UNIFACS
Feira de Santana

Noberto Maciel
nobertomaciel@ulife.com.br

Sistemas de Controle e Inteligência Artificial

AULA 09
Prática

Algoritmos de Agrupamento de Dados

Algoritmos de agrupamento

São algoritmos de aprendizado não-supervisionado (não necessitam de treinamento) que usam técnicas de clustering (agrupamento) visando encontrar padrões ocultos nos dados.

Podem ser:

Baseados em grade (grid-based), em particionamento, hierarquia, densidade, grafos, probabilísticos…

Finalidade dos algoritmos de agrupamento (clustering):

Descoberta de padrões

Tenta encontrar padrões ocultos nos dados, organizando os elementos em grupos (clusters) de acordo com sua similaridade ou dissimilaridade.

Aplicações práticas:

1) **Segmentação de Clientes:** no marketing, ajuda a dividir os clientes em grupos com comportamentos semelhantes para estratégias personalizadas;
2) **Agrupamento de Dados:** em big data, facilita a organização de grandes volumes de informações;
3) **Reconhecimento de Padrões:** identifica padrões em imagens, textos e outros tipos de dados.
4) **Compressão de Dados:** reduz a complexidade dos dados agrupando elementos semelhantes;
5) **Detecção de Anomalias:** ajuda a encontrar outliers (valores fora do padrão) em segurança cibernética e detecção de fraudes;
6) **Biologia e Genômica:** agrupa sequências genéticas para estudar semelhanças entre espécies ou mutações.
7) **...**

Principais Algoritmos de Clustering:

1) **K-Means (baseado em partição):** divide os dados em K grupos, ajustando iterativamente os chamados “centróides” (média dos elementos);
2) **Hierárquico:** cria uma hierarquia de clusters, podendo ser divisivo (top-down) ou aglomerativo (bottom-up);
3) **DBSCAN (baseado em densidade):** detecta clusters com base em densidade, ideal para dados com formatos irregulares;
4) **K-Medoids (baseado em particição):** semelhante ao K-Means, mas ajusta os centros de grupo em elementos reais (medoids);
5) **D-CLUSTER (Grid-based):** divide o espaço de dados em grades e ajusta dinamicamente o tamanho das células;
6) **Affinity Propagation (baseado em grafos):** baseia-se na troca de mensagens entre pontos para identificar centros de cluster, sem precisar definir um número fixo de clusters.

K-Means

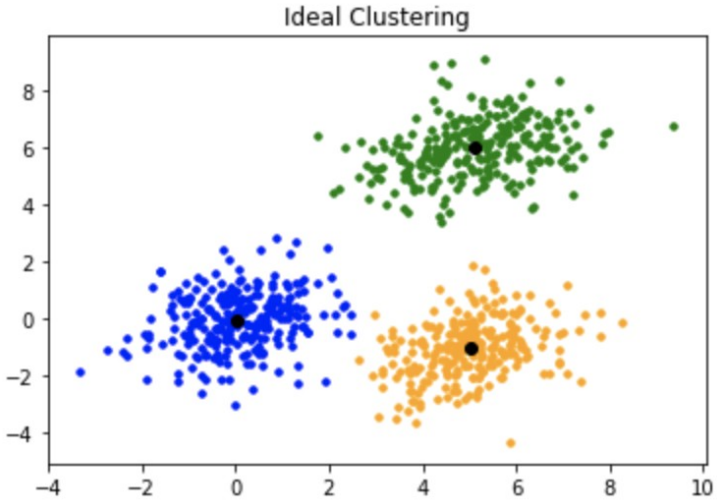
O que é:

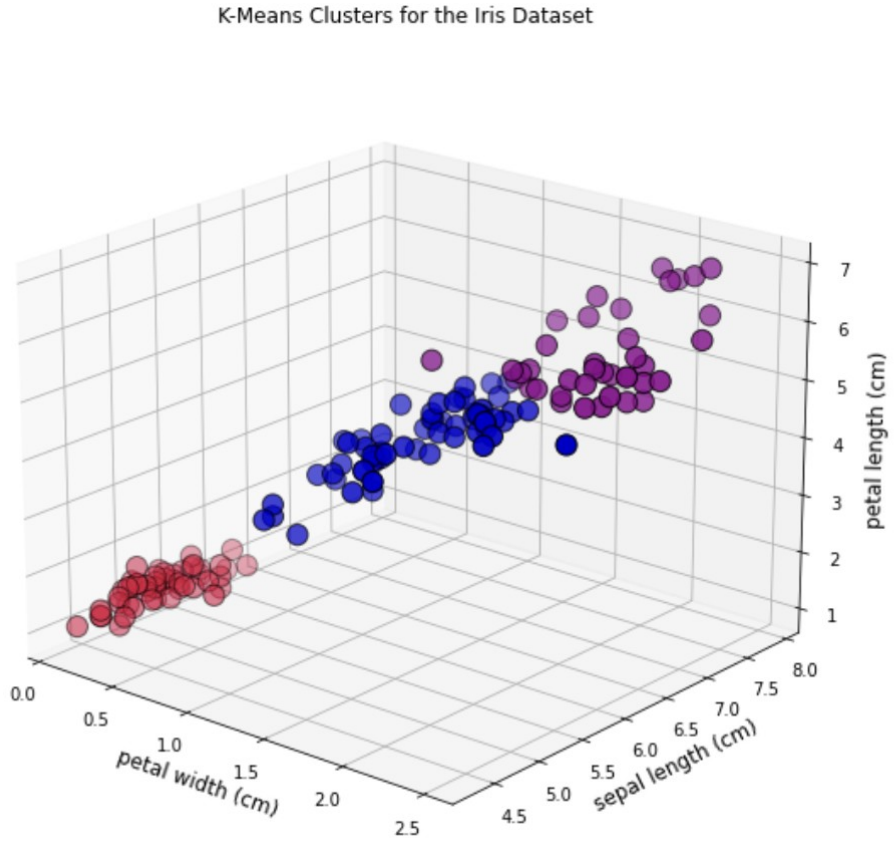
Algoritmo que usa uma técnica particional (partitional clustering) para encontrar um número K de clusters pré-determinado.

Objetivo:

Encontrar padrões ocultos nos dados, tentando gerar grupos com características similares.

K-Means


Ideal Clustering
8
6
4
2
0
−2
−4
−4
−2
0
2
4
6
8
10


K-Means Clusters for the Iris Dataset
petal length (cm)
1
2
3
4
5
6
7
petal width (cm)
0.0
0.5
1.0
1.5
2.0
2.5
sepal length (cm)
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

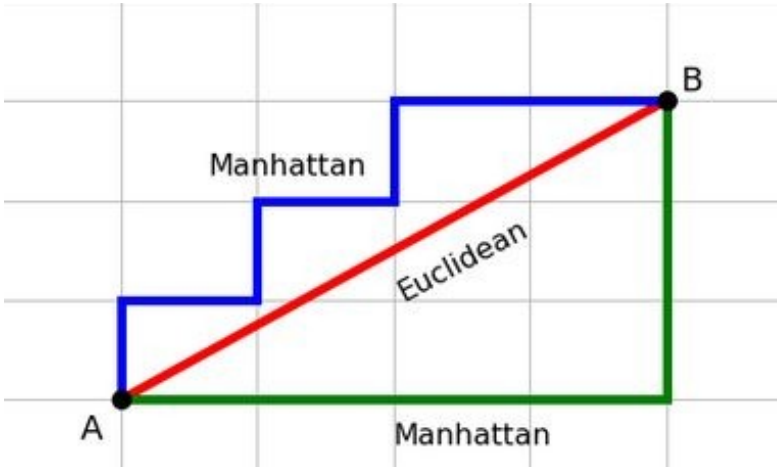
Algoritmo 1: K-means Básico

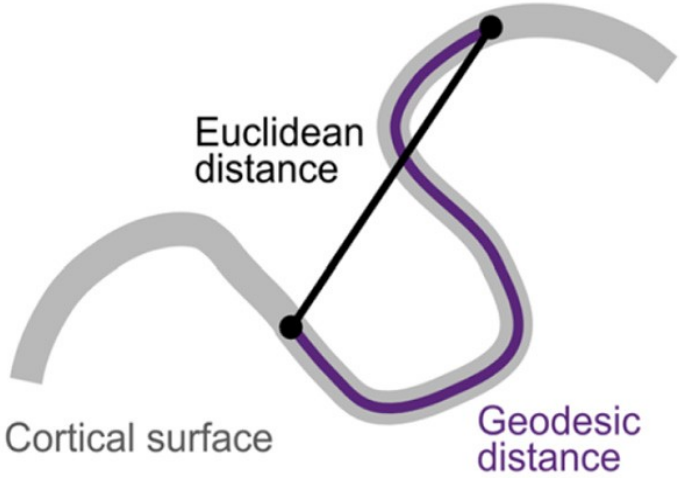
```
1: Obter K = inteiro;
2: Atribuir K pontos como os centróides iniciais, randomicamente
3: Enquanto (centróides mudarem de posição)
4:    Calcular a distância dos novos centróides para todos os pontos
5:    Formar K clusters atribuindo cada ponto ao centróide mais próximo
6:    Recalcular o centróide de cada cluster (ponto médio)
7: Fim
```

Métricas de distância:

São formas de calcular a distância entre os centróides e os elementos do conjunto de dados.

- Distância Euclidiana
- Distância Geodésica
- Distância de Manhattan


B
Manhattan
Euclidean
A
Manhattan


Euclidean
distance
Cortical surface
Geodesic
distance

Euclidiana:

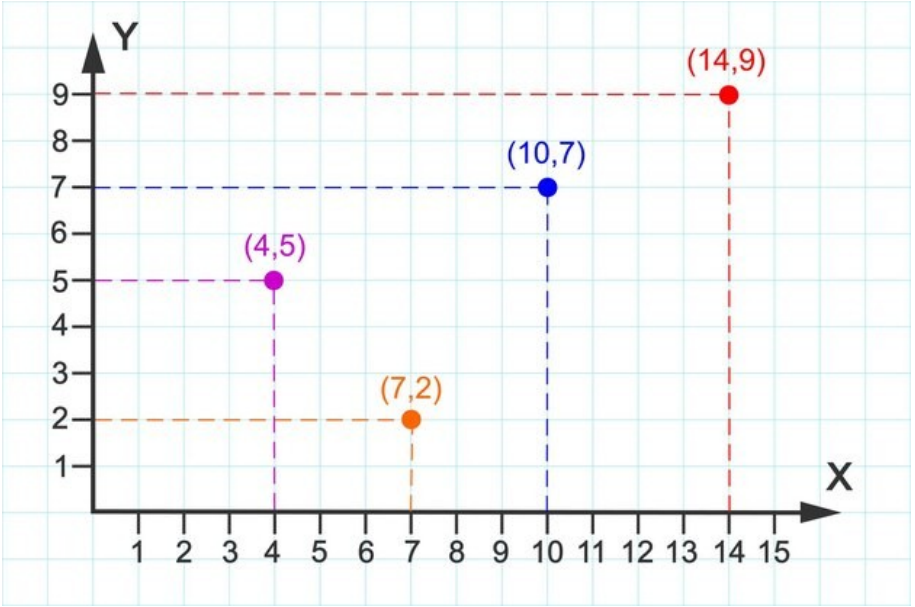
$$d = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}$$

Manhattan:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

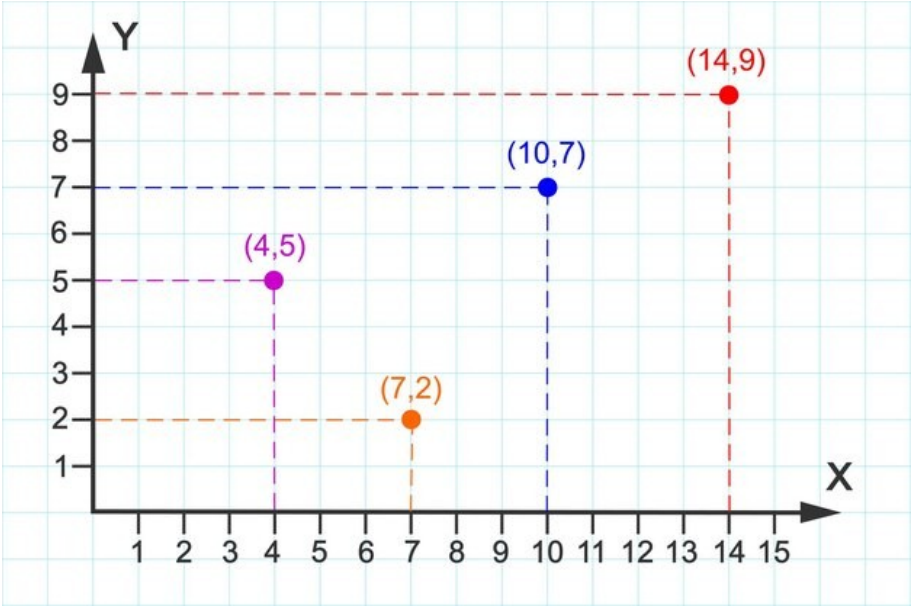
Vetores no espaço bidimensional:

P1(4.0, 5.0)
P2(7.0, 2.0)
P3(10.0, 7.0)
P4(14.0, 9.0)



Vetores no espaço bidimensional:

P1(4.0, 5.0)
P2(7.0, 2.0)
P3(10.0, 7.0)
P4(14.0, 9.0)



Em Python (Lista/DataFrame):

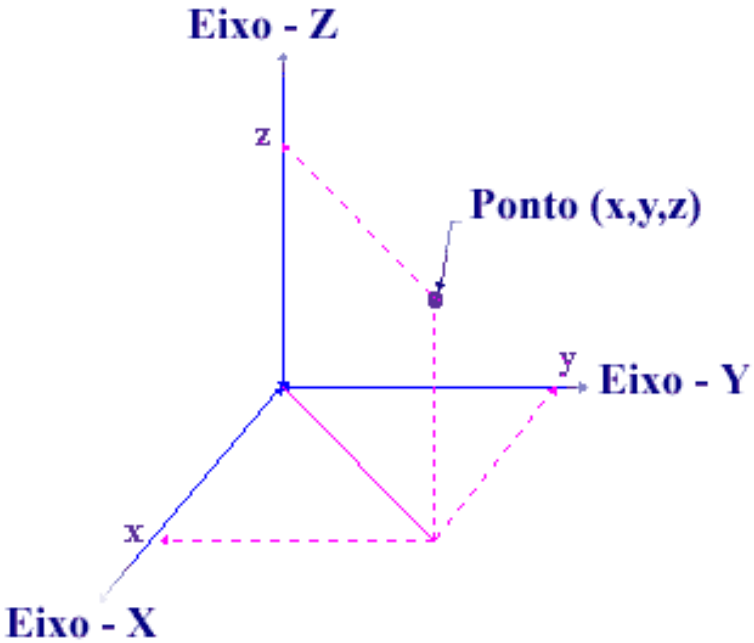
```
V = [ [4.0, 5.0],[7.0, 2.0],[10.0, 7.0],[14.0, 9.0] ]

ou

dados = {
    'X': [4.0, 7.0, 10.0, 14.0],
    'Y': [5.0, 2.0, 7.0, 9.0]
}
V = pd.DataFrame(dados)
```

Vetores no espaço tridimensional:

P1(4.0, 5.0, 3.0)
P2(4.5, 6.0, 3.2)
P3(4.4, 3.0, 7.0)



Vetores no espaço tridimensional:

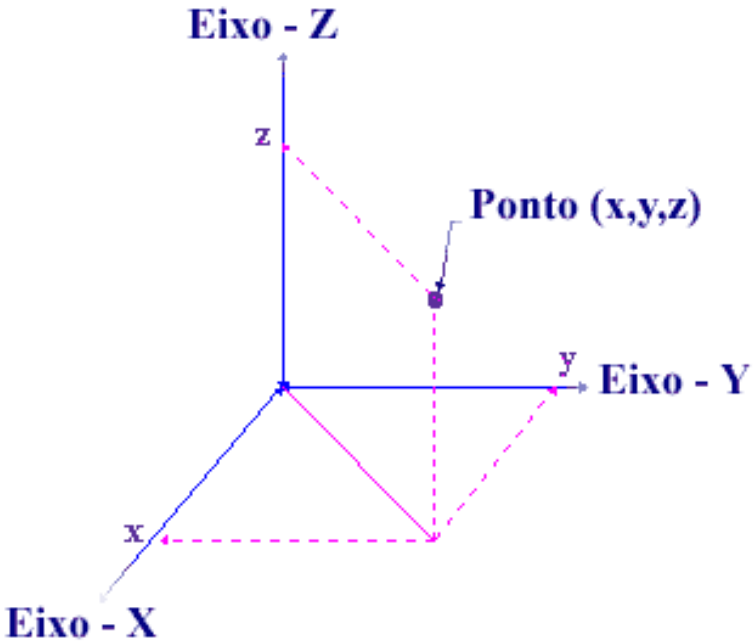
P1(4.0, 5.0, 3.0)
P2(4.5, 6.0, 3.2)
P3(4.4, 3.0, 7.0)

Em Python (Lista/DataFrame):

```
V = [ [4.0, 5.0, 3.0], [4.5, 6.0, 3.2], [4.4, 3.0, 7.0] ]
```

ou

```
dados = {
    'X': [4.0, 4.5, 4.4],
    'Y': [5.0, 6.0, 3.0],
    'Z': [4.4, 3.0, 7.0]
}
V = pd.DataFrame(dados)
```



Vetores no espaço n-dimensional:

P1(4.0, 5.0, 3.0, 4.0, 6.1…….. n)
P2(4.0, 5.0, 3.0, 4.0, 6.1…….. n)

Vetores no espaço n-dimensional:

P1(4.0, 5.0, 3.0, 4.0, 6.1…….. n)
P2(4.0, 5.0, 3.0, 4.0, 6.1…….. n)

Em Python (Lista/DataFrame):

V = [[4.0, 5.0, 3.0, 4.0, 6.1…….. n], [4.0, 5.0, 3.0, 4.0, 6.1…….. n] ….. Pλ]

ou

```
dados = {
    'X': [4.0, 4.0,... Pλ],
    'Y': [5.0, 5.0,... Pλ],
    'Z': [3.0, 3.0,... Pλ],
   ...
    'n': [P1, P2, Pλ]
}
V = pd.DataFrame(dados)
```

Aplicações:

1) Segmentação de Clientes (Marketing e Vendas): agrupar clientes com base em comportamento de compra, permitindo campanhas mais personalizadas; identificar perfis de clientes semelhantes para estratégias de fidelização.

2) Análise de Imagens e Processamento de Vídeo: compressão de imagens, reduzindo cores ao agrupar pixels similares; segmentação de imagens, separando objetos de fundo.

3) Agrupamento de Dados em Redes e Telecomunicações: identificar padrões de uso da rede para otimizar tráfego e recursos; detectar anomalias no tráfego, ajudando na segurança cibernética.

4) Bioinformática e Medicina: agrupamento de genes com funções semelhantes; identificação de padrões em exames médicos, como na detecção de tumores em imagens.

5) Geolocalização e Planejamento Urbano: clustering de locais para planejamento de infraestrutura (ex.: estações de transporte); segmentação de áreas com base em dados socioeconômicos.

6) Análise Financeira e Detecção de Fraudes: agrupar transações para identificar padrões de comportamento financeiro; detectar atividades suspeitas em cartões de crédito e contas bancárias.

7) Recomendação de Conteúdo: identificar grupos de usuários com gostos similares para recomendações personalizadas (ex.: Netflix, Spotify);

8) Manutenção Preditiva na Indústria: agrupar sensores e máquinas com comportamentos semelhantes para prever falhas antes que ocorram.

Vantagens:

Fácil de implementar;

Rápido;

Trabalha bem com grandes conjuntos de dados.

Desvantagens:

Sensibilidade a outliers (pontos discrepantes);

A qualidade dos agrupamentos gerados está diretamente ligada à inicialização randômica dos centróides;

Não trabalha bem com alta dimensionalidade;

Apresenta distorções em conjuntos de dados com agrupamentos desiguais.

Exercício:

1) Leitura do Capítulo 1 do livro “Inteligência Artificial: Uma Abordagem Moderna” de Stuart J. Russell; Peter Norvig

2) Desenvolva o algoritmo K-Means em Python (não pode utilizar biblioteca de IA);

    1) Use o algoritmo desenvolvido para encontrar padrões ocultos no dataset Iris

    2) Plotar os padrões para cada conjunto de atributos (sepal_lenght/sepal_witdh, petal_lenght/petal_width) separando em cores (ou seja, dois gráficos de dispersão);

    3) O número de agrupamentos a serem gerados (K) deve ser compatível com o número de classes do dataset;

a) Fazer em dupla

b) Entrega por e-mail até dia 17/03 17h

Referências

[1] CHOWDHARY, K. R. Fundamentals of artificial intelligence. New Delhi: Springer India, 2020.

[2] NORVIG, Peter; RUSSELL, Stuart. Inteligência Artificial: uma abordagem moderna. Tradução da 4ª Ed. Rio de Janeiro: Elsevier, 2022.

[3] DOS DA SILVA, Fabrício M; LENZ, Maikon L.; FREITAS, Pedro H C.; SANTOS, Sidney C. Bispo. Inteligência artificial. Porto Alegre: SAGAH, 2019. E-book.

[4] MACIEL, Noberto Pires. Métodos de descoberta adaptativa de subconsultas para busca diversificada de imagens, 2024, 160 f., Dissertação (mestrado) - Programa de Pós-Graduação em Ciência da Computação, Universidade Estadual de Feira de Santana, Feira de Santana, 2024. Disponível em: http://tede2.uefs.br:8080/handle/tede/1847

[5] https://www.datacamp.com/pt/tutorial/dbscan-clustering-algorithm

[6] https://www.datacamp.com/pt/tutorial/hierarchical-clustering

[7] https://medium.com/@prasanth32888/k-medoids-clustering-cee6042155c6

[8] https://medium.com/cwi-software/entendendo-clusters-e-k-means-56b79352b452