Implementação e Avaliação de Algoritmos de Clustering em Diversos Conjuntos de Dados

Artur F. Costa

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brasil

Abstract. *This paper describes the clustering problem, widely known and discussed in the field of computer science, as well as the implementation and empirical testing of three different algorithms. All of them try to approximate the optimal solution, but by three different approaches. Aside from their run-time, three other metrics were used to determine the quality of a given solution. The tests were performed in datasets obtained from three different sources. The metrics reflect the nature of these datasets with decent accuracy, highlighting the effectiveness of the algorithms under different conditions.*

Resumo. *Este artigo descreve o problema de agrupamento (clustering), amplamente conhecido e discutido dentro da ciência da computação, assim como a implementação e testes empíricos de três algoritmos diferentes. Além do tempo de execução, três outras métricas foram utilizadas para determinar a qualidade de uma dada solução. Os testes foram realizados em cima de conjuntos de dados obtidos de três fontes diferentes. As métricas refletem a natureza desses conjuntos com uma precisão razoável, destacando a efetividade desses algoritmos sob condições diferentes.*

1. Introdução

O problema de clustering, também conhecido como agrupamento, é uma tarefa fundamental na área de aprendizado de máquina e análise de dados. Refere-se ao processo de organizar um conjunto de objetos em grupos (clusters) de tal forma que os objetos pertencentes a um mesmo cluster sejam mais similares entre si do que em relação aos objetos de outros clusters. Essa similaridade é determinada por uma métrica de distância ou similaridade, que varia de acordo com o tipo de dados e com a aplicação dos resultados esperados.

Clustering é amplamente utilizado em diversas áreas, como reconhecimento de padrões, análise de imagens, segmentação de mercado, biologia computacional e mineração de dados. A capacidade de identificar padrões e estruturar grandes volumes de dados em clusters faz desse problema uma ferramenta poderosa para descobrir conhecimento implícito em bases de dados complexas. Os conjuntos de dados reais utilizados nos testes deste artigo são, em sua maioria, relacionados ao reconhecimento de padrões, mas também incluem dados de análise de imagens, biologia computacional e navegação.

Existem diferentes abordagens e algoritmos para resolver o problema de clustering, cada um com suas próprias vantagens e limitações. Os algoritmos implementados e testados neste trabalho incluem o K-Means, que busca minimizar a soma das distâncias

quadradas dentro dos clusters; o Furthest-First, que tenta maximizar a distância entre os pontos escolhidos como centros dos clusters; e o Binary-Search, que busca otimizar o tamanho dos clusters utilizando uma busca binária no intervalo inicial.

Neste artigo, busca-se explorar a implementação e a avaliação de diferentes algoritmos de clustering em vários conjuntos de dados, provindos de diferentes fontes. Ainda, tem-se como objetivo entender as vantagens e desvantagens de cada abordagem, assim como a eficácia dos algoritmos em situações variadas. A avaliação será conduzida utilizando métricas bem estabelecidas na literatura, e os resultados selecionados serão discutidos à luz das características dos dados e das especificidades de cada método.

2. Métricas

2.1. Raio

Como explicado anteriormente, um cluster é formado com base na similaridade entre diferentes instâncias, medida por uma métrica de distância. O raio de um cluster (ou raio máximo) é definido como a maior distância entre o centro do cluster e qualquer ponto pertencente a ele. Portanto, essa métrica representa a distância máxima de um ponto ao centro do próprio cluster. Em termos práticos, um raio menor sugere clusters mais compactos e bem definidos, com menor sobreposição entre os clusters.

2.2. Índice de Rand ajustado (ARI)

O índice de Rand ajustado (ARI) avalia a correspondência entre os rótulos de clusters previstos por um algoritmo e os rótulos verdadeiros conhecidos (ou *true labels*). Essa métrica só pode ser utilizada quando os rótulos verdadeiros dos dados são conhecidos, permitindo a comparação direta com os rótulos obtidos pelo algoritmo (*found labels*). O valor do ARI varia de -1 a 1, onde 1 indica uma correspondência perfeita entre os rótulos previstos e os verdadeiros, -1 indica uma total discordância, e valores próximos de 0 sugerem que a correspondência é tão ruim quanto uma atribuição aleatória.

2.3. Coeficiente de Silhueta

Diferentemente do ARI, o Coeficiente de Silhueta não requer os rótulos verdadeiros e avalia apenas os rótulos obtidos, Matematicamente, o Coeficiente de Silhueta s é definido como:

$$s = \frac{b - a}{\max(a, b)},$$

onde a é a distância média entre dois pontos de um mesmo cluster e b é distância média entre um ponto e os pontos do cluster mais próximo. Os valores variam de -1 a 1, onde 1 indica clusters bem definidos e densos, -1 sugere um agrupamento incorreto e valores próximos de 0 indicam clusters sobrepostos.

3. Implementação

3.1. *Furthest-First*

O algoritmo *Furthest-First* (abreviado para FF) é um dos mais simples e diretos. Após escolher um ponto arbitrário como o primeiro, basta buscar o ponto mais distante dos centros já escolhidos e adicioná-lo ao conjunto, até que hajam k centros. A implementação é dada abaixo:

```
S = conjunto de pontos
k = numero de clusters
Escolha um ponto arbitrario p em S
Remova p de S
C = {p}
ENQUANTO length(C) < k:
    Encontre q, o ponto mais distante dos pontos em C
    Remova q de S
    Adicione q a C
RETORNE C
```

Pseudocódigo 1. Algoritmo *Furthest-First*.

Por buscar os pontos mais distantes, o ponto forte do FF é identificar os centros corretamente com certa precisão, dependendo apenas do primeiro centro, que é escolhido arbitrariamente. Porém, em diversos casos, isso resulta em um raio relativamente maior.

3.2. Binary-Search

Sendo r^* o menor raio possível, sabe-se que $r^* \in [0, r_{\max}]$, onde $r_{\max}$ é a maior distância entre dois pontos do conjunto. Evidentemente, não se conhece r^*, mas é possível aproximá-lo com uma busca binária. Dado um intervalo $[a, b]$, o algoritmo *Binary-Search* (abreviado para BS) tenta produzir conjunto arbitrário de centros cujo raio seja $\frac{a+b}{2}$. Se esse conjunto existe e tem menos de k elementos, sabe-se que $r^* \in [a, \frac{a+b}{2}]$. Caso contrário, $r^* \in [\frac{a+b}{2}, b]$. A produção desse conjunto é dada abaixo:

```
S = conjunto de pontos
d = (a + b) / 2
k = numero de clusters
C = {}
ENQUANTO length(S) > 0:
    Escolha um ponto arbitrario p em S
    Adicione p ao conjunto C
    Remova de S todos os pontos tais que dist(p, s) <= d
SE length(C) <= k:
    RETORNE C
SENAO:
    RETORNE Falha
```

Pseudocódigo 2. Algoritmo *Binary-Search*.

Deve-se definir uma condição de parada, a fim de impor um limite ao tamanho da partição que o BS procura. O algoritmo foi implementado de maneira recursiva e lhe foi dada uma profundidade máxima a ser atingida, representando o número de chamadas recursivas. A largura relativa das partições buscadas, dada uma profundidade k, é dada por $2^{-k} \cdot 100\%$. A fim de comparar a largura da partição com o raio obtido, foram escolhidas as profundidades de 2 a 6, resultando em partições de tamanho $25\%, 12.5\%, 6.25\%, 3.12\%$ e 1.56%, respectivamente.

O ponto forte desse algoritmo, dada uma profundidade de busca razoável, é encontrar um raio relativamente pequeno. No entanto, como os centros são escolhidos arbitrariamente a cada passo, a precisão na identificação desses centros é relativamente mais baixa.

3.3. *K-Means*

O algoritmo *K-Means* (abreviado para KM) inicialmente escolhe centros arbitrários e, num processo iterativo, altera a posição desses centros a fim de que eles convirjam para bons centros. A sua implementação é dada abaixo:

```
S = conjunto de pontos
k = numero de clusters
Escolha k pontos arbitrarios em S como os centros iniciais dos
   clusters
ENQUANTO mudancas forem significativas:
    Para cada ponto p em S:
        Atribua p ao cluster mais proximo
    Para cada cluster:
        Recalcule o centro do cluster como a media dos pontos
           atribuidos a ele
RETORNE C
```

Pseudocódigo 3. Algoritmo *K-Means*.

Essa iteração termina quando as mudanças de uma iteração para outra não são mais significativas. Isso ocorre quando não há mais mudanças nas atribuições de pontos ou quando a mudança na posição dos centros está abaixo de um certo limite definido na implementação. Ainda, pode-se definir um número máximo de iterações.

Pode-se perceber como o KM é diferente dos dois algoritmos anteriores, já que os centros podem ser pontos que não pertencem ao conjunto original. Isso permite que a precisão na identificação seja maior, já que o centro está sempre sendo recalculado como o ponto médio do cluster, diminuindo também o raio daquele cluster.

Ainda, diferentemente do *Furthest-First* e do *Binary-Search*, foi utilizada uma implementação deste algoritmo disponibilizada pelo *Scikit-Learn*, acessível em [Pedregosa et al. 2011a].

4. Conjuntos de Dados

4.1. Repositório *UCI Machine Learning*

Esta seção cobre os conjuntos de dados reais, obtidos do *Machine Learning Repository* da *University of California, Irvine*, acessíveis em [UCI 2024]. Foram selecionados dez conjuntos de dados cujos atributos são numéricos ou categóricos, permitindo que as instâncias sejam tratadas como n-uplas e que o problema de clustering seja aplicado a esses pontos n-dimensionais.

Anuran Calls

Este conjunto possui 7195 instâncias, com 22 atributos cada. Os dados são características acústicas dos chamados de diversas espécies de sapos da ordem *Anura*. A área deste conjunto é reconhecimento de padrões. Acessível em [Amaral et al. 2024].

Indoor Localizaton and Navigation

Este conjunto possui 1420 instâncias, com 13 atributos cada. O conjunto original tem mais instâncias, mas muitas delas estão incompletas. Os dados são leituras de sinais provindos de sinalizadores *Bluetooth*. As áreas deste conjunto incluem localização e navegação. Acessível em [Angeloudis et al. 2024].

Facebook Live Sellers in Thailand

Este conjunto possui 7050 instâncias, com 9 atributos cada. Os dados são estatísticas de postagens no *Facebook*, incluindo comentários, reações e compartilhamentos. A área deste conjunto é reconhecimento de padrões. Acessível em [Sitasuwan and Chai 2024].

Mice Protein Expression

Este conjunto possui 1080 instâncias, com 80 atributos cada. Os dados são níveis de expressão de diversas proteínas no córtex cerebral de diversas espécies de ratos. A área deste conjunto é reconhecimento de padrões. Acessível em [Moradian and Grivell 2024].

GNFUV Unmanned Surface Vehicles Sensor Data

Este conjunto possui 1649 instâncias, com 3 atributos cada. Os dados são leituras de temperatura e umidade, além da hora, coletados por diversos sensores móveis. A área deste conjunto é reconhecimento de padrões. Acessível em [Stewart and Deverge 2024].

Estimation of Obesity Levels Based On Eating Habits and Physical Condition

Este conjunto possui 2111 instâncias, com 16 atributos cada. Os dados são respostas a uma pesquisa sobre hábitos alimentícios e condição física. Apenas 23% dos dados são reais, enquanto os outros 77% são sintéticos, gerados com base nos reais. A área deste conjunto é reconhecimento de padrões. Acessível em [Solorio and Rivas 2024].

Statlog (Vehicle Silhouettes)

Este conjunto possui 846 instâncias, com 18 atributos cada. Os dados são diversas medidas das silhuetas de diferentes veículos. A área deste conjunto é análise de imagens. Acessível em [Baek and Banfield 2024].

Website Phishing

Este conjunto possui 1353 instâncias, com 9 atributos cada. Os dados são diversas medidas relacionadas à legitimidade de vários websites. A área deste conjunto é reconhecimento de padrões. Acessível em [Aburrous and Hossain 2024].

Wireless Indoor Localization

Este conjunto possui 2000 instâncias, com 7 atributos cada. Os dados são diversas leituras do sinal de uma rede *WiFi*, capturadas por um *smartphone*. A área deste conjunto é localização. Acessível em [Ermiş and Ermiş 2024].

Yeast

Este conjunto possui 1484 instâncias, com 8 atributos cada. Os dados são resultados de diversos testes realizados em cima de diferentes proteínas. A área deste conjunto é localização. Acessível em [Horton and Nakai 2024].

4.2. *Scikit-Learn*

Esta seção cobre a primeira parte dos dados sintéticos, que foram gerados pelo gerador desenvolvido pelo *Scikit-Learn*, acessível em [Pedregosa et al. 2011b], e serão referenciados como conjuntos SKG. Esse gerador oferece diversas funções para a criação de pontos bidimensionais, como círculos, luas e aglomerados. Foram gerados dez conjuntos de 500 pontos cada, utilizando diferentes parâmetros para essas funções. Os conjuntos estão disponíveis no repositório do *Github*, acessível em [Costa 2024b], no diretório `src/datasets/sk-generated`. O código utilizado para gerar esses conjuntos `src/sk-generator.py` também está disponível no repositório.

4.3. Distribuição Normal Multivariada

Esta seção cobre a segunda parte dos dados sintéticos, que foram gerados por meio de uma distribuição normal multivariada, e serão referenciados como conjuntos NDG. Foram criados dez conjuntos de 500 pontos cada, utilizando diferentes parâmetros, como o número de clusters, a posição de cada centro, os desvios-padrão e a quantidade de pontos gerados em torno de cada centro. O primeiro conjunto contém clusters densos e bem definidos, enquanto os conjuntos subsequentes apresentam clusters gradualmente mais sobrepostos. Os dados estão disponíveis no repositório do *Github*, acessível em [Costa 2024b], no diretório `src/datasets/nd-generated`. O código utilizado para gerar esses conjuntos, disponível no arquivo `src/nd-generator.py`, também está no repositório.

5. Testes

Devido à natureza arbitrária dos algoritmos implementados, cada um foi executado 30 vezes em cada um dos conjuntos de dados citados na Seção 4. Os resultados individuais de cada experimento, bem como um resumo com as médias e os desvios-padrão de cada métrica, estão disponíveis em uma planilha online, acessível em [Costa 2024a]. Foram selecionados oito conjuntos de dados para ilustrar as principais diferenças apontadas pelas métricas, considerando a variedade dos dados. Destaca-se que os resultados apresentados são referentes à distância Euclidiana (distância de Minkowski com $p = 2$).

Nota-se que o tempo de execução do algoritmo KM sempre será mais alto, já que ele precisa calcular a matriz de distâncias. Em contraste, os algoritmos FF e BS foram implementados do zero, de forma que apenas utilizam uma matriz de distâncias calculada previamente.

5.1. *Mice Protein Expression*

Para boa parte dos conjunto do repositório *UCI*, incluindo o conjunto *Mice Protein Expression*, os dados não são próprios para o agrupamento. Isso pode ser visto na figura 1, onde estão apresentados os resultados dos testes, que todos os algoritmos tiveram uma performance bem baixa. Para o FF e o BS, vê-se valores de ARI extremamente próximos de 0, indicando uma classificação tão boa quanto uma rotulação aleatória. Da mesma

forma, nota-se um coeficiente de silhueta baixíssimo, sugerindo uma sobreposição significativa nos clusters. Nota-se que o raio do algoritmo BS é o mais baixo dentre todos, a partir de uma busca com profundidade 3. Destaca-se que o algoritmo KM teve um desempenho superior quanto ao ARI e à silhueta, fora os desvios-padrão também menores, mas o raio foi significativamente maior do que os outros, coisa que ocorreu na maioria dos conjuntos do repositório *UCI*.

Algoritmo	Raio (AVG)	Raio (STDEV)	Tempo (AVG)	Tempo (STDEV)	ARI (AVG)	ARI (STDEV)	Silhueta (AVG)	Silhueta (STDEV)
Binary-Search (profundidade 2)	4.757	0.444	0.005	0.001	0.134	0.045	0.122	0.022
Binary-Search (profundidade 3)	3.900	0.218	0.008	0.004	0.096	0.040	0.138	0.031
Binary-Search (profundidade 4)	3.770	0.210	0.009	0.002	0.090	0.037	0.139	0.023
Binary-Search (profundidade 5)	3.732	0.148	0.009	0.003	0.099	0.027	0.146	0.023
Binary-Search (profundidade 6)	3.684	0.151	0.011	0.002	0.088	0.031	0.145	0.021
Furthest-First	4.019	0.147	0.008	0.004	0.058	0.027	0.145	0.025
K-Means	5.772	0.206	0.229	0.028	0.209	0.016	0.178	0.008

Figura 1

5.2. *Wireless Indoor Localization*

O conjunto *Wireless Indoor Localization* é a única exceção dentro do repositório *UCI*. Isso pode ser visto na figura 2, onde estão apresentados os resultados dos testes deste conjunto, que todos os algoritmos tiveram uma performance no mínimo razoável. Para o FF e o BS, vê-se valores aceitáveis para o ARI e para a silhueta, sugerindo uma rotulação mais precisa e uma sobreposição bem menor nos clusters. Novamente, o BS teve o menor raio médio, que curiosamente foi atingido na profundidade 4. Porém, neste conjunto, os raios encontrados por FF e KM estão bem mais próximos deste valor mínimo. Por fim, o algoritmo KM teve um desempenho excelente quanto ao ARI e um desempenho levemente melhor quanto à silhueta.

Algoritmo	Raio (AVG)	Raio (STDEV)	Tempo (AVG)	Tempo (STDEV)	ARI (AVG)	ARI (STDEV)	Silhueta (AVG)	Silhueta (STDEV)
Binary-Search (profundidade 2)	31.448	2.152	0.016	0.004	0.483	0.134	0.309	0.057
Binary-Search (profundidade 3)	31.093	2.033	0.024	0.006	0.514	0.111	0.320	0.061
Binary-Search (profundidade 4)	29.644	1.534	0.025	0.004	0.520	0.092	0.330	0.034
Binary-Search (profundidade 5)	29.781	1.764	0.033	0.007	0.519	0.109	0.327	0.040
Binary-Search (profundidade 6)	30.222	1.724	0.034	0.005	0.472	0.128	0.309	0.058
Furthest-First	33.753	1.553	0.006	0.001	0.429	0.150	0.302	0.059
K-Means	35.508	0.768	0.149	0.028	0.878	0.052	0.406	0.021

Figura 2

5.3. Conjunto SKG-2

A figura 3 mostra a distribuição dos pontos do conjunto, em duas meias-luas levemente sobrepostas. Foram buscados dois clusters e, como se pode observar na figura 4, os clusters encontrados pelos três algoritmos não apresentam uma sobreposição muito significativa, apesar da rotulação, em média, ter sido ruim. Nota-se, porém, que o coeficiente de variação $\left(\text{razão } \frac{\text{desvio-padrão}}{\text{média}}\right)$ para o FF e o BS é bem alto, destacando a natureza aleatória desses dois algoritmos. Neste conjunto, o menor raio médio pertence ao KM, destacando a vantagem de posicionar o centro dos clusters fora do conjunto de pontos.

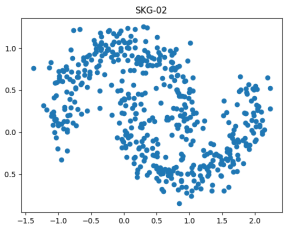

Figura 3

Algoritmo	Raio (AVG)	Raio (STDEV)	Tempo (AVG)	Tempo (STDEV)	ARI (AVG)	ARI (STDEV)	Silhueta (AVG)	Silhueta (STDEV)
Binary-Search (profundidade 2)	1.653	0.136	0.002	0.001	0.249	0.098	0.449	0.030
Binary-Search (profundidade 3)	1.623	0.176	0.002	0.001	0.240	0.087	0.439	0.035
Binary-Search (profundidade 4)	1.562	0.133	0.003	0.001	0.231	0.097	0.432	0.031
Binary-Search (profundidade 5)	1.572	0.132	0.003	0.001	0.232	0.066	0.437	0.034
Binary-Search (profundidade 6)	1.549	0.164	0.003	0.001	0.234	0.099	0.426	0.040
Furthest-First	1.757	0.167	0.001	0.001	0.196	0.061	0.414	0.034
K-Means	1.230	0.001	0.039	0.033	0.249	0.004	0.472	0.000

Figura 4

5.4. Conjunto SKG-4

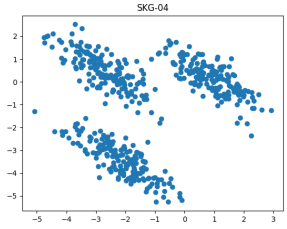

Figura 5

Algoritmo	Raio (AVG)	Raio (STDEV)	Tempo (AVG)	Tempo (STDEV)	ARI (AVG)	ARI (STDEV)	Silhueta (AVG)	Silhueta (STDEV)
Binary-Search (profundidade 2)	3.495	0.396	0.002	0.001	0.784	0.141	0.568	0.063
Binary-Search (profundidade 3)	3.343	0.417	0.003	0.001	0.827	0.116	0.585	0.053
Binary-Search (profundidade 4)	3.315	0.384	0.003	0.001	0.830	0.111	0.584	0.063
Binary-Search (profundidade 5)	3.198	0.250	0.003	0.001	0.843	0.125	0.590	0.058
Binary-Search (profundidade 6)	3.291	0.282	0.004	0.001	0.827	0.121	0.588	0.051
Furthest-First	4.018	0.257	0.001	0.000	0.650	0.191	0.500	0.103
K-Means	2.974	0.004	0.037	0.024	0.950	0.003	0.629	0.000

Figura 6

A figura 5 mostra a distribuição dos pontos do conjunto, em três listas bem separadas. Foram buscados três clusters e, como visto na figura 6, os clusters encontrados pelos três algoritmos apresentam uma baixa sobreposição e rotulação, desta vez, foi bem precisa. O coeficiente de variação do ARI é bem menor para o FF e o BS, mostrando que mesmo uma escolha arbitrária acaba levando às três listras na maioria dos casos. Novamente, o raio médio mais baixo pertence ao KM, que obteve um desvio-padrão quase nulo.

5.5. Conjunto SKG-8

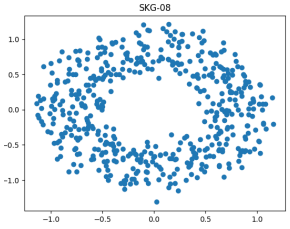

Figura 7

Algoritmo	Raio (AVG)	Raio (STDEV)	Tempo (AVG)	Tempo (STDEV)	ARI (AVG)	ARI (STDEV)	Silhueta (AVG)	Silhueta (STDEV)
Binary-Search (profundidade 2)	1.524	0.120	0.001	0.000	-0.001	0.001	0.371	0.011
Binary-Search (profundidade 3)	1.513	0.092	0.002	0.001	-0.001	0.001	0.369	0.012
Binary-Search (profundidade 4)	1.504	0.094	0.002	0.001	-0.001	0.001	0.369	0.009
Binary-Search (profundidade 5)	1.498	0.087	0.002	0.001	-0.001	0.001	0.370	0.008
Binary-Search (profundidade 6)	1.477	0.114	0.003	0.001	-0.001	0.001	0.373	0.006
Furthest-First	1.593	0.071	0.000	0.000	-0.001	0.001	0.367	0.012
K-Means	1.298	0.060	0.027	0.022	-0.002	0.000	0.381	0.004

Figura 8

A figura 7 mostra a distribuição dos pontos do conjunto, em um único anel levemente oval e uniformemente distribuído, onde foram buscados dois clusters. Como é possível ver na figura 8, os clusters encontrados pelos três algoritmos, apesar de indicarem sobreposição razoável, a rotulação foi tão boa quanto uma aleatória, resultando em valores negativos muito baixos. Isso se deve ao fato dos centros encontrados terem sido majoritariamente próximos do centro do anel, englobando pontos de ambos os clusters originais. Assim, como a figura é simétrica, os dois centros englobam aproximadamente metade dos pontos e a rotulação apresenta o caráter aleatório. Mais uma vez, o raio médio mais baixo pertence ao KM, apesar de todos terem tido raios bem próximos.

5.6. Conjunto NDG-1

A figura 9 mostra a distribuição dos pontos do conjunto, quatro aglomerados muito bem definidos, onde foram buscados quatro clusters. Como é possível ver na figura 10, os clusters foram facilmente encontrados pelos três algoritmos, indicando uma sobreposição quase inexistente e uma rotulação perfeita. Mais uma vez, o raio médio mais baixo pertence ao KM. Mesmo tendo identificado os clusters perfeitamente, os algoritmos FF e BS selecionam um ponto daquele cluster como centro, resultando em um raio ligeiramente maior. Enquanto isso, a cada iteração, o KM define o centro como a média dos pontos do cluster, resultando em um raio extremamente próximo do raio ótimo.

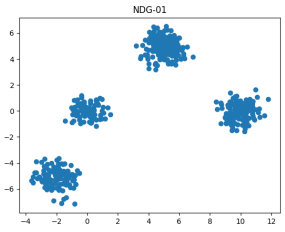

Figura 9

Algoritmo	Raio (AVG)	Raio (STDEV)	Tempo (AVG)	Tempo (STDEV)	ARI (AVG)	ARI (STDEV)	Silhueta (AVG)	Silhueta (STDEV)
Binary-Search (profundidade 2)	3.001	0.365	0.003	0.001	1.000	0.000	0.819	0.000
Binary-Search (profundidade 3)	2.915	0.347	0.003	0.001	1.000	0.000	0.819	0.000
Binary-Search (profundidade 4)	2.876	0.254	0.004	0.001	1.000	0.000	0.819	0.000
Binary-Search (profundidade 5)	2.886	0.282	0.005	0.001	1.000	0.000	0.819	0.000
Binary-Search (profundidade 6)	2.724	0.238	0.006	0.001	1.000	0.000	0.819	0.000
Furthest-First	3.570	0.223	0.001	0.000	1.000	0.001	0.819	0.001
K-Means	2.380	0.000	0.055	0.024	1.000	0.000	0.819	0.000

Figura 10

5.7. Conjunto NDG-4

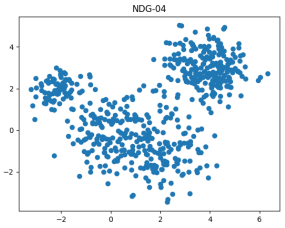

Figura 11

Algoritmo	Raio (AVG)	Raio (STDEV)	Tempo (AVG)	Tempo (STDEV)	ARI (AVG)	ARI (STDEV)	Silhueta (AVG)	Silhueta (STDEV)
Binary-Search (profundidade 2)	3.284	0.242	0.003	0.001	0.632	0.109	0.432	0.052
Binary-Search (profundidade 3)	3.311	0.269	0.003	0.001	0.612	0.113	0.423	0.059
Binary-Search (profundidade 4)	3.286	0.282	0.003	0.001	0.664	0.088	0.454	0.046
Binary-Search (profundidade 5)	3.214	0.240	0.004	0.001	0.683	0.074	0.464	0.034
Binary-Search (profundidade 6)	3.175	0.252	0.005	0.001	0.668	0.101	0.452	0.052
Furthest-First	3.676	0.278	0.001	0.000	0.551	0.134	0.395	0.072
K-Means	2.693	0.133	0.063	0.032	0.784	0.082	0.517	0.034

Figura 12

A figura 11 mostra a distribuição dos pontos do conjunto, em quatro aglomerados um pouco mais esparsos e sobrepostos, onde foram buscados quatro clusters. Como pode-se observar na figura 12, os clusters encontrados pelos três algoritmos indicam uma sobreposição baixa e uma boa rotulação, apesar do coeficiente de variação do ARI ter sido maior para o FF e o BS, novamente devido à natureza aleatória de ambos. Novamente, o raio médio mais baixo pertence ao KM, apesar de todos terem encontrado raios bem próximos.

5.8. Conjunto NDG-10

A figura 13 mostra a distribuição dos pontos do conjunto, em um retângulo uniformemente distribuído, onde foram buscados quatro clusters. Como era esperado, segundo a figura 14, os clusters encontrados por FF e BS apresentaram sobreposição e rotulação razoáveis. O algoritmo KM teve um desempenho superior, mas nada muito acima dos outros. Desta vez, como os clusters são mais esparsos, o raio médio mais baixo pertence ao BS, que ultrapassou o KM já com profundidade 3, apesar de ter tido um coeficiente de variação muito maior.

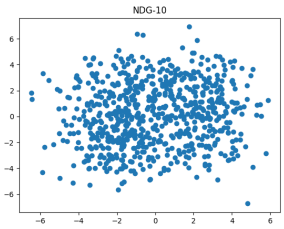

Figura 13

Algoritmo	Raio (AVG)	Raio (STDEV)	Tempo (AVG)	Tempo (STDEV)	ARI (AVG)	ARI (STDEV)	Silhueta (AVG)	Silhueta (STDEV)
Binary-Search (profundidade 2)	5.666	0.724	0.003	0.001	0.351	0.078	0.288	0.042
Binary-Search (profundidade 3)	5.417	0.395	0.004	0.001	0.344	0.075	0.297	0.038
Binary-Search (profundidade 4)	5.270	0.496	0.005	0.001	0.347	0.072	0.295	0.033
Binary-Search (profundidade 5)	5.370	0.452	0.006	0.002	0.348	0.080	0.292	0.042
Binary-Search (profundidade 6)	5.214	0.398	0.007	0.003	0.368	0.062	0.306	0.026
Furthest-First	6.134	0.319	0.001	0.000	0.229	0.093	0.236	0.052
K-Means	5.847	0.106	0.078	0.033	0.499	0.017	0.367	0.003

Figura 14

6. Conclusão

Neste trabalho, foi explorada e avaliada a eficácia de três algoritmos de clustering: K-Means, Furthest-First e Binary-Search, aplicados a conjuntos de dados variados provenientes de diferentes fontes. Os algoritmos foram avaliados com base em métricas de desempenho, incluindo o Índice de Rand Ajustado, o Raio da Solução e o Coeficiente de Silhueta.

Os resultados dos experimentos mostraram que cada algoritmo possui características distintas que o tornam mais ou menos adequado para diferentes tipos de dados. O K-Means, por exemplo, demonstrou uma excelente performance em cenários com clusters bem definidos, porém enfrentou dificuldades com clusters sobrepostos e mais esparsos. O Furthest-First mostrou uma abordagem interessante ao tentar maximizar a distância entre os pontos durante a formação dos clusters, o que pode ser vantajoso para certos tipos de dados, identificando corretamente clusters que são mais distantes. Já o Binary-Search usou uma abordagem clássica da busca binária para determinar o menor raio dos clusters, obtendo um desempenho tão bom quanto o Furthest-First, mas não tão bom quanto o K-Means, em geral.

A análise dos dados, incluindo aqueles gerados sinteticamente e obtidos de repositórios públicos, evidenciou a importância da escolha adequada do algoritmo de clustering em função das características específicas dos dados. Os conjuntos de dados analisados apresentaram uma ampla gama de complexidades, desde clusters bem definidos até sobrepostos, o que evidenciou a necessidade de métodos diferentes para cada tipo de dado.

Referências

Aburrous, M. and Hossain, A. (2024). Website phishing. https://archive.ics.uci.edu/dataset/379/website+phishing. Acessado em: 5 agosto, 2024.

Amaral, V. V. P., Souza, F. L., and Corrêa, J. L. C. (2024). Anuran calls (mfccs). https://archive.ics.uci.edu/dataset/406/anuran+calls+mfccs. Acessado em: 5 agosto, 2024.

Angeloudis, D., Angeloudis, P., Roberts, C., and Kuo, Y. K. (2024). Ble rssi dataset for indoor localization and navigation. https://archive.ics.uci.edu/dataset/435/ble+rssi+dataset+for+indoor+localization+and+navigation. Acessado em: 5 agosto, 2024.

Baek, S. and Banfield, G. (2024). Statlog (vehicle silhouettes). https://archive.ics.uci.edu/dataset/149/statlog+vehicle+silhouettes. Acessado em: 5 agosto, 2024.

Costa, A. F. (2024a). Planilha com resultados completos dos experimentos de clustering. https://docs.google.com/spreadsheets/d/1WQZZlBaH7Vg1k0moSk3QbJ0spR3cPWLapmiNRLNooYA/edit?usp=sharing. Acessado em: 14 agosto 2024.

Costa, A. F. (2024b). Repositório clustering. https://github.com/Tuzass/clustering. Acessado em: 14 agosto, 2024.

Ermiş, A. and Ermiş, I. (2024). Wireless indoor localization. https://archive.ics.uci.edu/dataset/422/wireless+indoor+localization. Acessado em: 5 agosto, 2024.

Horton, P. and Nakai, K. (2024). Yeast. https://archive.ics.uci.edu/dataset/110/yeast. Acessado em: 5 agosto, 2024.

Moradian, P. H. and Grivell, J. F. (2024). Mice protein expression. https://archive.ics.uci.edu/dataset/342/mice+protein+expression. Acessado em: 5 agosto, 2024.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011a). Scikit-learn: Machine learning in Python. https://scikit-learn.org/stable/. Acessado em: 11 agosto 2024.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011b). Scikit-learn: Machine learning in Python - cluster comparison example. https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html. Acessado em: 8 agosto 2024.

Sitasuwan, A. and Chai, R. (2024). Facebook live sellers in thailand. https://archive.ics.uci.edu/dataset/488/facebook+live+sellers+in+thailand. Accessado em: 5 agosto, 2024.

Solorio, I. S. and Rivas, M. A. C. (2024). Estimation of obesity levels based on eating habits and physical condition. https://archive.ics.uci.edu/dataset/544/estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition. Acessado em: 5 agosto, 2024.

Stewart, C. V. and Deverge, N. (2024). Gnfuv unmanned surface vehicles sensor data. https://archive.ics.uci.edu/dataset/452/gnfuv+unmanned+surface+vehicles+sensor+data. Acessado em: 5 agosto, 2024.

UCI (2024). Machine learning repository. https://archive.ics.uci.edu/datasets. Acessado em: 5 agosto, 2024.