

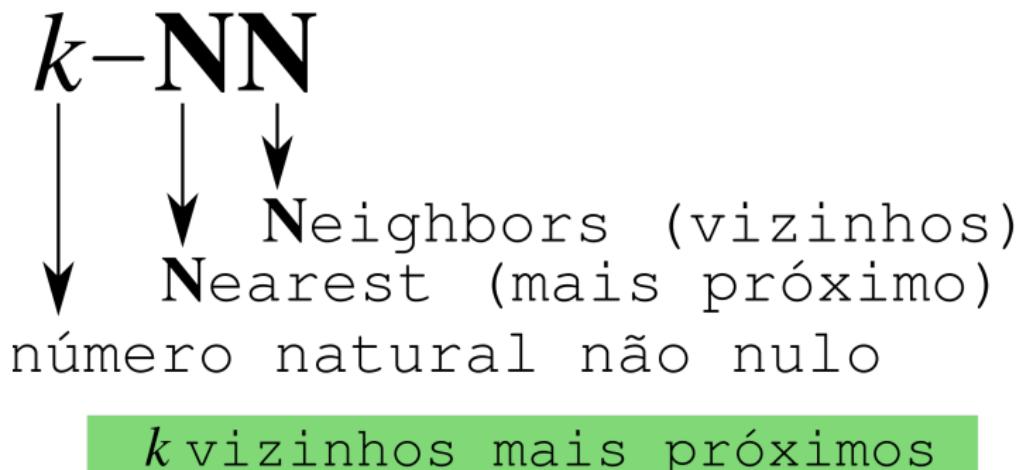
K-NN e K-Means

ENG04471



Tiago Oliveira Weber
2023

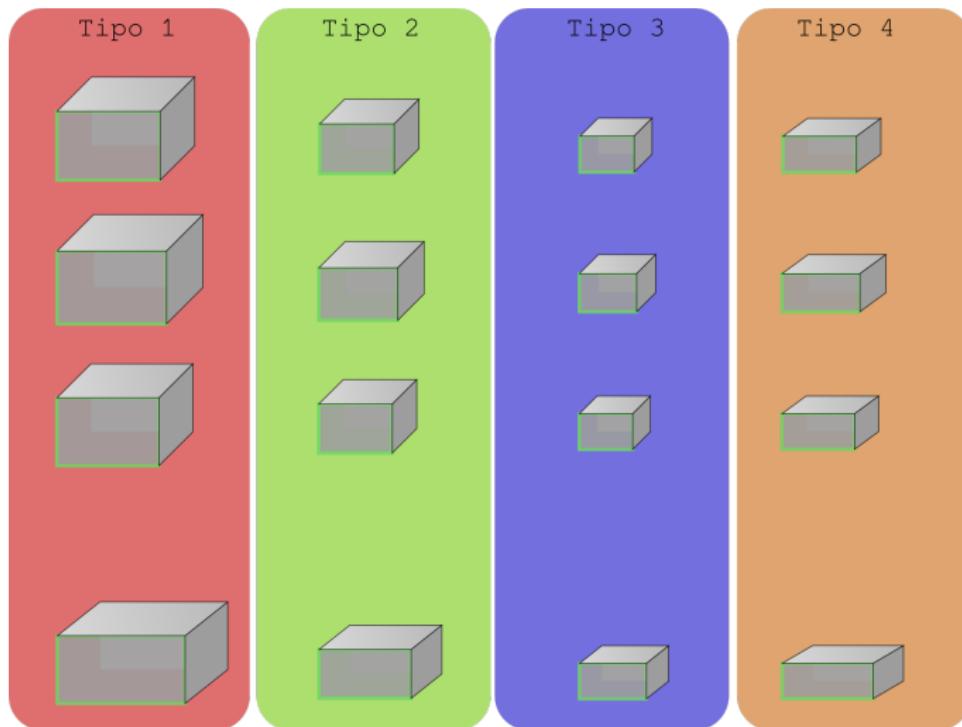
- ▶ Contextualização
- ▶ K-nn
 - ▶ Exemplos em Python
- ▶ Introdução a Clustering
- ▶ K-means
 - ▶ Exemplos em Python



Ideia básica

- ▶ instâncias com atributos similares provavelmente tem classes similares;
- ▶ o princípio pelo qual o k-NN aborda o aprendizado é por vias de memorização;
 - ▶ **sem representação interna** derivada dos exemplos (instâncias usadas no treinamento);
 - ▶ **simples armazenamento** dos exemplos.

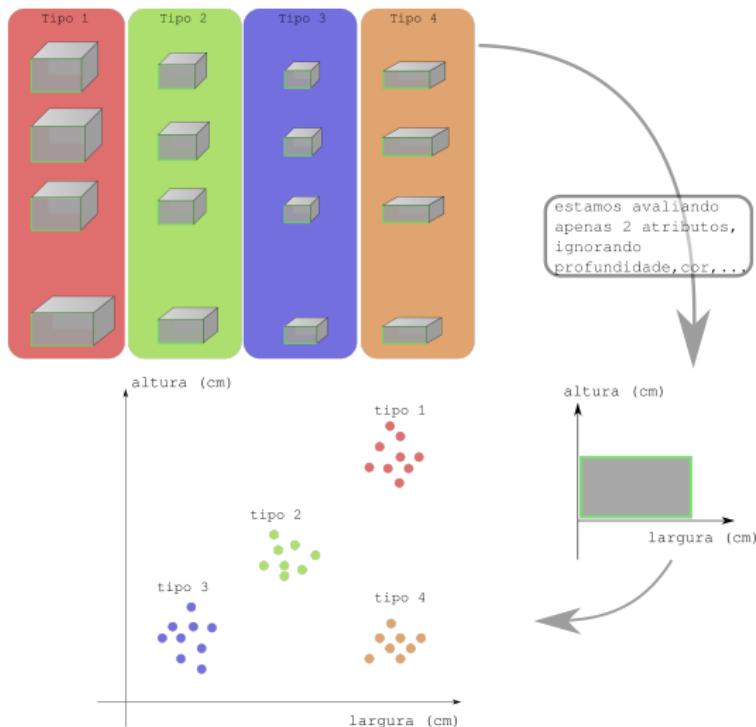
► Exemplo



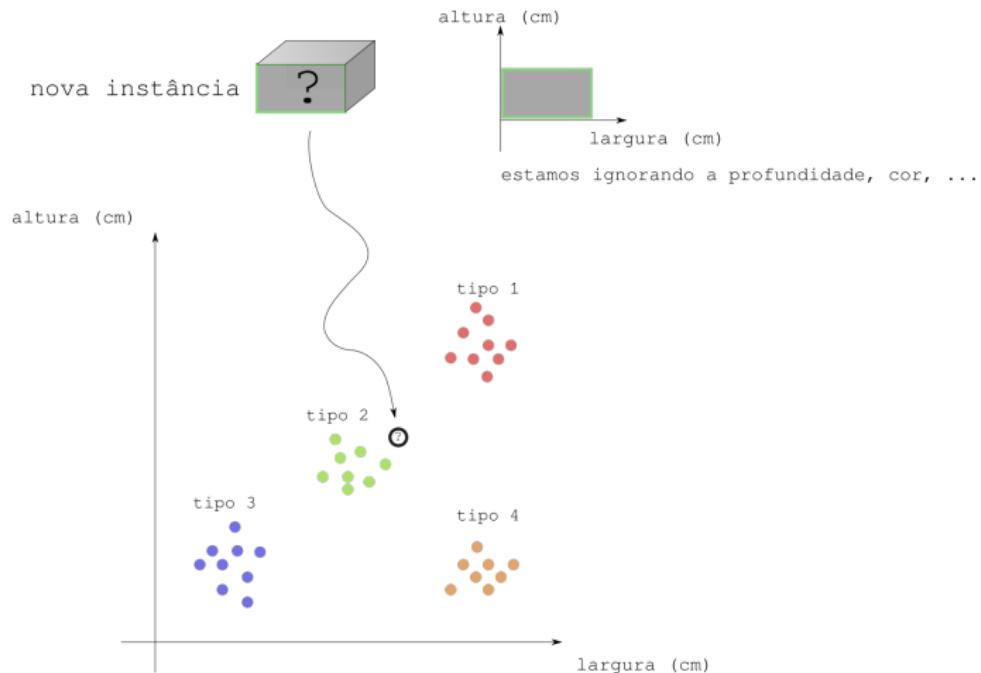
Técnica de k-NN

Tiago Oliveira Weber - UFRGS

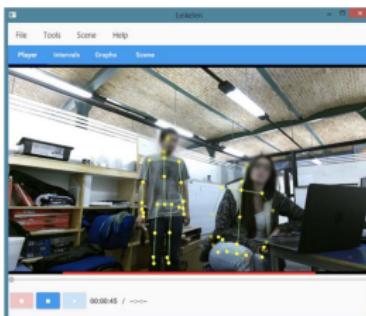
► Exemplo



► Exemplo (continuação)



- ▶ Exemplo de situação em que técnicas como k-NN podem ser aplicadas:
 - ▶ classificação de diferentes posturas pode ser obtida através de análise de dados processados de sensores de profundidade (como o Microsoft Kinect), relacionada ao tópico de Multimodal Learning Analytics¹

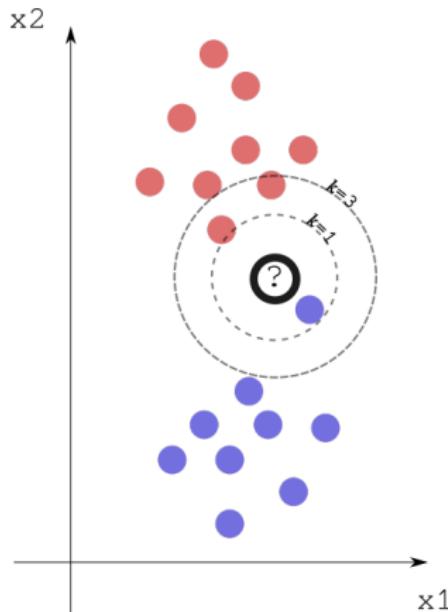


¹Roberto Munoz et al. "Development of a Software that Supports Multimodal Learning Analytics: A Case Study on Oral Presentations". Em: *Journal of Universal Computer Science* 24.2 (28 de fev. de 2018), pp. 149–170.

- ▶ O algoritmo vai comparar os atributos da nova instância com atributos de exemplos conhecidos (**memória**)
- ▶ o exemplo com *menor distância Euclidiana* da nova instância é o **vizinho mais próximo** (nearest neighbor, NN)

Por quê k (do k-NN)?

- ▶ basear-se apenas no vizinho mais próximo pode tornar o algoritmo sujeito a ruído nos exemplos ou nos dados de entrada;
- ▶ basear-se em k vizinhos mais próximos oferece alguma proteção a este ruído;
- ▶ assim, k é um número natural não nulo.



Se $k=1$ $\text{?} = \bullet$

Se $k=3$ $\text{?} = \bullet$

Observação sobre k

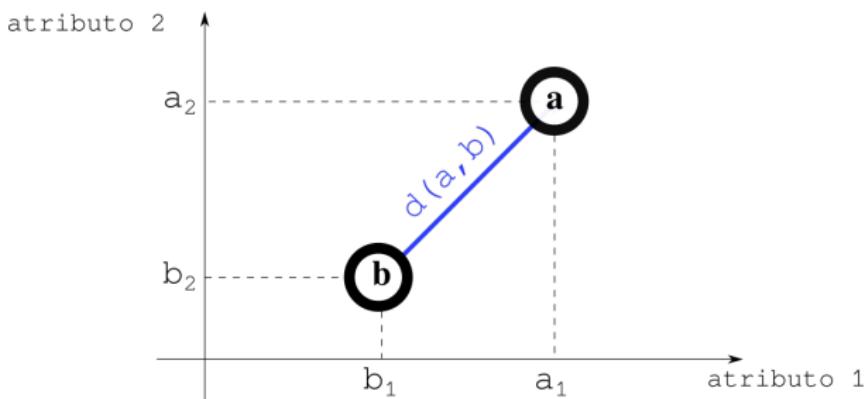
- ▶ são em geral, valores pequenos;
- ▶ números ímpares são preferidos;
- ▶ em problemas com apenas 2 classes de saída, k deve ser ímpar.

Medindo a Distância

Tiago Oliveira Weber - UFRGS

- ▶ Supondo um espaço bidimensional (um plano);
- ▶ Dada uma instância $\mathbf{a} = (a_1, a_2)$ e $\mathbf{b} = (b_1, b_2)$;
- ▶ A distância Euclidiana será:

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$



- para n atributos (dimensões):

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

- considerando a possibilidade de atributos contínuos e discretos:

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^n d(a_i, b_i)}$$

onde:

- para atributos contínuos: $d(a_i, b_i) = (a_i - b_i)^2$
- para atributos discretos: $d(a_i, b_i) = 0$ se $x_i = y_i$, 1 caso contrário.

Para uma nova instância:

- ▶ calcula-se a distâncias;
- ▶ define-se os k vizinhos mais próximos;
- ▶ o valor de saída:
 - ▶ para problema de classificação: é o valor pluralidade dos k -vizinhos (a classe majoritária entre os k vizinhos);
 - ▶ para problema de regressão: é a média do valor dos k -vizinhos.

Exemplo

Instância	Valores	Classe de Saída
1	[2.0 1.5 1.0]	0
2	[1.0 1.0 0.5]	0
3	[1.0 3.0 1.5]	1
4	[0.8 2.2 0.7]	1
5	[3.0 1.0 0.2]	0
6	[0.4 2.0 1.7]	1
7	[1.2 1.2 1.5]	1

- ▶ Suponha uma instância $[0.9 \ 1.1, \ 0.5]$ de classe desconhecida;
- ▶ As distâncias ficariam:

Instância	Valores	Classe de Saída	Distância
1	[2.0 1.5 1.0]	1	1,27
2	[1.0 1.0 0.5]	0	0,14
3	[1.0 3.0 1.5]	1	2,15
4	[0.8 2.2 0.7]	0	1,12
5	[3.0 1.0 0.2]	1	2,12
6	[0.4 2.0 1.7]	1	1,58
7	[1.2 1.2 1.5]	1	1,05

- organizando por proximidade:

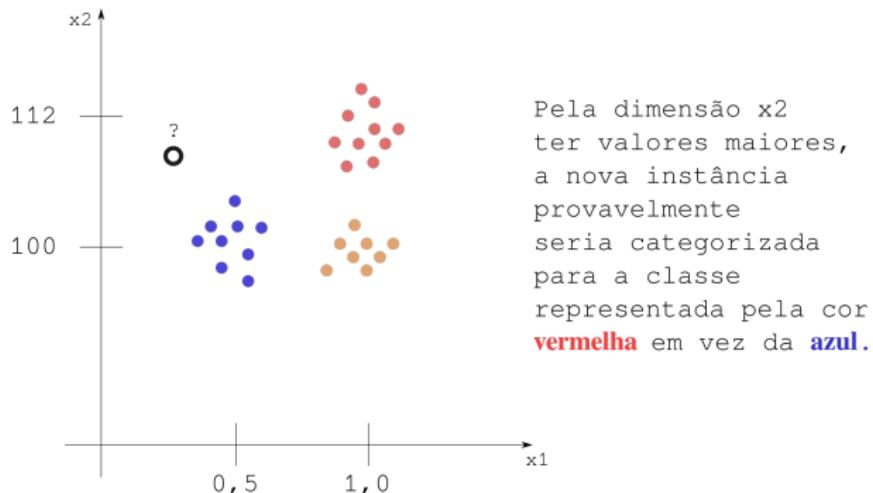
mais próximo	Instância	Valores	Classe de Saída	Distância
1	2	[1.0 1.0 0.5]	0	0,14
2	7	[1.2 1.2 1.5]	1	1,05
3	4	[0.8 2.2 0.7]	0	1,12
4	1	[2.0 1.5 1.0]	1	1,27
5	6	[0.4 2.0 1.7]	1	1,58
6	5	[3.0 1.0 0.2]	1	2,12
7	3	[1.0 3.0 1.5]	1	2,15

- Para $k=1$, classe é 0;
- Para $k=3$, classe é 0;
- Para $k=5$, classe é 1.

- ▶ Exemplo em Python

- ▶ atributos irrelevantes podem prejudicar desempenho
 - ▶ possibilidade: estudar o dataset e as correlações entre variáveis de entrada e saída
- ▶ a escala dos atributos é importante
 - ▶ um (ou mais) atributo pode dominar os demais

O problema:



- ▶ a dimensão com valores absolutos menores será "prejudicada", pois não será representativa na equação da distância, visto que outra dimensão tem magnitude muito maior.

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{(\Delta x_1)^2 + (\Delta x_2)^2}$$

Adaptar Escala (attribute scaling)

- ▶ podemos dividir o valor de cada atributo por um valor (para todos os exemplos)
 - ▶ por exemplo, o máximo valor de cada atributo

$$\mathbf{x}' = \frac{\mathbf{x}}{max(\mathbf{x})}$$

onde:

- ▶ \mathbf{x} é o vetor com os valores originais do atributo x para todos os exemplos;
- ▶ \mathbf{x}' é o vetor adaptado (escalado) do atributo x para todos os exemplos;
- ▶ No entanto, apenas dividir pode gerar problemas quando tratamos de atributos cujos valores mínimos (e por consequência o intervalo de valores) sejam muito diferentes de um atributo para outro.

Adaptar Escala (attribute scaling)

Exemplo

Normalizar (normalization)

- A normalização Min-Max:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

onde:

- x é o vetor com os valores originais do atributo x para todos os exemplos;
- x' é o vetor adaptado (escalado) do atributo x para todos os exemplos;

Normalizar (normalization)

Exemplo

Exemplo

Exemplo de Aplicação em Instrumentação

Tiago Oliveira Weber - UFRGS

- Detecção de falhas em sistema de classificação de gases²

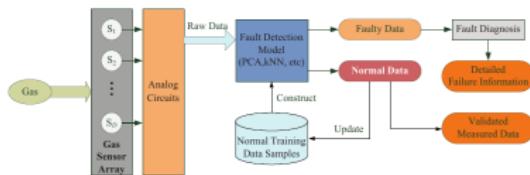


Figura: Método de detecção de falhas para matriz de sensores de gás [Yang, Sun, Chen, 2016].

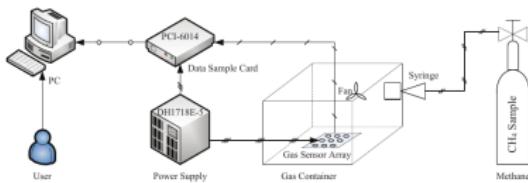


Figura: Sistema experimental para matriz de sensores de gás [Yang, Sun, Chen, 2016].

² Jingli Yang, Zhen Sun e Yinsheng Chen. “Fault Detection Using the Clustering-kNN Rule for Gas Sensor Arrays”. Em: *Sensors* 16.12 (2016), p. 2069. DOI: 10.3390/s16122069.

- ▶ Classificação de indivíduos com base em padrão de caminhada³:
 - ▶ dados coletados através de acelerômetro com 3 eixos posicionado nas costas dos sujeitos;
 - ▶ proposta de técnicas de extração de *features*;
 - ▶ uso de K-NN para classificação

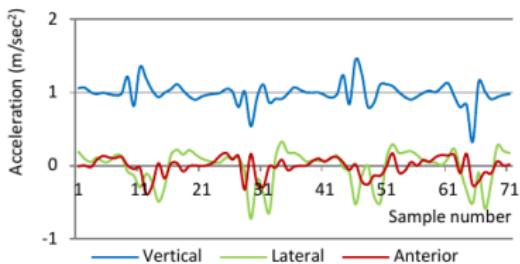


Figura: Dados de aceleração de um dos sujeitos do experimento [Choi et al., 2014]

³Sangil Choi et al. "Biometric gait recognition based on wireless acceleration sensor using k-nearest neighbor classification". Em: 2014 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2014. DOI: 10.1109/iccnc.2014.6785491.

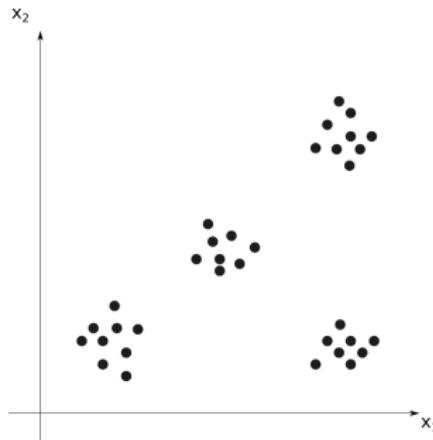
- ▶ relacionado à tarefa de *Reconhecimento de Padrões*;
- ▶ relacionado à percepção
- ▶ aprendizado não-supervisionado

O que é?

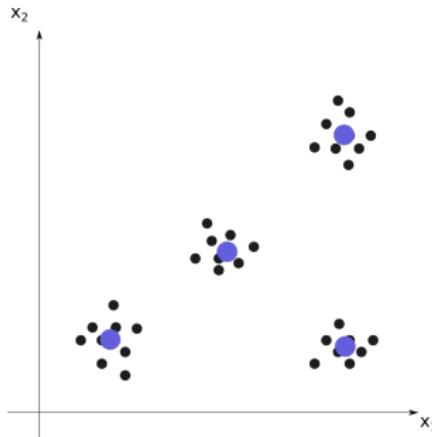
- ▶ encontrar agrupamentos nos dados, de forma que exista grande similaridade entre os dados de um mesmo grupo (categoria)
- ▶ tem interesse em descobrir características úteis nos dados

Quando é útil?

- ▶ tarefas de classificação em que não saibamos categorias pré-definidas (aprendizado não-supervisionado)
 - ▶ Apesar de não termos rótulos pré-definidos, será interessante descobrir as características que distinguem os grupos
 - ▶ Os rótulos podem ser criados *a posteriori*
 - ▶ Exemplos:
 - ▶ segmentação de clientes
 - ▶ filtrar spams
 - ▶ análise de documentos: agrupar por tema
 - ▶ detecção de fraude
- ▶ tarefas em que existe um número muito grande de possíveis classes (eg.: identificar cores RGBA)
 - ▶ clustering é utilizado para identificar um número reduzido de classes presentes (e agrupadas), para facilitar uso de aprendizado supervisionado posteriormente
- ▶ como parte de aprendizado semi-supervisionado
- ▶ ...



- ▶ O objetivo é encontrar agrupamentos que:
 - ▶ não tem intersecção; cada amostra estará em apenas 1 agrupamento;
 - ▶ sejam uma partição.
 - ▶ dentro de um agrupamento, as amostras devem serem similares entre si (no que diz respeito as features)



- ▶ Representação dos Agrupamentos por **Centroides**:
 - ▶ o ponto que representa a média das amostras em seu grupo para cada dimensão.

Quantos grupos devem existir?

- ▶ deve ser fornecido previamente;
- ▶ há formas de obter ou tentar descobrir o valor ótimo automaticamente.

Medição de distância

- ▶ Da mesma forma como analisamos para o k-NN, é importante tomar cuidados com as diferentes escalas dos atributos;
- ▶ uma possibilidade: normalizar atributos para o intervalo [0,1]
- ▶ cálculo da distância para atributos contínuos e discretos:

$$dist(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^n d(a_i, b_i)}$$

onde:

- ▶ para atributos contínuos: $d(a_i, b_i) = (a_i - b_i)^2$
- ▶ para atributos discretos: $d(a_i, b_i) = 0$ se $x_i = y_i$, 1 caso contrário.

- ▶ algoritmo que faz uma partição do conjunto de dados em k clusters;
- ▶ ao fim da execução, cada dado deve pertencer ao cluster cuja centróide está mais próxima do mesmo.

Example (Centróide de um conjunto)

É a média dos elementos do conjunto em cada uma de suas coordenadas.

- ▶ Características do algoritmo
 - ▶ fácil de implementar;
 - ▶ sensível a inicialização;
 - ▶ depende que seja passado o número k de clusters.

► Pseudocódigo

```
1: função K-MEANS(exemplos, atributos, k)
2:    $\{Cluster_1, \dots, Cluster_k\} \leftarrow$  criar k clusters iniciais.
3:    $\{C_1, \dots, C_k\} \leftarrow$  calcular o centróide para cada cluster
4:   enquanto critério de parada não satisfeito faça
5:     escolher um exemplo x
6:      $j \leftarrow argmin_{i \in \{1, \dots, k\}} distânciia(x, C_i)$ 
7:     se  $x \in Cluster_j$  então
8:       não faça nada
9:     senão
10:      mover x para  $Cluster_j$ 
11:       $\{C_1, \dots, C_k\} \leftarrow$  recalcular o centróide de cada cluster
12:    fim se
13:  fim enquanto
14: fim função
```

- o critério de parada é atingido quando não há mais modificações nos clusters
- o algoritmo não garante convergência ao mínimo global

Pontos importantes

- ▶ normalização será importante, tal qual era para o k-NN
- ▶ a inicialização poderá:
 - ▶ tornar o cálculo mais rápido ou mais lento
 - ▶ influenciar nos resultados finais

Inicialização

- ▶ Forma de inicialização:
 - ▶ aleatória
 - ▶ k-means++:
 - ▶ seleciona o primeiro centro aleatoriamente;
 - ▶ seleciona o centro dos demais com probabilidade proporcional ao quadrado da distância dos centros já escolhidos

Exemplo em Python

- ▶ podemos avaliar o quanto bem uma determinada partição (\mathcal{P}) está se desempenhando.
- ▶ uma forma de fazer tal avaliação (função custo) é através da soma do erro quadrático (SSE):

$$SSE(\mathcal{P}) = \sum_{j=1}^k \sum_{i=1}^{n_j} d(x_i^j, c_j)^2$$

onde:

- ▶ k é o número de clusters;
- ▶ n_j é o número de exemplos em um determinado cluster j ;
- ▶ $d(a,b)$ é a distância entre a e b ;
- ▶ x_i^j é o exemplo i contido dentro do cluster j .

Exemplo de Aplicação em Instrumentação

Tiago Oliveira Weber - UFRGS

- ▶ uso como forma de reduzir quantidade de dados para permitir análise em sistema inteligente para controle localizado de ar-condicionado⁴
 - ▶ k-means auxilia na diminuição da dimensionalidade dos dados

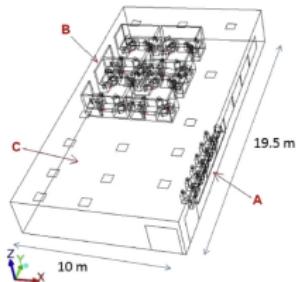


Figura: Modelagem da sala de laboratório utilizada para os testes [Zhou; So;, Wu; 2015]

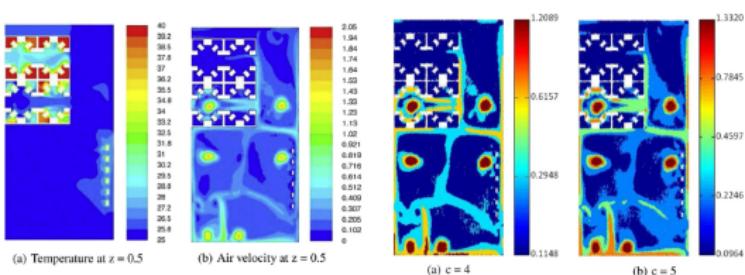


Figura: Simulação de dinâmica de fluidos para temperatura e velocidade do ar em uma sala de laboratório [Zhou; So;, Wu; 2015]

Figura: Resultados do K-means para velocidade do ar com diferentes números de clusters [Zhou; So;, Wu; 2015]

⁴Hongming Zhou, Yeng Chai Soh e Xiaoying Wu. "Integrated analysis of CFD data with K-means clustering algorithm and extreme learning machine for localized HVAC control". Em: *Applied Thermal Engineering* 76 (2015), pp. 98–104. DOI: 10.1016/j.applthermaleng.2014.10.004.

Tópicos estudados

- ▶ K-nn
- ▶ K-means
- ▶ Exemplos em Python