



Universidade do Minho
Departamento de Informática

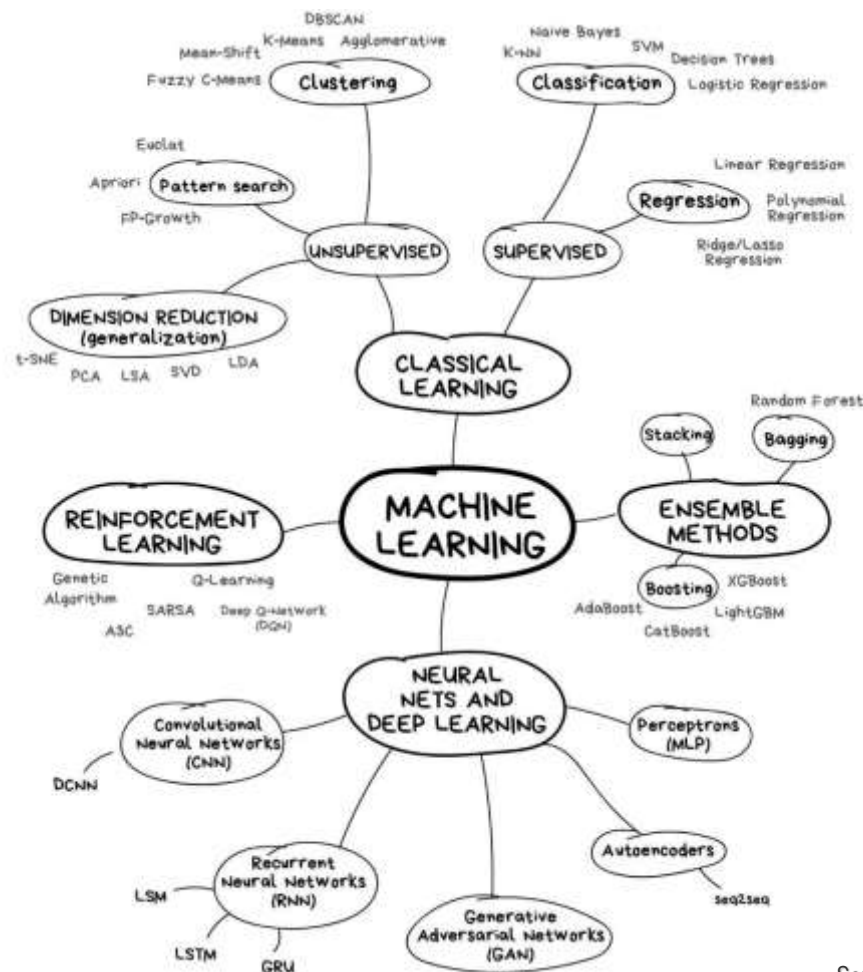
APRENDIZAGEM E DECISÃO INTELIGENTES

**LEI/MiEI @ 2022/2023, 2º sem
[ADI³]**

- Aprendizagem não Supervisionada
- Segmentação (*Clustering*)
 - Tipos de dados para análise
- Métodos de Segmentação
 - Particionamento
 - Hierarquização
 - Outros
- Algoritmos de Segmentação



Aprendizagem Automática (Machine Learning)



Source: The map of the machine learning world
Vasily Zubarev (vas3k.com)



Universidade do Minho
Departamento de Informática

Aprendizagem não Supervisionada



Definição de Aprendizagem

aprendizagem

aprendizagem | *n. f.*

a·pren·di·za·gem

(francês *apprentissage*)

nome feminino

1. Acto ou efeito de aprender.
2. Tempo durante o qual se aprende.
3. Experiência que tem quem aprendeu.

aprendizagem automática

• [Portugal] • [Informática] Disciplina central da inteligência artificial que se dedica ao estudo e ao desenvolvimento de algoritmos e programas que permitem dotar o computador da capacidade de efectuar determinada tarefa sem recurso à intervenção humana (por exemplo, classificação de *emails* como lixo, reconhecimento de voz, etc.). [Equivalente no português do Brasil: aprendizado automático.]



Definição de Decisão

decisão

decisão | *n. f.*

de·ci·são

(latim *decisio*, -*onis*, diminuição, enfraquecimento, solução, compromisso)

nome feminino

1. Acto ou efeito de decidir.
2. Resolução tomada após discussão ou exame prévio. = DELIBERAÇÃO, JUÍZO, SENTENÇA
3. Capacidade para decidir ou resolver algo. = CORAGEM, INTREPIDEZ ≠ COVARDIA
4. [Teologia] Prescrição sobre matéria de fé ou de dogma.

Palavras relacionadas: [contencioso](#), [meias-medidas](#), [sentença](#), [recurso](#), [jurisprudência](#), [conciliação](#), [decisório](#).

PLUS



Aprendizagem Automática

**Aprendizagem
por Reforço**
Reinforcement Learning

**Aprendizagem
Automática**
Machine Learning

**Aprendizagem
com Supervisão**
Supervised Learning

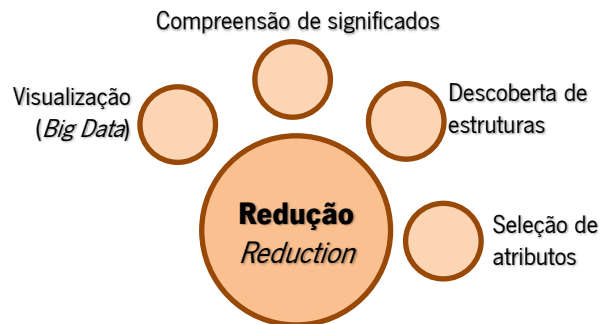
**Aprendizagem
sem Supervisão**
Unsupervised Learning

**Aprendizagem
por Reforço**
Reinforcement Learning

**Aprendizagem
Automática**
Machine Learning

**Aprendizagem
com Supervisão**
Supervised Learning

Aprendizagem Automática



**Aprendizagem
sem Supervisão**
Unsupervised Learning



Aprendizagem Automática não Supervisionada

**Aprendizagem
Automática**
Machine Learning

- Paradigma de computação em que a característica essencial do sistema se revela pela sua capacidade de aprender de modo autónomo e independente;

**Aprendizagem
com Supervisão**
Supervised Learning

- Paradigma de aprendizagem em que os casos que se usam para aprender contêm informação acerca dos resultados pretendidos, sendo possível estabelecer uma relação entre os valores pretendidos e os valores produzidos pelo sistema;

**Aprendizagem
sem Supervisão**
Unsupervised Learning

- Paradigma de aprendizagem em que não são conhecidos resultados sobre os casos, apenas os enunciados dos problemas, tornando necessário a escolha de técnicas de aprendizagem que avaliem o funcionamento interno do sistema;

**Aprendizagem
por Reforço**
Reinforcement Learning

- Paradigma de aprendizagem que, apesar de não ter informação sobre os resultados pretendidos, permite efetuar uma avaliação sobre se os resultados produzidos são bons ou maus.

Aprendizagem sem Supervisão

Aprendizagem sem Supervisão

Unsupervised Learning

- Paradigma de aprendizagem em que não são conhecidos resultados sobre os casos, apenas os enunciados dos problemas, tornando necessário a escolha de técnicas de aprendizagem que avaliem o funcionamento interno do sistema;
 - A aprendizagem não supervisionada significa que existem dados de entrada (x) mas não existem os correspondentes resultados;
 - O objetivo deste tipo de aprendizagem é o de modelar a estrutura ou a distribuição dos dados do problema;
 - São, normalmente, divididos em duas categorias:
 - Segmentação: quando se pretende organizar os dados em grupos coerentes (agrupar clientes que comprem bebidas açucaradas);
 - Associação: quando se pretende conhecer regras que associem o comportamento demonstrado pelos dados (pessoas que comprar bebidas açucaradas não compram bebidas alcoólicas):





Universidade do Minho
Departamento de Informática



Segmentação

Clustering

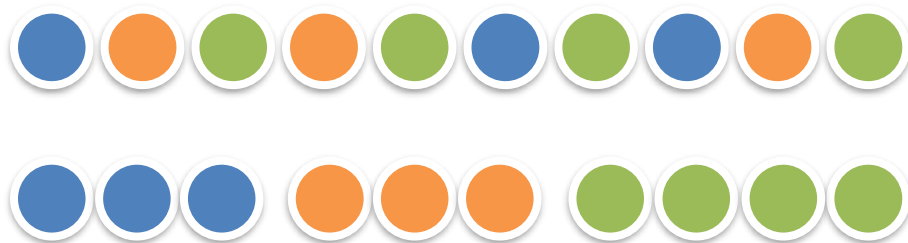
O que é Segmentação?

- A Segmentação/ *Clustering* de dados é um processo através no qual se **particiona** um conjunto de **dados em segmentos/ clusters** de menor dimensão, que agrupam conjuntos de dados **similares**.



O que é Segmentação?

- A Segmentação/ *Clustering* de dados é um processo através no qual se **particiona** um conjunto de **dados em segmentos/ clusters** de menor dimensão, que agrupam conjuntos de dados **similares**.



O que é Segmentação?

- A Segmentação/ *Clustering* de dados é um processo através no qual se **particiona** um conjunto de **dados em segmentos/ *clusters*** de menor dimensão, que agrupam conjuntos de dados **similares**.



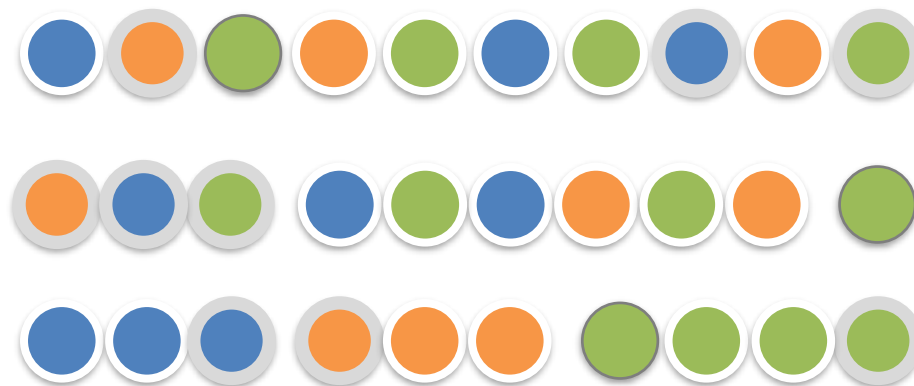
O que é Segmentação?

- Um Segmento/ *Cluster* é uma coleção de valores/objetos que:
 - são similares entre si, dentro de um mesmo segmento;
 - são diferentes dos valores/objetos de outros segmentos:



O que é Segmentação?

- Um Segmento/ *Cluster* é uma coleção de valores/objetos que:
 - são similares entre si, dentro de um mesmo segmento;
 - são diferentes dos valores/objetos de outros segmentos:



- Medidas de similaridade:
 - distância Euclidiana ou de Manhattan, para atributos contínuos;
 - coeficiente de Jacqard, para atributos discretos/binários;
 - etc.

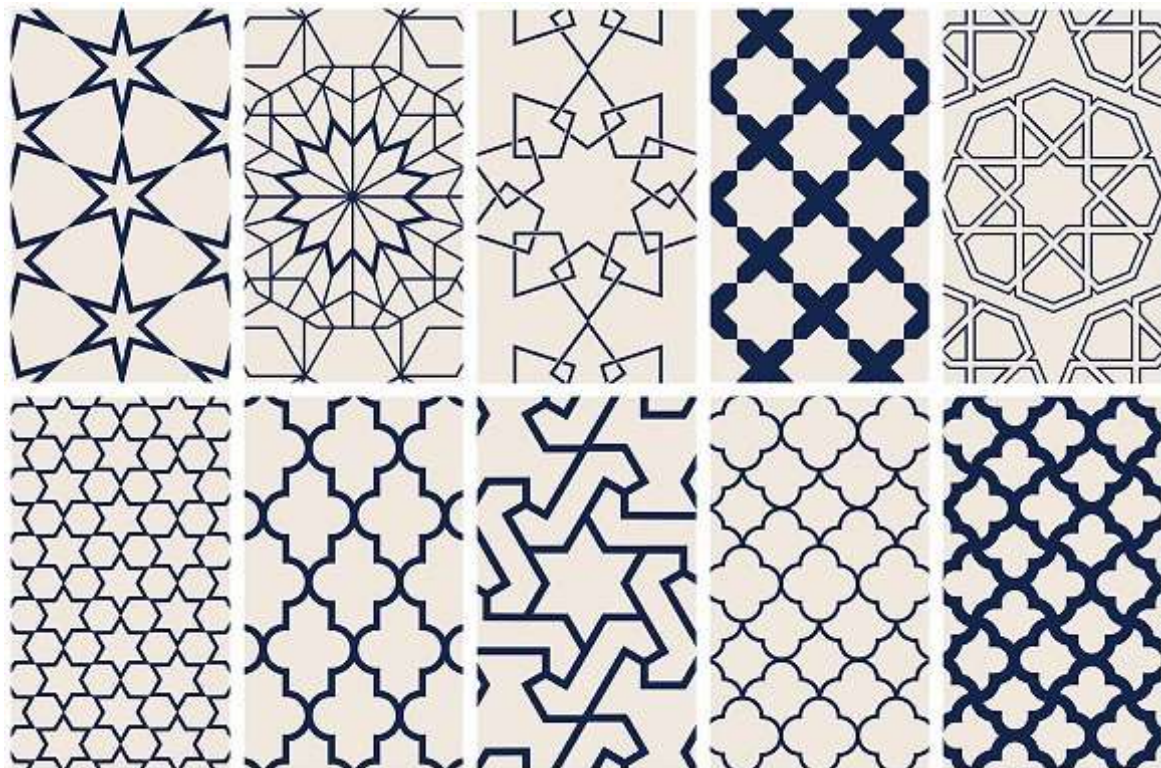
Aplicações da Segmentação?

- Como uma ferramenta *per si*, para pesquisar “dentro” dos dados, sobre a distribuição dos seus valores;
- Como uma das fases do pré-processamento, por forma a organizar os dados a submeter a outros algoritmos;
- Em problemas de reconhecimento de padrões (*pattern matching*);
- No processamento de imagem;
- Na pesquisa em mercados económicos;
- etc.



Utilização da Segmentação?

- A detecção de segmentos é útil:
 - quando se suspeita da **existência de agrupamentos** “naturais”, que podem representar grupos de clientes, de produtos ou de bens que partilhem (muita) informação;
 - quando existam **muitos padrões diferentes** nos dados, dificultando a tarefa de identificar um determinado padrão;
a criação de segmentos semelhantes reduz a complexidade do problema.

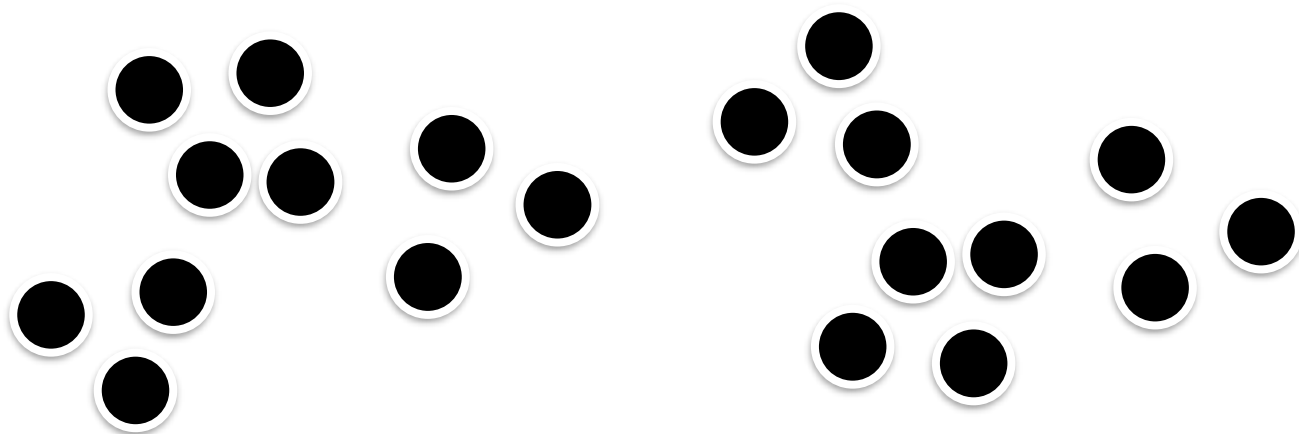


Exemplos de aplicação

- *Marketing*:
 - ajuda na descoberta de grupos de clientes para desenvolver estratégias de comercialização;
- Previsão de sismos:
 - a observação de epicentros sísmológicos permite identificar segmentos comuns de falhas continentais;
- Seguradoras:
 - identificação de grupos de utentes que representam maior risco de contratação;
- Banca:
 - identificação de categorias de clientes (económicas, sociais, etc.).

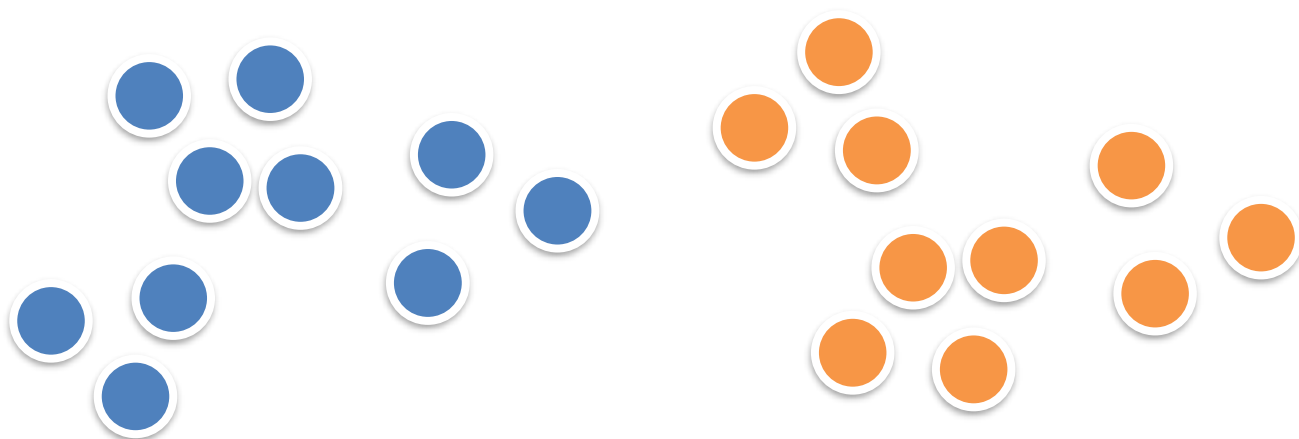


- A noção de segmento é ambígua:



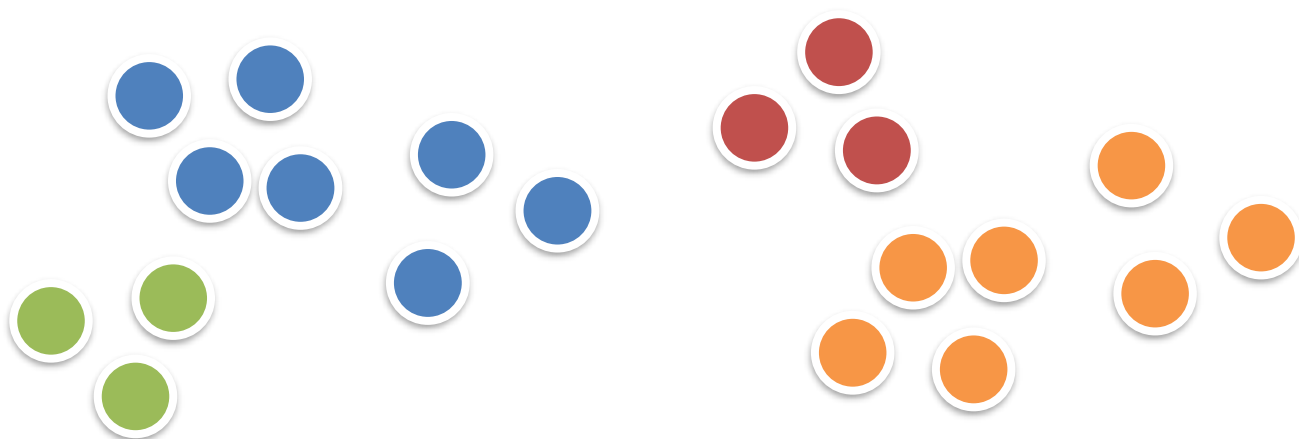
Pontos iniciais

- A noção de segmento é ambígua:



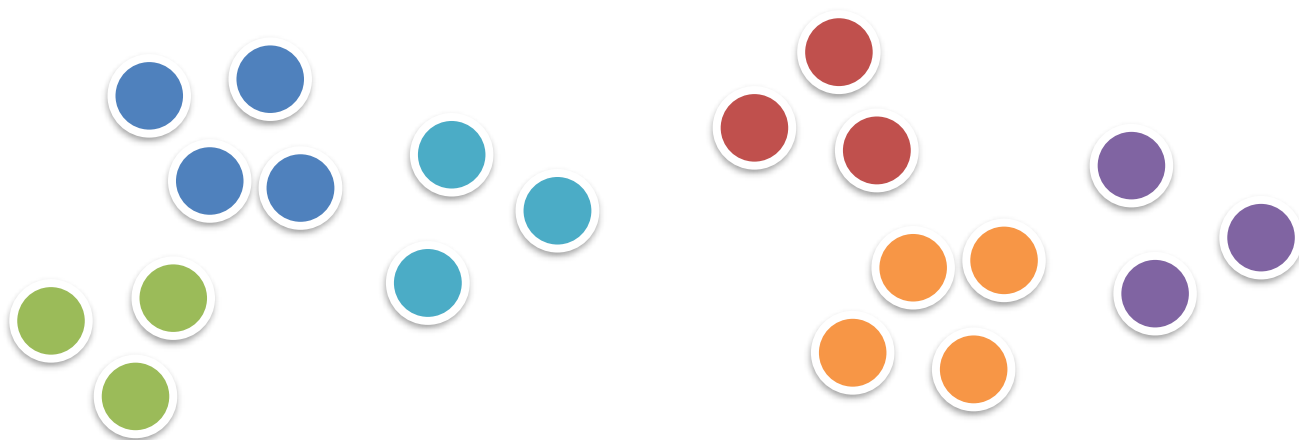
2 segmentos

- A noção de segmento é ambígua:



4 segmentos

- A noção de segmento é ambígua:



6 segmentos

Tipos de dados para análise

- Matriz de dados: representa 'n' objetos com 'p' atributos;

X_{11}	...	X_{1j}	...	X_{1p}
...	
X_{i1}	...	X_{ij}	...	X_{ip}
...	
X_{n1}	...	X_{nj}	...	X_{np}

- Matriz de distâncias: mede a proximidade entre pares de objetos;
- Tanto mais similar quanto mais próximo de 0 (zero).

0				
$d(2,1)$	0			
$d(3,1)$...	0		
...	0	
$d(n,1)$	$d(n,2)$	0

Tipos de dados para análise

- Atributos contínuos;
- Atributos binários;
- Atributos nominais;
- Atributos ordinais;
- Atributos mistos.



Tipos de dados para análise

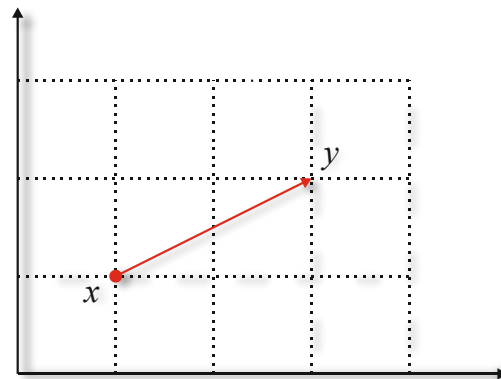
- Atributos contínuos:
 - normalizar os dados: evita que os resultados dependam das unidades de medida;
 - normalmente, utilizam-se medidas de distância para calcular a proximidade (similaridade) entre objetos;

Tipos de dados para análise

- Atributos contínuos:

- normalizar os dados: evita que os resultados dependam das unidades de medida;
- normalmente, utilizam-se medidas de distância para calcular a proximidade (similaridade) entre objetos:
 - distância Euclidiana: é a medida de distância geométrica no espaço (a mais usada):

$$d(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (\text{para 2 dimensões})$$



Tipos de dados para análise

■ Atributos contínuos:

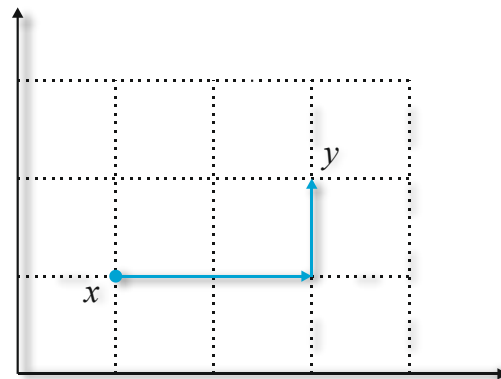
- normalizar os dados: evita que os resultados dependam das unidades de medida;
- normalmente, utilizam-se medidas de distância para calcular a proximidade (similaridade) entre objetos:

- distância Euclidiana: é a medida de distância geométrica no espaço (a mais usada):

$$d(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (\text{para 2 dimensões})$$

- distância *Manhattan*: mede a distância pela diferença entre os pontos (função não quadrática):

$$d(x, y) = |x_1 - x_2| + |y_1 - y_2| \quad (\text{para 2 dimensões})$$



■ Atributos contínuos:

- normalizar os dados: evita que os resultados dependam das unidades de medida;
- normalmente, utilizam-se medidas de distância para calcular a proximidade (similaridade) entre objetos:

- distância Euclidiana: é a medida de distância geométrica no espaço (a mais usada):

$$d(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (\text{para 2 dimensões})$$

- distância *Manhattan*: mede a distância pela diferença entre os pontos (função não quadrática):

$$d(x, y) = |x_1 - x_2| + |y_1 - y_2| \quad (\text{para 2 dimensões})$$

- distância *Minkowski*: mede o peso progressivo em função da distância dos pontos:

$$d(i, j) = \left(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 \right)^{1/2} \quad (\text{para 2 dimensões})$$

(é uma generalização das duas anteriores).

$$d(i, j) = \left(\sum_{k=1}^n |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}} \quad (\text{para n dimensões, } c/ \ p \geq 1)$$

Tipos de dados para análise

- Atributos binários:
 - são classificados em:
 - **Simétricos**: significado de ser 0 é o mesmo de ser 1;
 - **Assimétricos**: significado de ser 0 é diferente de ser 1;
 - a similaridade calculada com base em atributos simétricos é designada **similaridade invariante**;
no caso oposto diz-se **similaridade não-invariante**;

Tipos de dados para análise

■ Atributos binários:

- são classificados em:
 - **Simétricos**: significado de ser 0 é o mesmo de ser 1;
 - **Assimétricos**: significado de ser 0 é diferente de ser 1;
- a similaridade calculada com base em atributos simétricos é designada **similaridade invariante**; no caso oposto diz-se **similaridade não-invariante**;
- tabela de contingência para os dados binários:

- coeficiente simples (simétricos):

$$d(i, j) = \frac{b + c}{a + b + c + d}$$

- coeficiente Jaccard (assimétricos):

$$d(i, j) = \frac{b + c}{a + b + c}$$

	Sexo	Febre	Tosse	Dor
João	M	Sim	Não	Não
Maria	F	Sim	Não	Sim
José	M	Sim	Sim	Não

Tipos de dados para análise

■ Atributos binários:

- são classificados em:
 - **Simétricos:** significado de ser 0 é o mesmo de ser 1;
 - **Assimétricos:** significado de ser 0 é diferente de ser 1;
- a similaridade calculada com base em atributos simétricos é designada **similaridade invariante**; no caso oposto diz-se similaridade não-invariante;
- tabela de contingência para os dados binários:

- coeficiente simples (**simétricos**):

$$d(i, j) = \frac{b + c}{a + b + c + d}$$

- coeficiente Jaccard (assimétricos):

$$d(i, j) = \frac{b + c}{a + b + c}$$

	Sexo	Febre	Tosse	Dor
João	M	Sim	Não	Não
Maria	F	Sim	Não	Sim
José	M	Sim	Sim	Não

		Maria		
João	Sexo	M	F	Soma
	M	a = 0	b = 1	a+b
	F	c = 0	d = 0	c+d
	Soma	a+c	b+d	

Tipos de dados para análise

■ Atributos binários:

- são classificados em:
 - Simétricos: significado de ser 0 é o mesmo de ser 1;
 - **Assimétricos**: significado de ser 0 é diferente de ser 1;
- a similaridade calculada com base em atributos simétricos é designada similaridade invariante; no caso oposto diz-se **similaridade não-invariante**;
- tabela de contingência para os dados binários:

- coeficiente simples (simétricos):

$$d(i, j) = \frac{b + c}{a + b + c + d}$$

- coeficiente Jaccard (**assimétricos**):

$$d(i, j) = \frac{b + c}{a + b + c}$$

	Sexo	Febre	Tosse	Dor
João	M	Sim	Não	Não
Maria	F	Sim	Não	Sim
José	M	Sim	Sim	Não

		Maria		
João	F/T/D	S	N	Soma
	S	a = 1	b = 0	a+b
	N	c = 1	d = 1	c+d
	Soma	a+c	b+d	

Tipos de dados para análise

- Atributos nominais:
 - trata-se de uma generalização dos atributos binários, em que os dados podem assumir mais do que 2 valores;
 - Método 1:
 - *matching* simples;
 - $d(i, j) = \frac{n^{\circ} \text{variáveis} - n^{\circ} \text{matches}}{n^{\circ} \text{variáveis}}$
 - Método 2:
 - Utilizar variáveis binárias;
 - Criar uma variável binária para cada valor nominal.

Tipos de dados para análise

- Atributos ordinais:
 - a ordem é relevante:
 - primeiro, segundo, terceiro, ..., penúltimo, último;
 - podem ser tratados como atributos contínuos, sendo que a ordenação dos valores define uma classificação:
 - 1, 2, 3, ..., Máx;
 - as similaridades devem ser calculadas utilizando os mesmos métodos que para os atributos contínuos.

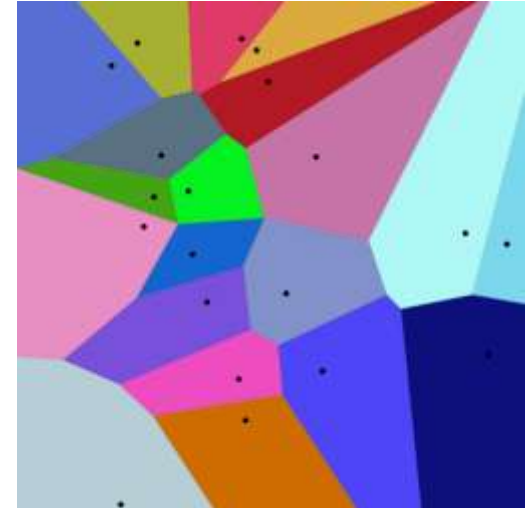
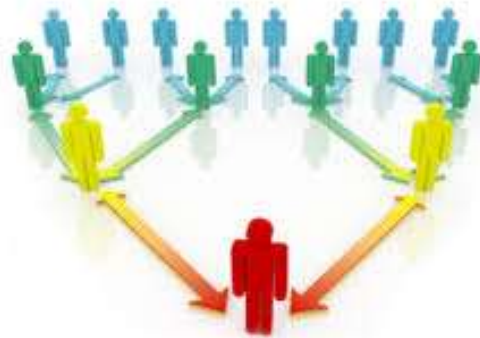
Tipos de dados para análise

- Atributos contínuos;
- Atributos binários;
- Atributos nominais;
- Atributos ordinais;
- Atributos mistos:
 - o conjunto de dados pode conter diversos tipos de atributos;
 - tipicamente, utiliza-se uma função pesada para ponderar e medir os efeitos de cada atributo.



Principais Métodos de Segmentação

- **Particionamento:**
 - criar várias partições e adotar um critério de avaliação;
- **Hierarquização:**
 - decompor hierarquicamente o conjunto de dados;

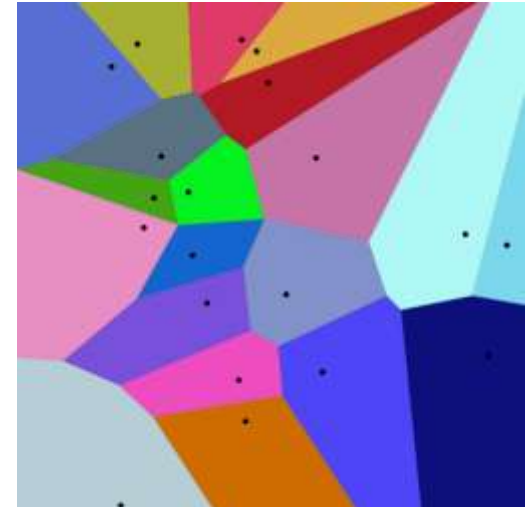


Principais Métodos de Segmentação

- **Particionamento:**
 - criar várias partições e adotar um critério de avaliação;
- **Hierarquização:**
 - decompor hierarquicamente o conjunto de dados;
- **Outros:**
 - Baseados na Densidade:
 - aumentar o segmento enquanto a densidade de pontos estiver num determinado limite (utilizam-se funções de conectividade e densidade);
 - Baseados no Modelo:
 - criar modelos hipotéticos para cada segmento e testar a capacidade de adequação de cada ponto ao segmento.

Algoritmos de Particionamento

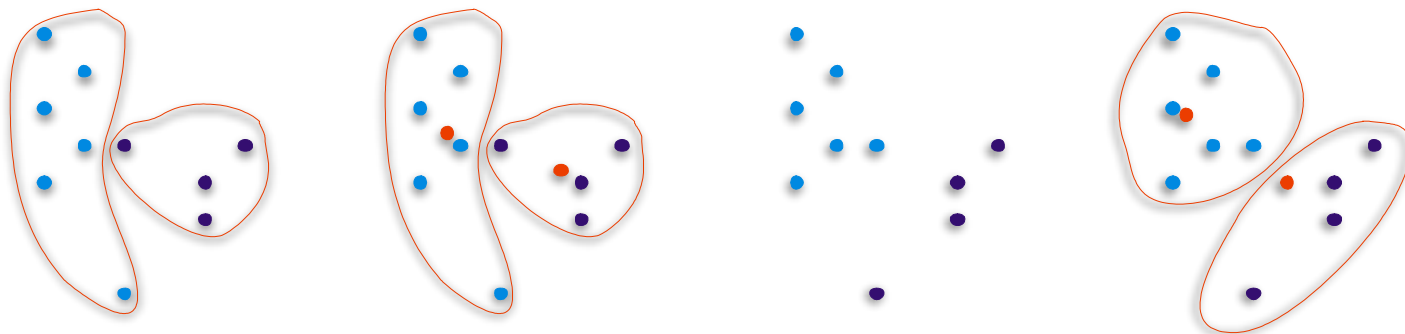
- Particionar um conjunto de dados 'D' contendo 'n' objetos num conjunto de 'k' segmentos/ *clusters*;
- Sendo dado 'k', particionar 'D' em 'k' segmentos de forma a otimizar o critério de particionamento:
 - Ótimo Global: enumeração exaustiva de todas as partições;
 - Métodos heurísticos:
 - k-means: cada segmento é representado pelo **centro** do segmento (centroid);
 - k-medoids: cada segmento é representado por **um dos elementos** do segmento (medoid).



Algoritmos de Particionamento

Método k-means

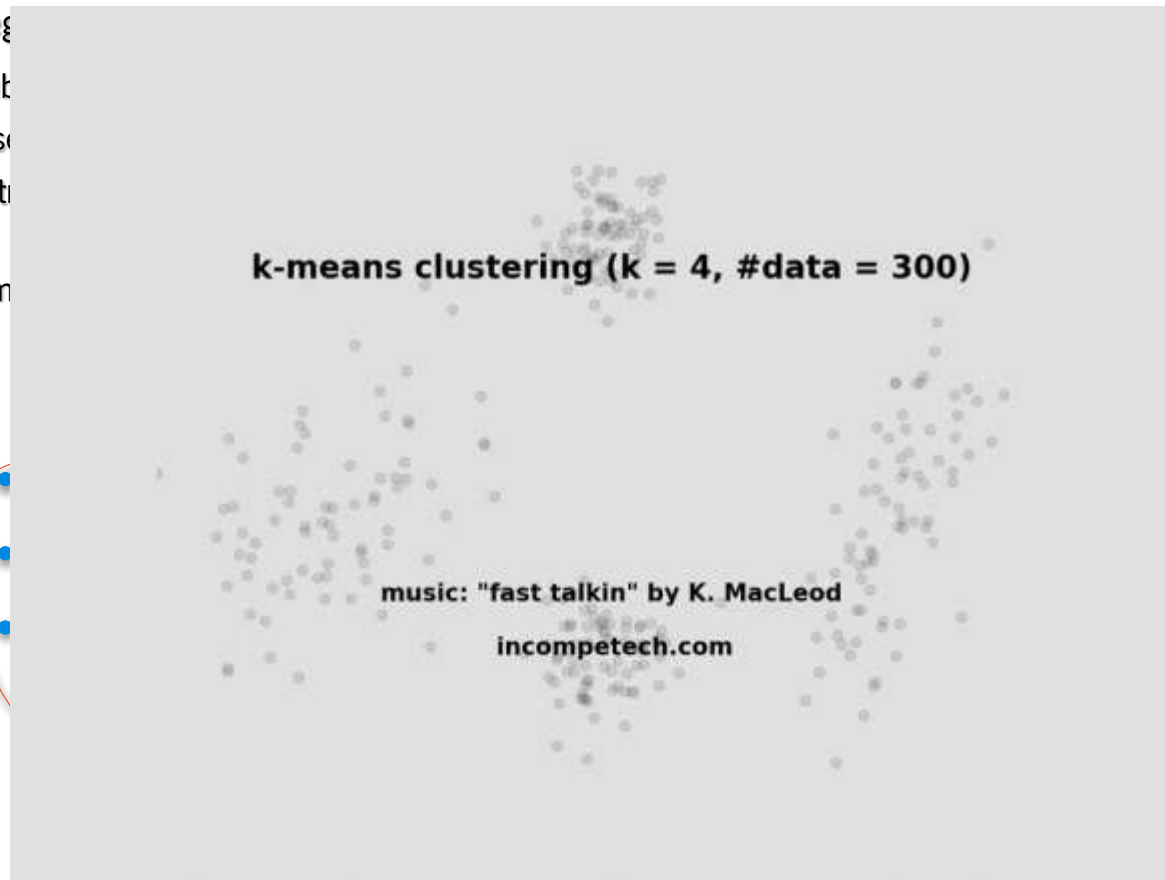
- Sendo dado 'k' (número de segmentos), seguir os 4 passos:
 1. Dividir os objetos em 'k' subconjuntos não vazios;
 2. Calcular o centro de cada segmento (centroid);
 3. Atribuir cada objeto ao centroid mais próximo;
 4. Voltar ao ponto 2.;parar quando não houver mais possibilidades de atribuição.



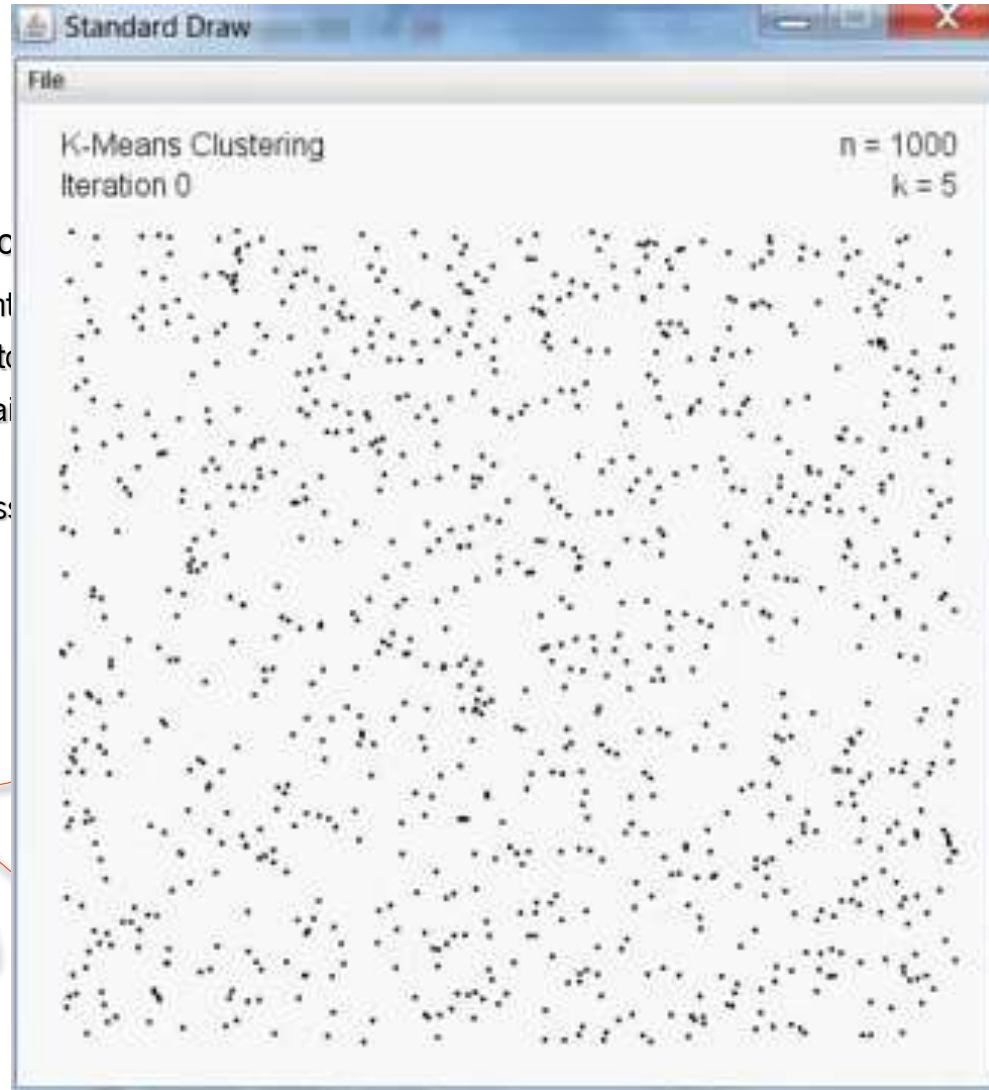
Algoritmos de Particionamento

Método k-means

- Sendo dado 'k' (número de segmentos)
 1. Dividir os objetos em 'k' subconjuntos
 2. Calcular o centro de cada subconjunto
 3. Atribuir cada objeto ao centro mais próximo
 4. Voltar ao ponto 2.;
parar quando não houver mais mudanças



- Sendo dado 'k' (número de segmentos)
 1. Dividir os objetos em 'k' subconjuntos
 2. Calcular o centro de cada segmento
 3. Atribuir cada objeto ao centroid mais próximo
 4. Voltar ao ponto 2.;
parar quando não houver mais pos.



Particionamento
Método k-means

Método k-means (exemplo *by hand*)

- Começamos com 9 objetos que pretendemos dividir em 2 segmentos;



Método k-means (exemplo *by hand*)

- Iniciamos com um posicionamento aleatório de 2 centroids;



Método k-means (exemplo *by hand*)

- Medimos a distância de cada objeto a cada centroid para determinar qual o mais próximo;



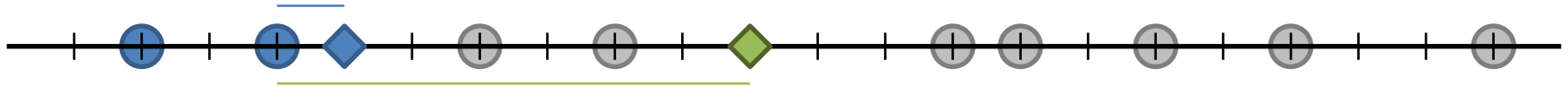
Método k-means (exemplo *by hand*)

- Atribuímos o primeiro objeto ao segmento representado pelo centroid mais próximo;



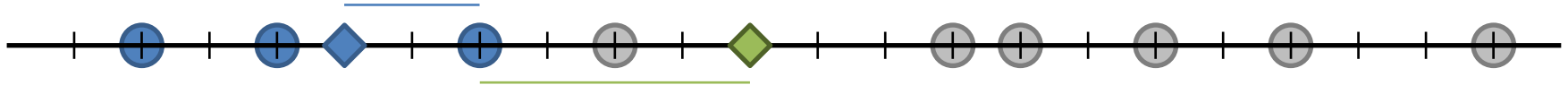
Método k-means (exemplo *by hand*)

- Fazemos a mesma comparação para todos os (restantes 8) objetos...



Método k-means (exemplo *by hand*)

- ... para os associar ao centroid respetivo...



Método k-means (exemplo *by hand*)

- ... sempre baseado na medida de distância menor que mede a maior similaridade;



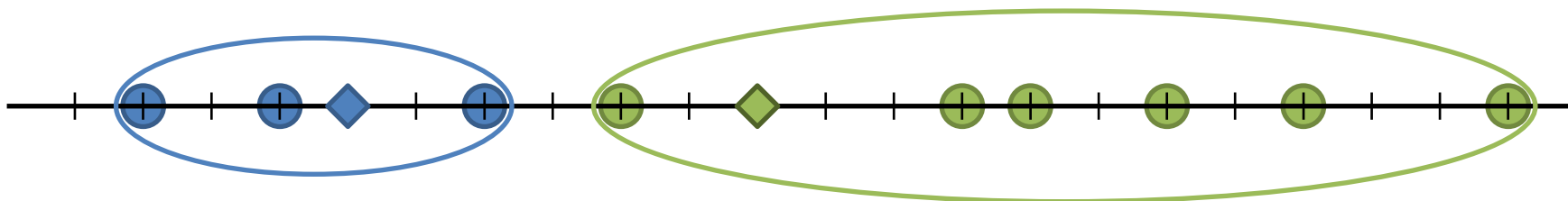
Método k-means (exemplo *by hand*)

- No final da primeira iteração, temos todos os objetos associados ao seu centroid...



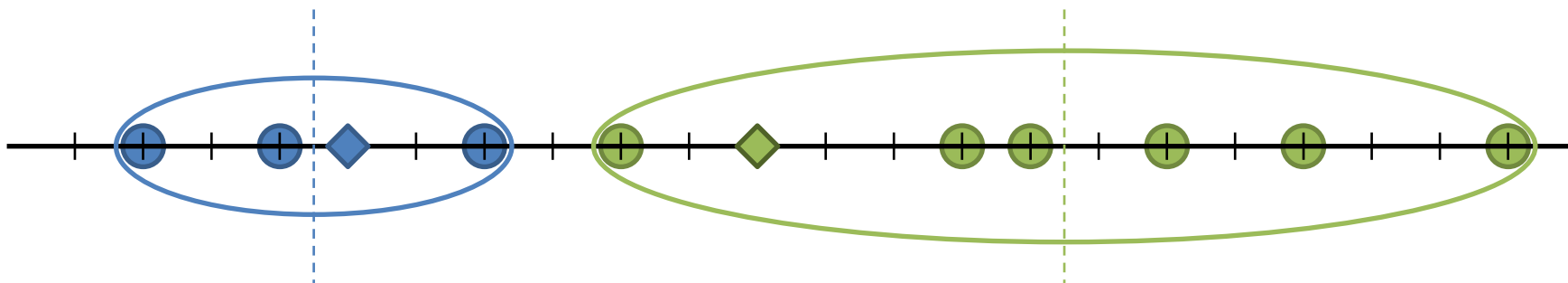
Método k-means (exemplo *by hand*)

- ... o que identifica a primeira solução de 2 segmentos;



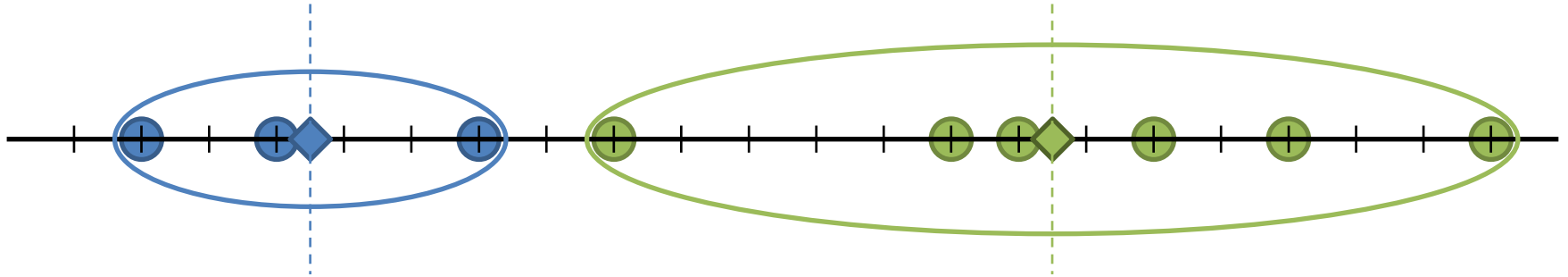
Método k-means (exemplo *by hand*)

- Calculamos o centro do segmento...



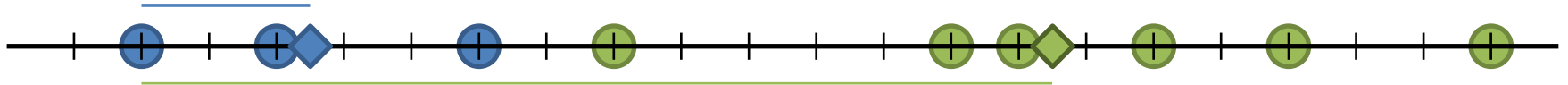
Método k-means (exemplo *by hand*)

- ... e colocamos lá o centroid;



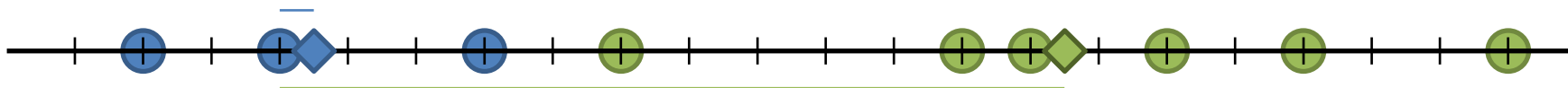
Método k-means (exemplo *by hand*)

- A partir daqui o processo repete-se...



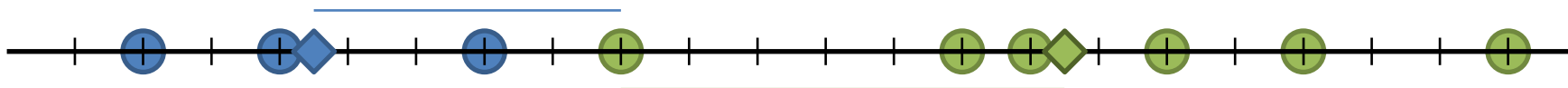
Método k-means (exemplo *by hand*)

- ... no sentido de reorganizar a associação dos objetos aos centroid mais próximos;



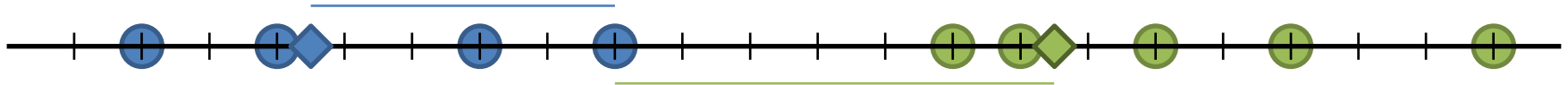
Método k-means (exemplo *by hand*)

- Desta forma, este objeto...



Método k-means (exemplo *by hand*)

- ... vai passar para o outro segmento;



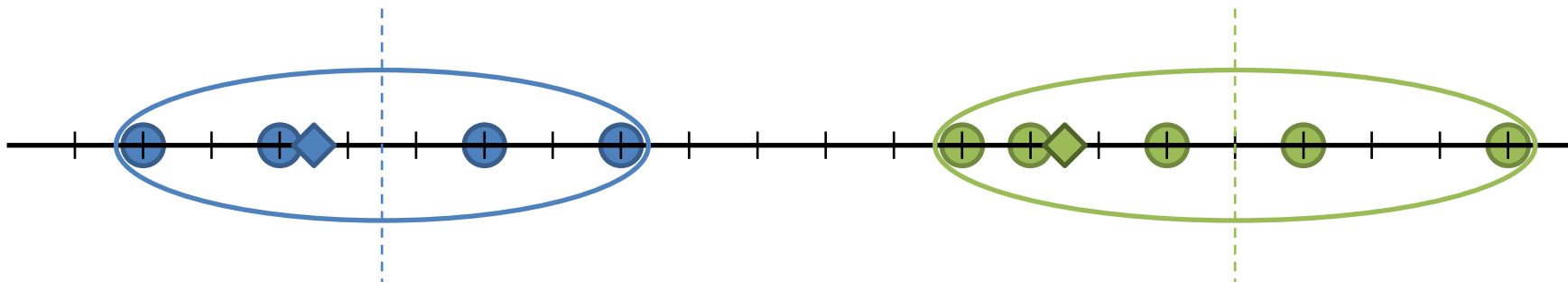
Método k-means (exemplo *by hand*)

- Os 2 segmentos têm, agora, esta configuração;



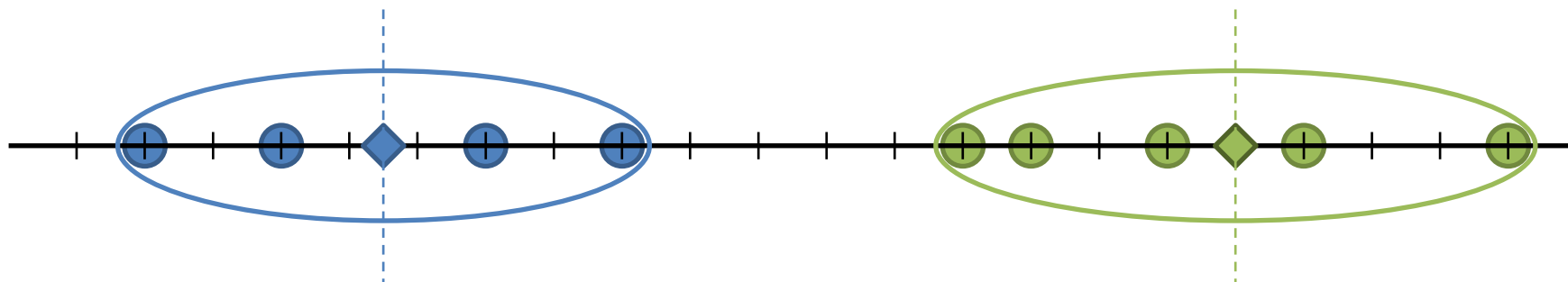
Método k-means (exemplo *by hand*)

- Calculamos, novamente, o centro de cada segmento...



Método k-means (exemplo *by hand*)

- ... e recolocamos o respectivo centroid nessa posição;



Método k-means (exemplo *by hand*)

- O processo continua, iterativamente, associando os objetos aos centroids mais similares;



Algoritmos de Particionamento

Método k-means

■ Vantagens:

- Relativamente eficiente:
sendo 'n' o número de objetos, 'k' o número de segmentos e 'i' o número de iterações, normalmente acontece $k, i \ll n$;
- Termina com ótimos locais.

■ Desvantagens:

- Aplicável, apenas, quando é possível calcular a média (*mean*);
- É necessário identificar o número de segmentos *a priori*;
- Incapacidade de lidar com ruído nos dados;
- Inadequado para determinar segmentos côncavos.



Algoritmos de Particionamento

Método k-medoids

- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.

Algoritmos de Particionamento

Método k-medoids

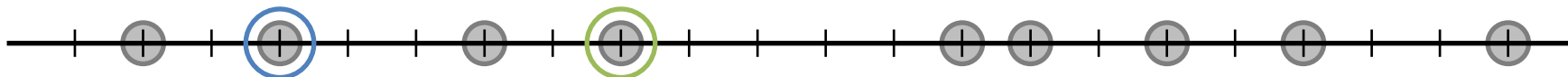
- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



Algoritmos de Particionamento

Método k-medoids

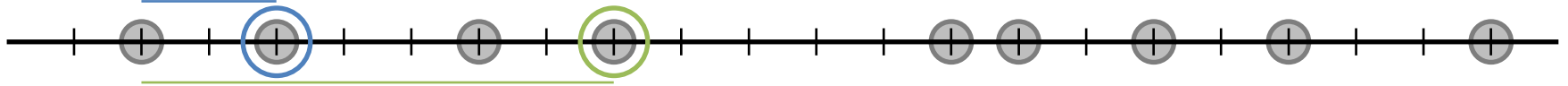
- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



Algoritmos de Particionamento

Método k-medoids

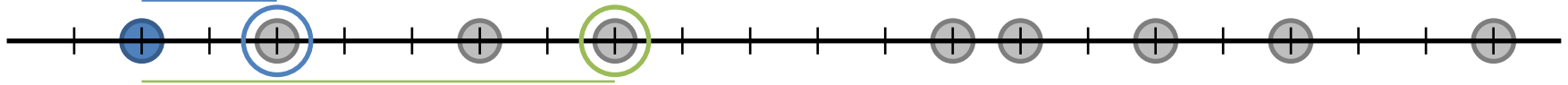
- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



Algoritmos de Particionamento

Método k-medoids

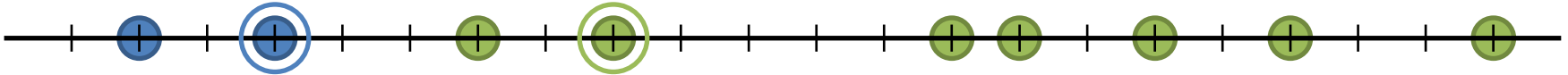
- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



Algoritmos de Particionamento

Método k-medoids

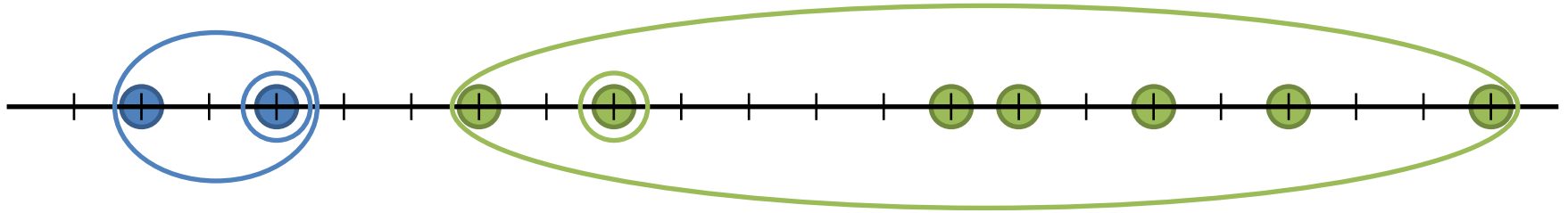
- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



Algoritmos de Particionamento

Método k-medoids

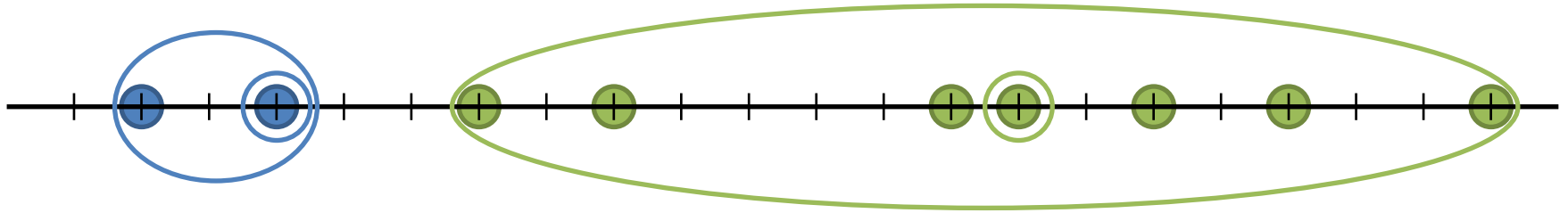
- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



Algoritmos de Particionamento

Método k-medoids

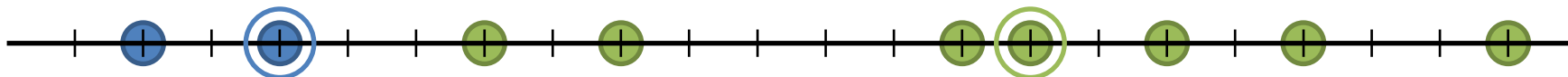
- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



Algoritmos de Particionamento

Método k-medoids

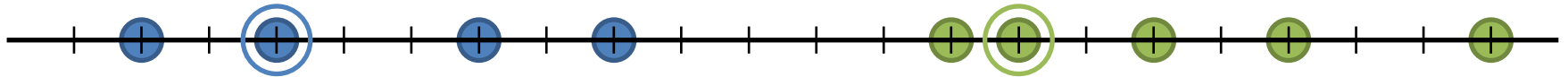
- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



Algoritmos de Particionamento

Método k-medoids

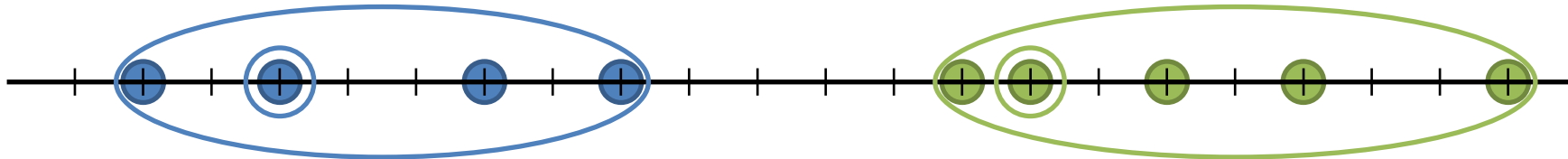
- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



Algoritmos de Particionamento

Método k-medoids

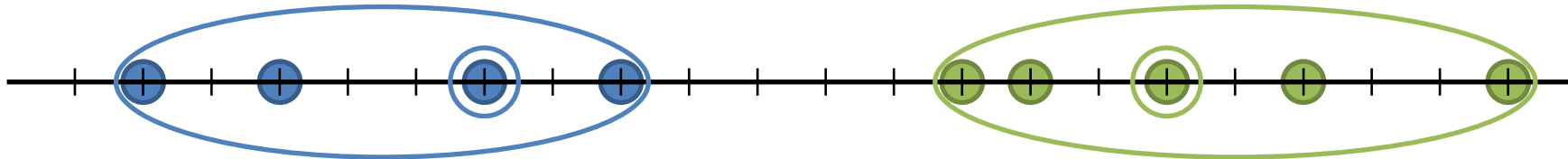
- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



Algoritmos de Particionamento

Método k-medoids

- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



Algoritmos de Particionamento

Método k-medoids

- Vantagens e Desvantagens:
 - É mais robusto do que o método k-means na presença de dados ruidosos, uma vez que os objetos selecionados são menos influenciáveis por valores extremos do que a média (*mean*);
 - Produz bons resultados para conjuntos de dados de pequenas dimensões;
 - Não se comporta tão bem quando se pretende a sua aplicação em conjuntos de dados de grandes dimensões.



Principais Métodos de Segmentação

- **Particionamento:**
 - criar várias partições e adotar um critério de avaliação;
- **Hierarquização:**
 - decompor hierarquicamente o conjunto de dados;
- **Outros:**
 - Baseados na Densidade:
 - aumentar o segmento enquanto a densidade de pontos estiver num determinado limite (utilizam-se funções de conectividade e densidade);
 - Baseados no Modelo:
 - criar modelos hipotéticos para cada segmento e testar a capacidade de adequação de cada ponto ao segmento.

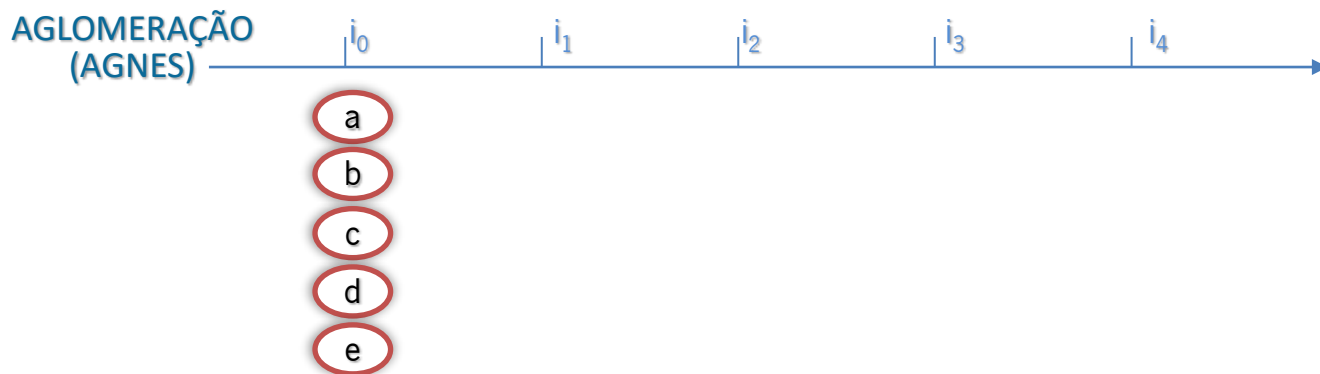


Algoritmos de Hierarquização

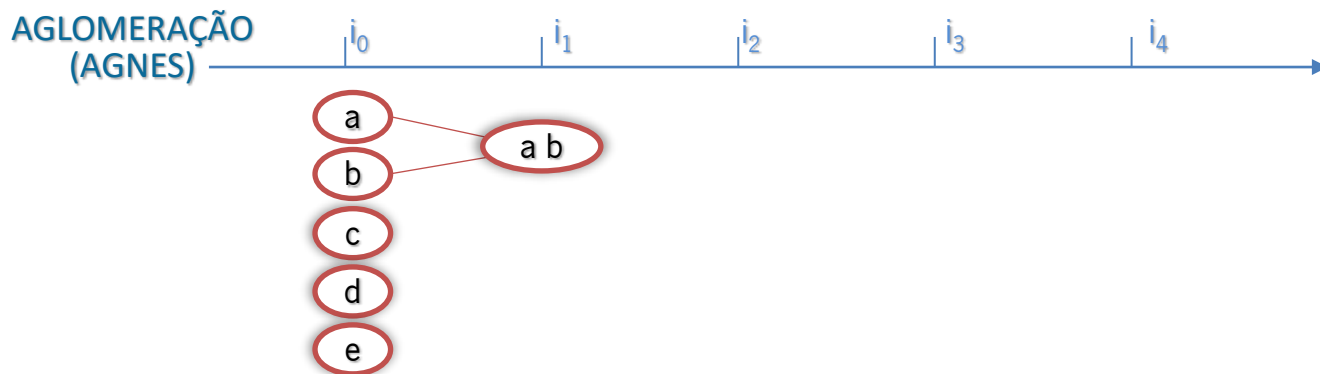
- Utilizam a matriz de distâncias como critério de segmentação;
- Os dados são agrupados em árvores de segmentos;
- Não requerem a definição do número de segmentos a procurar;
- Exigem a definição de uma condição de paragem:
 - quantidade de segmentos;
 - distância mínima entre objetos;
 - etc.
- Existem dois tipos de algoritmos de hierarquização:
 - Aglomeração: estratégia *bottom-up*;
 - Divisão: estratégia *top-down*.



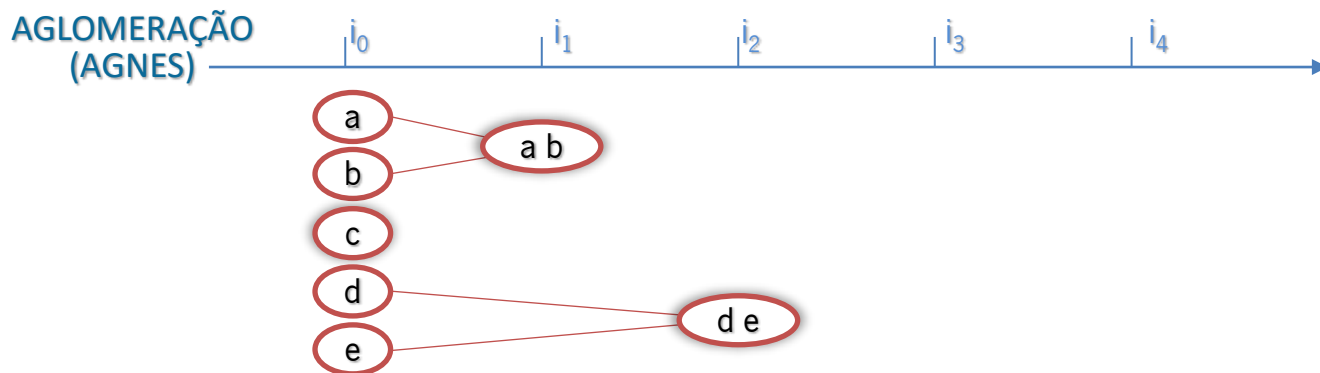
- Aglomeração:
 - Inicia-se formando segmentos com um objeto, para todos os objetos;
 - Prossegue juntando segmentos atômicos em segmentos cada vez mais amplos.



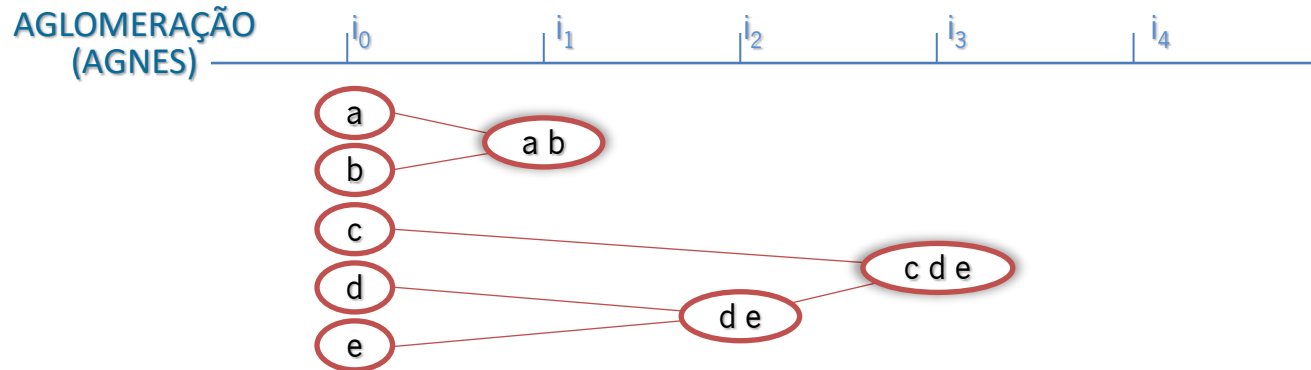
- Aglomeração:
 - Inicia-se formando segmentos com um objeto, para todos os objetos;
 - Prossegue juntando segmentos atômicos em segmentos cada vez mais amplos.



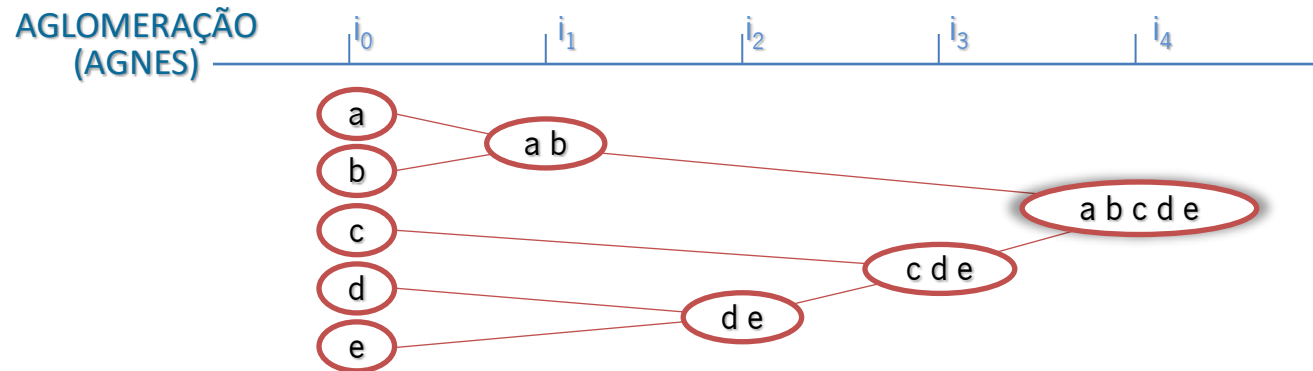
- Aglomeração:
 - Inicia-se formando segmentos com um objeto, para todos os objetos;
 - Prossegue juntando segmentos atômicos em segmentos cada vez mais amplos.



- Aglomeração:
 - Inicia-se formando segmentos com um objeto, para todos os objetos;
 - Prossegue juntando segmentos atômicos em segmentos cada vez mais amplos.



- Aglomeração:
 - Inicia-se formando segmentos com um objeto, para todos os objetos;
 - Prossegue juntando segmentos atômicos em segmentos cada vez mais amplos.



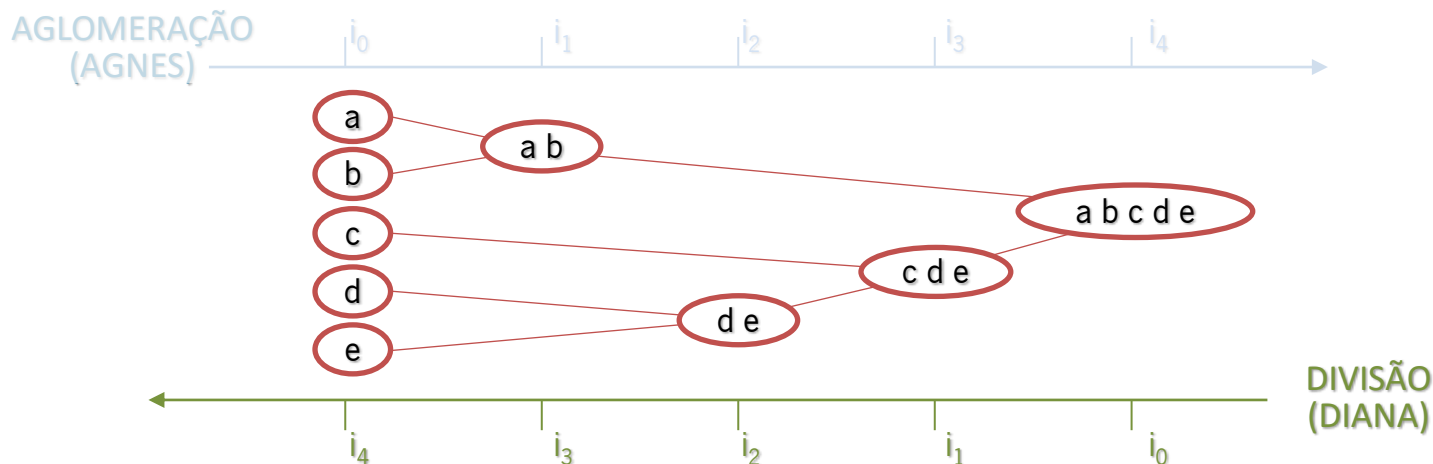
Algoritmos de Hierarquização

■ Aglomeração:

- Inicia-se formando segmentos com um objeto, para todos os objetos;
- Prossegue juntando segmentos atômicos em segmentos cada vez mais amplos.

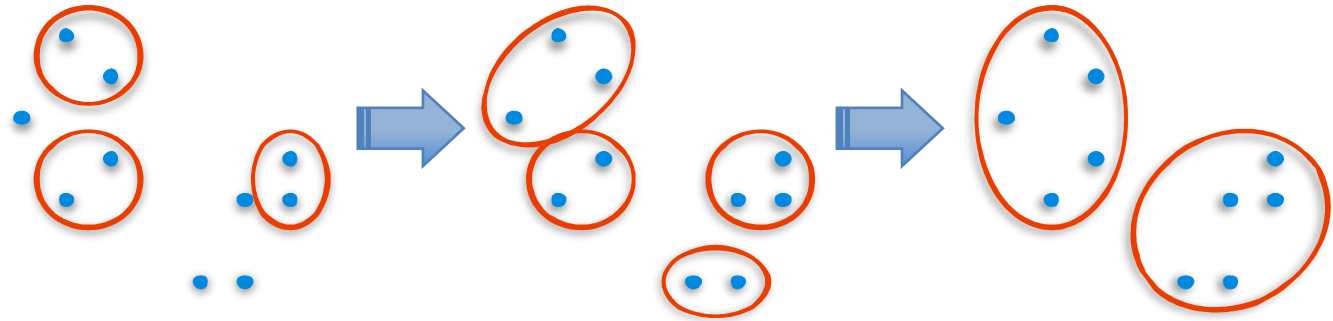
■ Divisão:

- Inicia-se com todos os objetos em um só segmento que se vai subdividindo em segmentos de menor dimensão;
- Aplicação prática muito rara.



Algoritmos de Hierarquização **AGNES: Agglomerative Nesting**

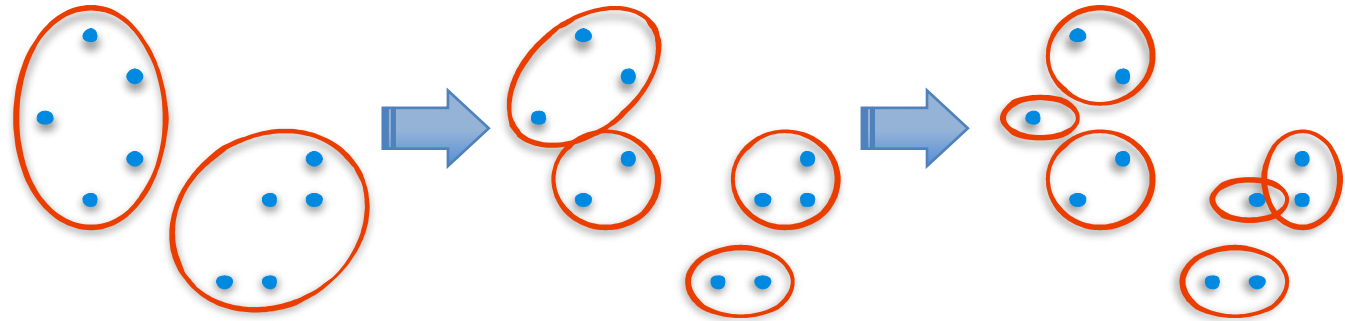
- Iterativamente, vai juntando objetos que apresentam menores valores de dissimilaridade: os conjuntos C1 e C2 são juntos se os objetos de C1 e de C2 produzem o menor valor de distância Euclidiana entre quaisquer dois objetos de segmentos distintos.



Algoritmos de Hierarquização

DIANA: Divisive Analysis

- Iterativamente e partindo de um segmento composto por todos os objetos, dividir em segmentos menores que maximizam a distância Euclidiana entre objetos vizinhos de segmentos diferentes.



Segmentação Hierárquica

- Dificuldades com o aumento de atributos ou de objetos:
 - à medida que aumentam os objetos a agrupar, aumenta o tempo necessário para procurar tais grupos;
- Não é necessário especificar o número de segmentos 'k';
basta “cortar” a árvore no nível 'k-1';
- Produz melhores resultados do que os algoritmos k-means;
- Uma hierarquia traduz alguma organização dos segmentos, ao contrário de um simples conjunto de segmentos.



- BIRCH: *Balanced Iterative Reducing and Clustering using Hierarchies*;
- Usa árvores com características sobre os segmentos e ajusta, iterativamente, a qualidade dos segmentos;
- É construída uma árvore que captura informação necessária para realizar as operações de segmentação:
 - *Clustering Feature*: contém informação sobre o segmento;
 - *Clustering Feature Tree*: contém informação sobre a organização arbórea da hierarquia.

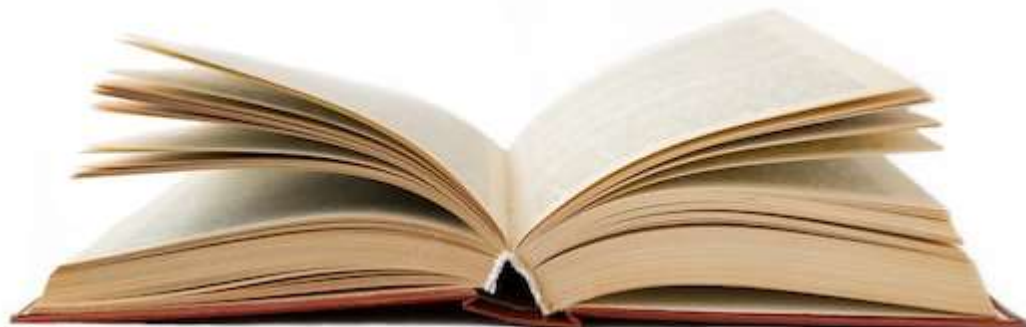
Outros Algoritmos

- CURE: *Clustering Using Representatives*;
- Seleciona pontos dispersos do segmento e vai reduzindo o tamanho do segmento em direção ao seu centro;
- Usa múltiplos pontos representativos;
- Em cada iteração, dois segmentos com o par de pontos representativos mais próximos são juntos.

- DBSCAN: *Density Based Spatial Clustering of Applications with Noise*;
- Algoritmo baseado no cálculo de valores de densidade e de conectividade locais;
- Características assinaláveis:
 - capaz de descobrir segmentos de formas não regulares;
 - capaz de lidar com ruído nos dados;
 - algoritmo de um só passo (scan);
 - obriga à definição de parâmetros de densidade como condição de paragem.

Referências bibliográficas

- Russell, S., & Norvig, P. (2009). Artificial Intelligence: A Modern Approach (3rd ed.). Upper Saddle River, NJ, USA: Prentice Hall Press.
- Mitchell, T. M. (1997/2015/2016). Machine Learning (1st ed.). McGraw-Hill International Editions.
- Hulten, G. (2018). Building Intelligent Systems. Berkeley, CA: Apress. <https://doi.org/10.1007/978-1-4842-3432-7>
- Feinberg, E. A., & Schwartz, A. (2002). Handbook of Markov Decision Processes: Methods and Applications. Springer US
- Data Mining: Concepts and Techniques
Jiawei Han, Micheline Kamber
- Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations
Ian Witten, Eibe Frank





Universidade do Minho
Departamento de Informática

APRENDIZAGEM E DECISÃO INTELIGENTES

**LEI/MiEI @ 2022/2023, 2º sem
[ADI³]**