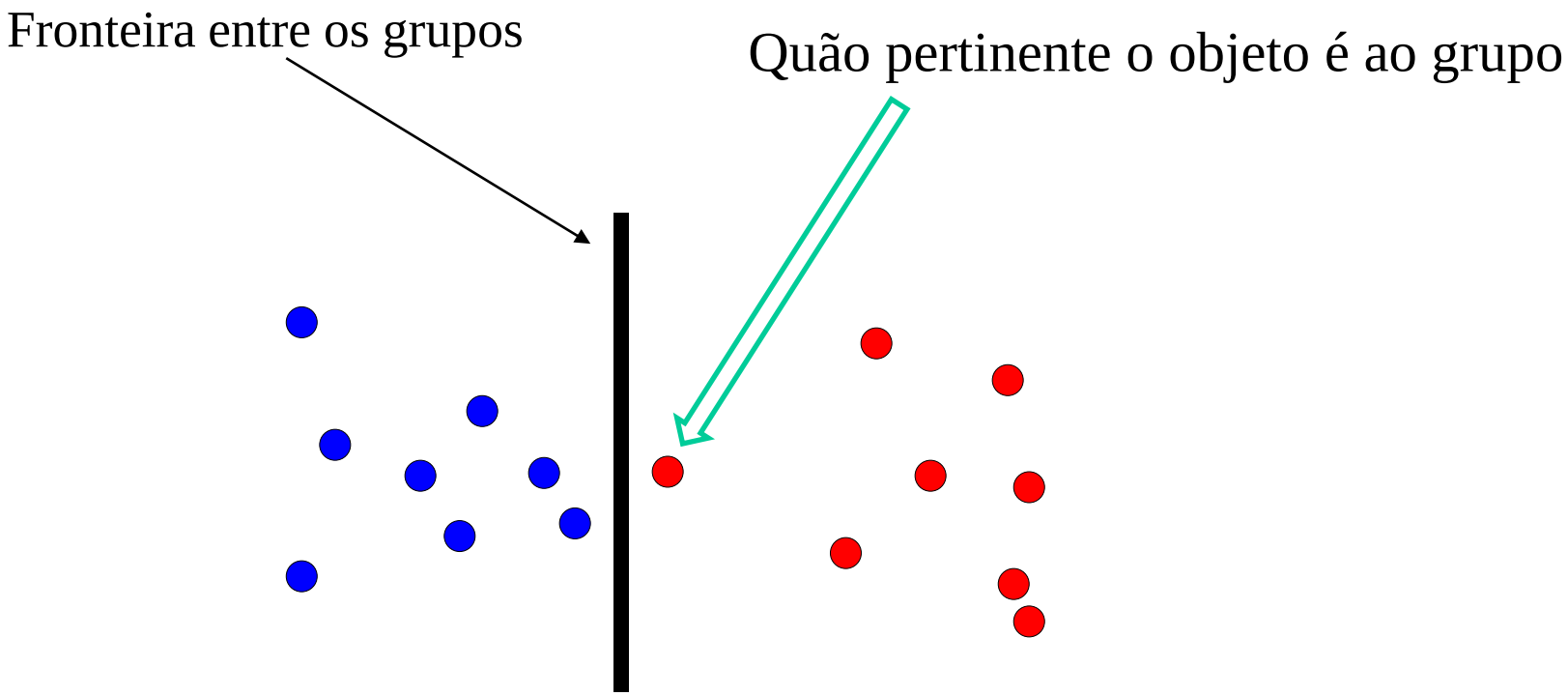
- **Quais as vantagens da abordagem nebulosa?**
  - *Objetos podem pertencer a mais de um grupo, simultaneamente*
    - Ex.: Base de dados de atletas



Abordagem nebulosa

Quais as vantagens da abordagem nebulosa?

– *Quantificar a pertinência de objetos a grupos*

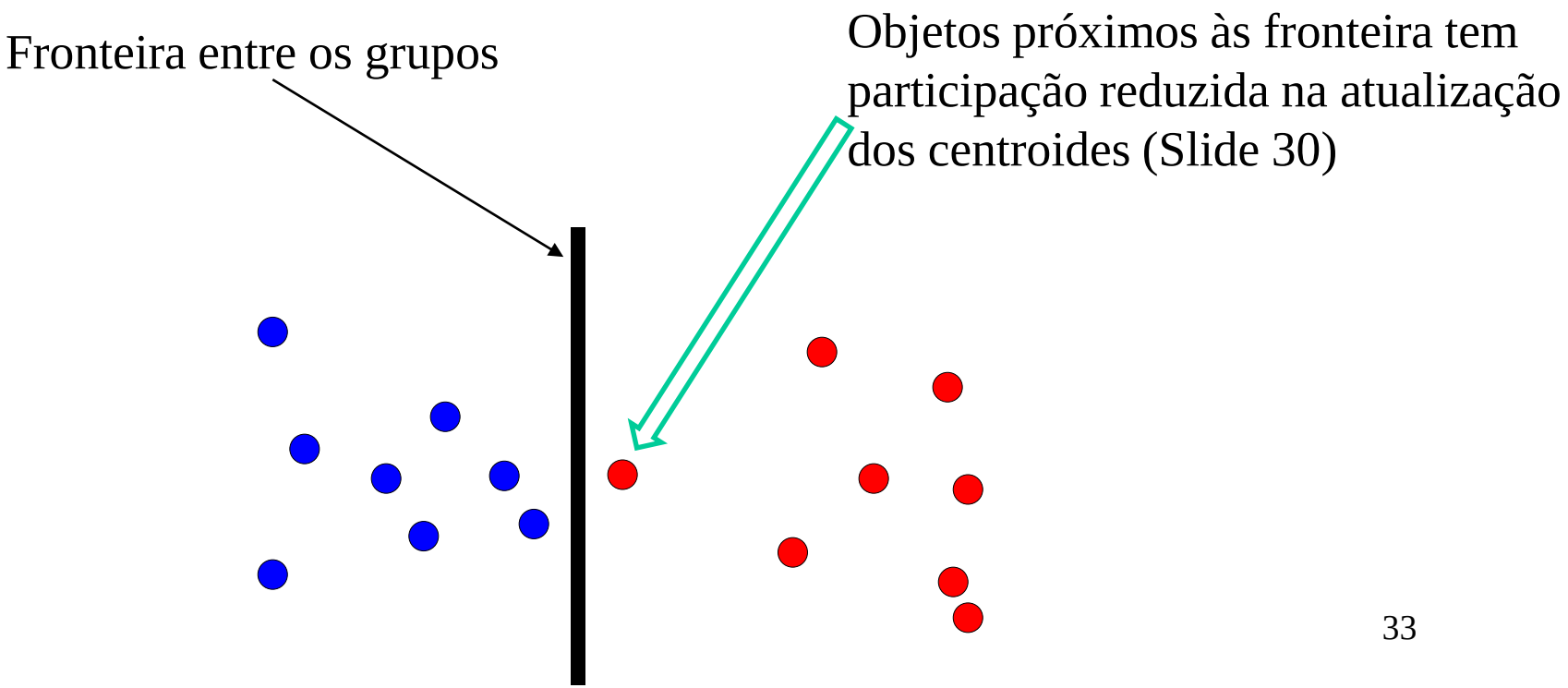


Abordagem nebulosa

Quais as vantagens da abordagem nebulosa?

– *Minimizar a participação de objetos ruidosos na atualização dos protótipos*

(minimiza a classificação incorreta e maximiza a representatividade dos protótipos aos respectivos grupos)



Abordagem nebulosa

- **Como atribuir diferentes graus de pertinência aos objetos?**

  - É usado o parâmetro $m(1,\infty)$, conhecido como *expoente de ponderação*, ou *fuzzificador*

  - É comum o intervalo [1,25, 2,00]

  - O *m* é usado para aumentar ou para reduzir a pertinência (importância) de objetos aos grupos. Assim, quanto maior o valor de *m*, mais sobrepostos serão so grupos. Portanto, aumentar o valor para *m* significa reduzir a fronteira entre os grupos

Abordagem nebulosa

- **Como atribuir diferentes graus de pertinência aos objetos?**
  - $\mu_{ij} = \frac{1}{\sum_{p=1}^{k}\left(\frac{d_{ij}}{dip}\right)^{\frac{2}{m-1}}}$, pertinência do objeto i ao centroide j
  - d_{ij}, distância do objeto i ao centroide j
  - k é a quantidade de centroides
  - d_{ip}, distância do objeto i a todos os centroides
  - $\mu_{ij} \in \mathbf{U}, \mathbf{U}$ é a matriz de pertinências de tamanho *N*(objetos) x *K*(centroides)

Abordagem nebulosa

- **Qual deve ser o valor para *m*?**

  - O valor de *m* depende das características do problema; quanto mais sobrepostos forem os grupos, maior deverá ser o valor para *m*

  - Se *m* for muito elevado, todos os objetos contribuirão com a mesma importância para a atualização dos protótipos. Com isso, o grau de pertinência (μ_{ij}) tenderá a $1/k$ (k = número de grupos), tornando homogênea a matriz de pertinências, ocasionando sobreposição dos protótipos; veja a equação no slide anterior

Abordagem nebulosa

- **Qual deve ser o valor para *m*?**

  - O valor de *m* deve ser tal que os objetos próximos à fronteira entre os grupos tenham sua participação reduzida na atualização dos protótipos, de modo a minimizar a classificação incorreta e maximizar a representatividade dos protótipos aos respectivos grupos

(Veja a Figura dos slides 28 e 33)

O algoritmo c-Médias Nebuloso (ou Fuzzy c-Means)

- É um dos algoritmos clássicos da literatura de agrupamento *fuzzy*. É a versão nebulosa do k-Médias

- E caracterizado por ponderar iterativamente a contribuição dos objetos na atualização dos protótipos. **Os protótipos são atualizados pela média ponderada da posição de todos os objetos da base de dados**, ou seja, todos os objetos contribuem parcialmente na atualização da posição dos protótipos

O algoritmo c-Médias Nebuloso (ou Fuzzy c-Means)

- O algoritmo consiste em atualizar iterativamente a matriz de pertinências (**U**) e a posição dos protótipos (**C**), respectivamente:

$$\mu_{ij} = \frac{1}{\sum_{p=1}^{k} \left(\frac{d_{ij}}{dip} \right)^{\frac{2}{m-1}}}$$

sendo *k* o total de centroides.

$$\mathbf{C}_j = \frac{\sum_{i=1}^{n} \mathbf{X}_i \left(\mu_{ij} \right)^m}{\sum_{i=1}^{n} \left(\mu_{ij} \right)^m}$$

sendo $\mathbf{X}_i$ o objeto de índice *i*

O algoritmo c-Médias Nebuloso (ou Fuzzy c-Means)

- O que se deseja é a minimização de uma função-objetivo

$$J_{fcm} = \sum_{j=1}^{k} \sum_{i=1}^{n} \mu_{ij}^{m} \; {d_{ij}}^{2}$$

- Sendo:
  - *k* o número total de protótipos;
  - *n* o número total de objetos na base de dados;
  - *m* o expoente de ponderação;
  - d_{ij} a distância do objeto *i* ao protótipo *j*; e
  - $\mu_{ij} \in \mathbf{U}$ é o grau de pertinência do objeto *i* ao grupo *j*, indicando quão representativo (pertinente) o objeto é para este grupo em relação aos demais grupos

O algoritmo c-Médias Nebuloso (ou Fuzzy c-Means)

- A função-objetivo possui as seguintes restrições:

1. $\mu_{ij} \in [0,1]$

2. $\sum_{j=1}^{k} \mu_{ij} = 1,\ i = 1, \ldots, n$

   A restrição acima garante que a soma das pertinências de um objeto a todos os grupos seja igual a um

3. $0 < \sum_{i=1}^{n} \mu_{ij} < n$

   A restrição acima garante que cada grupo tenha pelo menos um objeto com grau de pertinência maior do que zero

O algoritmo c-Médias Nebuloso (ou Fuzzy c-Means)

Parâmetros de Entrada:

- **X** //base de dados $X = \{x_1, x_2, \ldots, x_n\}$
- *k* //número de protótipos
- *m* //expoente de ponderação

Parâmetros de Saída:

- **U** //matriz de pertinências $U = \{\mu_{11}, \ldots, \mu_{n,k}\}$
- **C** //protótipos resultantes $C = \{c_1, c_2, \ldots, c_k\}$
- *cf* //valor da função de custo
- *iter* //iterações necessárias para a convergência

1. **U** = Initialize_U(*n*,*k*,*m*)
2. *old_cf* = 0
3. *stop* = 0
4. while !*stop*
5. *iter* = *iter* + 1
6. **C** = Compute_Prototypes (**X**,**U**,*m*)
7. **U** = Update_U(**X**,**C**,*m*)
8. *cf* = Calculate_Cost(**X**,**C**,**U**,*m*)
9. *stop* = Check_Stopping_Criterion(*cf*,*old_cf*,*iter*)
10. *old_cf* = *cf*
11. end while