Universidade Federal do Maranhão
Programa de Pós Graduação Em Ciência da Computação

Aluno
CARLOS EDUARDO NASCIMENTO CAJADO

Professor
Prof. Dr. LUCIANO REIS COUTINHO

CCOM0002 - APRENDIZAGEM DE MÁQUINA

Lista de Atividades 2 - Questão 2 e 3
Algoritmo K-means e
Agrupamento Hierárquico

São Luís, MA
Julho- 2024

1. Introdução

1.1 Contexto

De acordo com Steinbach et al. [1], a clusterização hierárquica é considerada uma técnica de alta qualidade para formação de clusters, proporcionando uma visão detalhada das relações entre os dados. Essa abordagem constrói uma árvore hierárquica (ou dendrograma) que permite visualizar a estrutura dos dados em vários níveis de granularidade. No entanto, sua aplicação é limitada pela complexidade quadrática, que pode tornar o processamento lento para grandes conjuntos de dados. Essa limitação faz com que a clusterização hierárquica seja mais adequada para conjuntos de dados menores ou para casos onde a qualidade dos clusters é mais crítica do que a velocidade de processamento.

Por outro lado, o K-means e suas variações apresentam uma complexidade de tempo linear, sendo muito mais eficientes em termos de processamento. O K-means é amplamente utilizado devido à sua simplicidade e velocidade, o que o torna ideal para grandes conjuntos de dados. No entanto, essa eficiência vem ao custo da qualidade dos clusters, que muitas vezes é inferior quando comparada à clusterização hierárquica. O K-means tende a produzir clusters esféricos e de tamanhos semelhantes, o que pode não refletir a verdadeira estrutura dos dados.[2]

1.2 problema

Neste trabalho, o algoritmo K-means será empregado para explorar o melhor particionamento de uma base de dados não anotada em grupos naturais. Serão realizados múltiplos agrupamentos variando tanto o número desejado de clusters (valor K) quanto o número e tipo de atributos considerados, além das medidas de distância ou similaridade utilizadas. Além, também, de utilizar o agrupamento hierárquico para comparação com o K-means.

1.3 objetivos

- Realizar múltiplos agrupamentos dos dados, variando o número de clusters desejados (valor K), o número de atributos considerados e a medida de distância ou similaridade utilizada.

- Avaliar os resultados dos experimentos utilizando análise do cotovelo e da silhueta dos grupos para determinar o particionamento mais adequado.
- Decidir qual dos agrupamentos realizados proporciona o melhor particionamento dos dados em grupos naturais, com base nas métricas de avaliação utilizadas.
- Variar a medida de distância entre grupos utilizada (utilizando agrupamento hierárquico), para Avaliar e comparar os resultados com os resultados do K-means.

1.4 Configuração do Ambiente

Para desenvolvimento do projeto, temos as seguintes configurações:

1.3.1 Software:

- Windows 11
- Linguagem python versão 3.12.2
- Jupyter Notebook
- Bibliotecas scikitlearn, pandas, numpy, matplotlib, seaborn.

1.3.2 Hardware:

- Processador: Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz   1.80 GHz
- Memória RAM: 16,0 GB (utilizável: 15,9 GB)

2. Bases de Dados

2.1 Descrição

- Título da base : Classificação multiclasse de feijão seco

Desenvolvido a partir de sete tipos diferentes de feijão seco com variedades registradas e características semelhantes, o objetivo foi obter uma classificação uniforme das sementes visando um agrupamento dos grãos conforme suas características.

2.2 Estatísticas

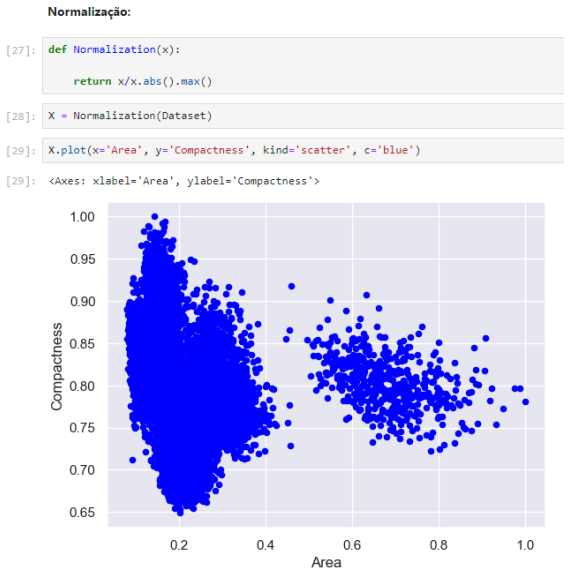
Para o modelo de classificação, foram obtidas imagens de 13.611 grãos de 7 tipos diferentes de feijão (SEKER, BARBUNYA, BOMBAY, CALI, HOROZ, SIRA, DERMESON) usando uma câmera de alta resolução. Foram extraídas 16 características: 12 dimensões e 4 formatos a partir dos grãos. [3]

3. Preparação (Pré- processamento)

3.1 Seleção

Serão utilizadas 16 características e 13.611 instâncias para a criação do modelo. Os dados passarão por um processo de normalização para melhorar a aplicação. Entretanto, para a visualização, a dimensionalidade será reduzida para as características raio e compacidade, como pode ser visto na figura 1.

Figura 1: Redução para raio e compacidade



Autor: próprio

3.2 Limpeza dos dados

A base de dados possui originalmente dados anotados. No entanto, para aplicar o modelo k-means, será necessário remover a coluna contendo as classificações.

4 Experimentos e Resultados

4.1 Usando o algoritmo K-means para base de dados.

Sabendo que a base de dados possui 7 classes, o modelo K-Means será treinado com 7 clusters, utilizando inicialização aleatória e 10 reinicializações para encontrar a melhor solução, figura 2. Após o treinamento, duas características específicas (raio e compacidade) são selecionadas para visualização. A visualização dos clusters é realizada através de um

gráfico de dispersão, onde os pontos são coloridos de acordo com os rótulos dos clusters atribuídos pelo modelo. Essa abordagem permite uma análise visual da distribuição dos clusters nas duas dimensões selecionadas, conforme ilustrado na Figura 3.

Figura 2: modelo para 7 clusters.

```
# dados em escala
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Treinamento do modelo KMeans
kmeans = KMeans(n_clusters=7, n_init='auto', init='random')
kmeans.fit(X)

# Selecionar duas características para visualização (raio e compacidade)
X_treino_vis = pd.DataFrame(X[:, [1, 11]], columns=['raio', 'Compactness'])

# Visualização do KMeans
plt.figure(figsize=(10, 8))
scatter = plt.scatter(X_treino_vis['raio'], X_treino_vis['Compactness'], c=kmeans.labels_, s=20, cmap='viridis', alpha=0.6)
```

Autor: próprio

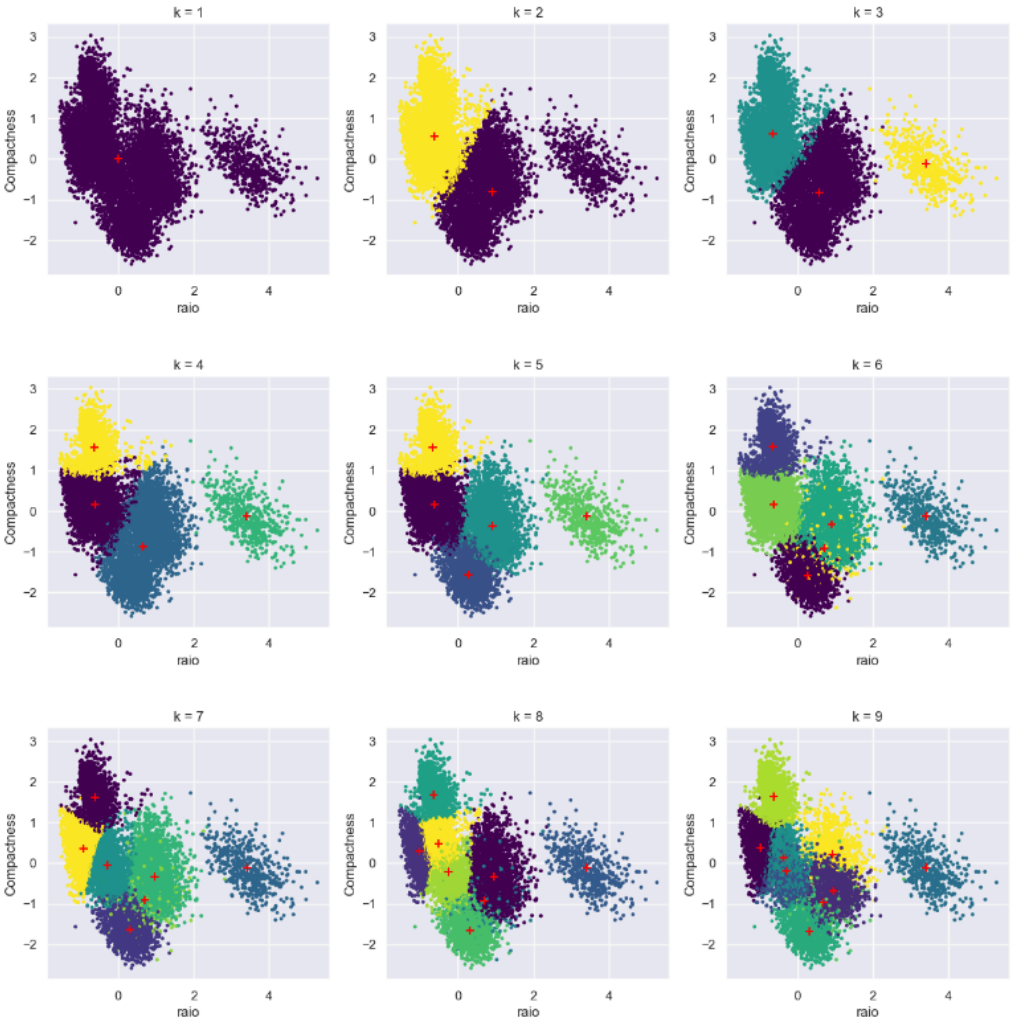
Figura 3: visualização para 7 clusters.

Autor: próprio

4.2 Variando o número de clusters desejados (valor K)

Esse experimento se utiliza de diferentes valores de k (número de clusters) de 1 a 9, treinando um modelo KMeans para cada valor de k e armazenando a inércia (soma dos erros quadráticos dentro dos clusters) correspondente. Além disso, para cada valor de k, é selecionado as duas características (raio e compacidade) e visualiza os clusters em gráficos de dispersão mostrados na figura 4.

Figura 4: Experimento matriz de gráficos de dispersão.



Autor: próprio

Figura 5: Gráfico do cotovelo.

Autor: próprio

Cada gráfico mostra a distribuição dos dados nas duas dimensões selecionadas, com os pontos coloridos de acordo com os rótulos dos clusters atribuídos pelo modelo e os centros dos clusters destacados. Isso permite uma análise visual da formação dos clusters. A análise visual revela que, com 7 clusters, há uma boa separação das instâncias. Esta conclusão é corroborada pelo gráfico de cotovelo, mostrado na Figura 5, que indica que 7 clusters representam a partição ideal para o conjunto de dados, evidenciada pela diminuição acentuada na inércia até esse ponto, seguido por uma estabilização.

4.3 Comparação agrupamento hierárquico x K-Means

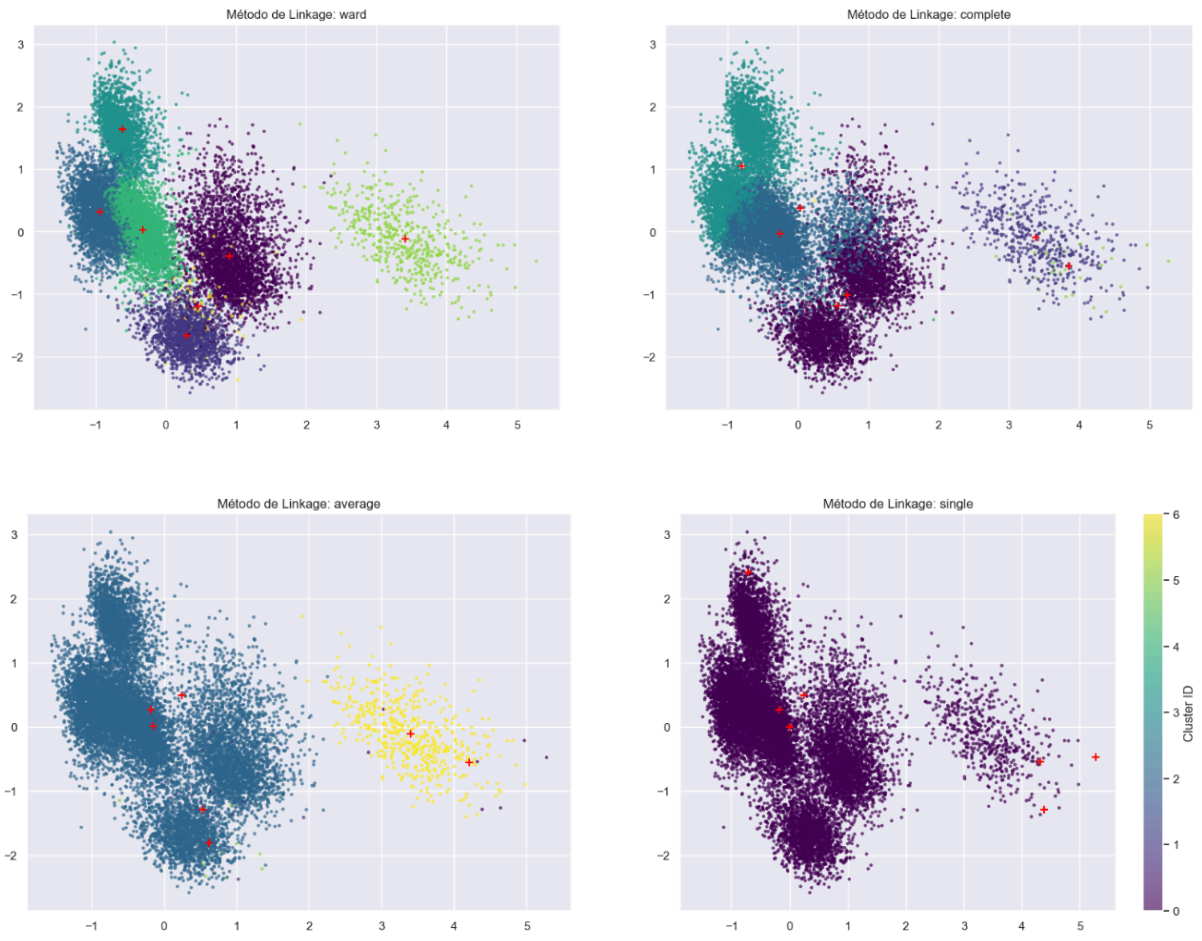
Os métodos mais comuns para calcular a distância entre os clusters são *Ward*, *Complete*, *Average* e *Single*. Em resumo, por definição:

- **Ward**: Este método minimiza a soma das diferenças ao quadrado dentro de todos os clusters. Ele considera a soma dos quadrados das distâncias entre os pontos e os centróides dos clusters, buscando formar clusters que minimizem a variabilidade interna.
- **Complete**: A distância entre dois clusters é definida como a distância máxima entre qualquer par de pontos, onde cada ponto é de um cluster diferente.
- **Average**: É a distância entre dois clusters é a média das distâncias entre todos os pares de pontos em cada cluster.
- **Single**: A distância entre dois clusters é a distância mínima entre qualquer par de pontos, onde cada ponto é de um cluster diferente.

Para a comparação, foi desenvolvido um algoritmo de clusterização hierárquica aglomerativa (Agglomerative Clustering). Inicialmente, foram configurados quatro subplots em uma figura para exibir os resultados de cada método de linkage (ward, complete, average e single). Cada método foi aplicado ao conjunto de dados com o número de clusters definido como 7.

Os resultados da clusterização foram visualizados em gráficos de dispersão, onde as instâncias são coloridas de acordo com os rótulos dos clusters, figura 6. Além disso, os centros de cada cluster, calculados como a média dos valores das instâncias pertencentes a cada cluster, foram destacados em vermelho com um marcador de "+", facilitando a visualização das diferenças entre os métodos de linkage utilizados.

Figura 6: Diferentes métodos de linkage (ward, complete, average e single)

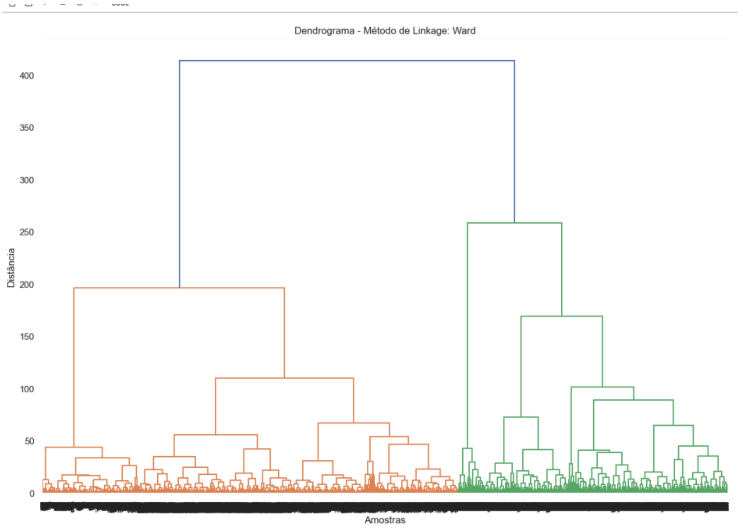


Autor: próprio

Os métodos de linkage (*ward* e *complete*), nessa ordem, apresentaram os melhores resultados no processo de classificação. É importante salientar que, em comparação ao K-means, o algoritmo de clusterização hierárquica utilizando a métrica *ward* obteve um desempenho muito semelhante, embora com um tempo de execução mais elevado nos testes.
Os métodos de linkage, ward e complete, não conseguiram realizar a clusterização de forma satisfatória. Os resultados não permitiram distinguir qual cluster cada ponto de dado pertence.

Ademais, para uma maior representação visual dos diferentes níveis de agrupamento. tens-se um dendrograma na figura 7, que visa além de ajuda a identificar o número de clusters apropriado, como também, ao observar a altura dos ramos onde ocorre a fusão de clusters.

Figura 7: Visualização dendrograma



Autor: próprio

6. Conclusão

Após a análise comparativa entre os métodos de agrupamento hierárquico (utilizando as métricas Ward e Complete) e o método K-means, com o objetivo de determinar o particionamento mais adequado dos dados em grupos naturais, concluiu-se que o método Ward conseguiu agrupar os dados de forma relativamente precisa, demonstrando um desempenho muito semelhante ao do K-means, embora com um tempo de execução mais elevado. Em contraste, o método Complete teve um desempenho menos eficiente na formação de clusters bem definidos em comparação ao método Ward e ao K-means.

Quanto à avaliação das métricas, constatou-se que a análise do cotovelo auxiliou na escolha do valor K mais apropriado para ambos os métodos hierárquicos e para o K-means. A análise da silhueta, utilizada para avaliar a coesão e a separação dos clusters formados, indicou que o método *Ward* tinha uma coesão e separação dos clusters comparável ao K-means, enquanto o método Complete mostrou uma performance inferior.

7. Referências

[1] STEINBACH, Michael; KARYPIS, George; KUMAR, Vipin. **A comparison of document clustering techniques**. 2000.

[2] FONSECA, Felipe Cesar Stanzani; BELTRAME, Walber Antônio Ramos. **Aplicações Práticas dos Algoritmos de Clusterização K-means e Bisecting K-means**. UFES, Vitória, 2010.

[3] Dry Bean. (2020). **UCI Machine Learning Repository**. https://doi.org/10.24432/C50S4B.