Algoritmos Aproximativos para o Problema dos k-Centros

André Alves de Souza Barros[1], **Ramiro Noronha Reis Ribeiro**[1]

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

{andreasb,ramironrribeiro}@ufmg.br

Abstract. *This paper addresses the k-Center Problem, a data clustering technique that involves partitioning data into k groups, which is considered NP-hard. Two 2-approximation algorithms are presented for the problem, ensuring a solution where the maximum radius is at most twice the value of the optimal solution. The first algorithm uses a binary search to refine the ideal radius, while the second adopts a greedy approach, selecting points with the greatest distance from each other as centers. The algorithms were evaluated using real and synthetic datasets, and the results indicate that the first algorithm offers higher accuracy, while the second provides faster performance.*

Resumo. *Este artigo aborda o Problema dos k-Centros, uma técnica de particionamento de dados em k grupos, considerado NP-difícil. São apresentados dois algoritmos aproximativos 2-aproximados para o problema, que garantem uma solução onde o raio máximo é no máximo duas vezes o valor da solução ótima. O primeiro algoritmo utiliza uma busca binária para refinar o valor do raio ideal, enquanto o segundo adota uma abordagem gulosa, selecionando pontos com maior distância entre si como centros. A avaliação dos algoritmos foi realizada com bases de dados reais e sintéticas, e os resultados indicam que o primeiro algoritmo proporciona maior precisão e o segundo maior velocidade.*

1. Introdução

O método de k-Centros é uma técnica de mineração de dados usada para particionar n observações em k grupos. Esse processo envolve a escolha de k centros e a subsequente atribuição de cada ponto ao grupo correspondente ao centro mais próximo, o que resulta em um diagrama de Voronoi no espaço dos dados.

O problema de k-Centros, assim como demonstrado por Mahajan, é classificado como NP-difícil, o que significa que não há algoritmos conhecidos que possam encontrar a solução ótima em tempo polinomial. No entanto, como discutiremos neste artigo, existem algoritmos aproximativos que, em tempo polinomial, conseguem produzir soluções onde a maior distância de um ponto ao seu centro é, no máximo, duas vezes a maior distância observada na solução ótima.

2. Métodos e métricas

2.1. Métodos

Neste artigo, serão apresentados dois algoritmos aproximativos, ambos 2-aproximados, para o problema de k-Centros. O primeiro algoritmo assume inicialmente que o valor

ótimo de r (a maior distância entre um ponto e seu centro) é conhecido. O algoritmo então seleciona arbitrariamente pontos para serem centros, removendo os pontos que estão a uma distância de até $2r$ desses centros, repetindo o processo até que nenhum ponto reste. Se o número de centros escolhidos exceder k, conclui-se que não há solução viável para esse r e k.

Esse algoritmo tem complexidade de tempo $O(n^2)$, onde n é o número de pontos, pois, no pior caso, podemos ter até n centros, e, para cada um deles, é necessário iterar sobre todos os demais pontos. É garantido, por construção, que se o algoritmo retornar um agrupamento C, então $r(C) \leq 2r$. Se mais de k centros forem encontrados, pode-se provar, por contradição, que para qualquer solução C* com até k centros, $r(C*) > r$. Isso ocorre porque, pelo princípio da casa dos pombos, haveria pelo menos dois centros da solução aproximativa pertencendo ao mesmo cluster na solução ótima. Nesse caso, teríamos $distância(c, c*) + distância(c, c*) \leq 2r < distância(c, c)$, o que contradiz a hipótese de que a distância é uma métrica e implicando $|C*| \geq |C| > k$.

Além disso, é possível superar a limitação de não conhecer o valor da solução ótima utilizando outro algoritmo que, através de uma sequência de tentativas, refina o valor de r até a convergência. Sabe-se que o valor ótimo de r está entre 0 e r_{max} (maior distância entre dois pontos). O algoritmo anterior é executado inicialmente com $\frac{r_{max}}{2}$. Se os dados puderem ser agrupados em até k centros, o algoritmo é executado novamente com $\frac{r_{max}}{4}$; caso contrário, é executado com $\frac{3r_{max}}{4}$. Esse processo continua em uma busca binária até que o algoritmo convirja, com fator de aproximação 2.

O segundo algoritmo, por sua vez, não depende do valor ótimo de r. Ele funciona da seguinte maneira: inicialmente, um ponto arbitrário é escolhido e adicionado ao conjunto de centros. Em seguida, até que se tenha k centros, o próximo centro é escolhido como o ponto com a maior distância em relação aos centros já selecionados.

Esse algoritmo guloso garante um agrupamento cujo raio é, no máximo, duas vezes o valor ótimo. A ideia da prova é a seguinte: suponha que a solução ótima tenha um raio r* e que a solução encontrada pelo algoritmo tenha um raio $r > 2r*$. Nesse caso, como os centros são escolhidos entre os pontos de S, deve haver pelo menos $k + 1$ pontos com uma distância mínima de r entre si, do contrário teríamos a solução ótima. Na solução ótima, dois desses pontos estariam no mesmo cluster, implicando em um diâmetro maior que $2r*$ para esse cluster, o que contradiz a suposição de que o raio ótimo é $r*$. Assim, temos que $r \leq 2r*$.

2.2. Métricas

As métricas utilizadas para avaliação dos algoritmos foram: o valor de r em cada execução, o tempo de duração da execução, o índice de silhueta e o índice Rand ajustado. O índice de silhueta avalia a similaridade de cada ponto com os outros pontos de seu cluster, variando de -1 a 1, onde valores próximos a 1 indicam uma boa coesão interna. Já o índice Rand ajustado mede a similaridade entre os clusters provenientes do algoritmo e os rótulos originais dos dados, ajustado para a probabilidade de similaridade ocorrer por acaso, com valores variando de -1 a 1, onde 1 indica uma correspondência perfeita.

3. Implementação

A implementação dos algoritmos foi realizada em Python, utilizando técnicas de compilação just-in-time proporcionadas pela biblioteca Numba para melhorar a eficiência computacional. O código foi estruturado em várias etapas, que incluem a preparação dos dados, a definição dos algoritmos, a execução e a avaliação dos resultados. A avaliação dos algoritmos conduzida por meio das métricas de silhueta e Rand ajustado foi realizada empregando as funções disponíveis na biblioteca scikit-learn. As funções de k-Means recebem como parâmetro um valor p, que é utilizado no cálculo da distância de Minkowski empregada pelo algoritmo.

4. Experimentos

Para testar os algoritmos, foram utilizadas três tipos de bases de dados. A primeira foi composta por dados reais, extraídos do UCI Machine Learning Repository. Em seguida, utilizamos dados sintéticos 2D *interessantes*, gerados com a biblioteca sklearn, seguindo o exemplo disponível na documentação oficial, permitindo a avaliação dos algoritmos em cenários variados, com diferentes padrões de agrupamento. Por fim, foram gerados dados sintéticos próprios, onde os centros dos clusters foram sorteados aleatoriamente, e os pontos ao redor de cada centro foram gerados com uma distribuição normal multivariada. O desvio padrão dessa distribuição variou entre 0,01 (dados bem separados) e 0,1 (dados com grande sobreposição), permitindo testar a robustez dos algoritmos em cenários com diferentes graus de complexidade.

Para cada tipo de dado, foram testadas 10 instâncias distintas. Cada algoritmo foi executado 60 vezes em cada instância, sendo 30 execuções com a distância euclidiana e 30 com a distância de Manhattan, a fim de abater a parte randômica da escolha dos pontos. As médias e desvios padrão das métricas foram calculadas a partir desses resultados. Além dos algoritmos implementados, utilizou-se também o algoritmo k-means da biblioteca sklearn como base de comparação. Para a métrica de Manhattan, utilizou-se o algoritmo k-medoids da mesma biblioteca, que difere do k-means padrão por, assim como os algoritmos aproximativos apresentados, escolher seus centros entre os pontos do conjunto de dados.

5. Resultados

5.1. Análise dos resultados

Para os diferentes tipos de bases de dados, o primeiro algoritmo demonstrou um desempenho superior na obtenção de menores valores de r, o que indica uma maior precisão na aproximação do valor ótimo. Por outro lado, o segundo algoritmo destacou-se por ser o mais rápido entre os três e pela consistência nos resultados, com valores de r geralmente situados entre os obtidos pelo primeiro algoritmo e pelo algoritmo da biblioteca scikit-learn. Embora o algoritmo do scikit-learn tenha sido o mais lento e tenha tendido a apresentar os maiores valores de r, ele compensou isso com uma consistência notável nos resultados e melhor desempenho em termos do índice Rand ajustado, refletindo uma maior correspondência com os rótulos originais dos dados.

Além disso, observou-se que o algoritmo disponível no scikit-learn não apresenta um tempo de execução proporcional ao número de pontos, sugerindo que ele é mais influenciado pela dimensionalidade do espaço, em contraste com os algoritmos implementados

Tabela 1. Primeiro algoritmo (por busca binária de r) - para bases de dados reais

Instância	p	Centro	Tamanho	Raio ± std	Silhouette ± std	Rand ± std	Tempo ± std
ObesityDataSet_raw_and_data_sinthetic	1	7	2111	31.63±1.64	0.35±0.04	0.28±0.02	0.08±0.00
ObesityDataSet_raw_and_data_sinthetic	2	7	2111	20.59±1.45	0.37±0.05	0.29±0.02	0.08±0.00
Raisin_Dataset	1	2	900	118786.33±25080.31	0.60±0.07	0.17±0.10	0.02±0.00
Raisin_Dataset	2	2	900	84012.89±19331.37	0.61±0.05	0.19±0.11	0.02±0.00
abalone	1	28	4176	0.55±0.01	0.26±0.04	0.07±0.00	0.32±0.02
abalone	2	28	4176	0.28±0.01	0.28±0.05	0.07±0.00	0.31±0.01
data_banknote_authentication	1	2	1371	22.23±2.53	0.38±0.07	0.06±0.02	0.03±0.00
data_banknote_authentication	2	2	1371	12.64±1.77	0.37±0.06	0.05±0.02	0.04±0.01
flare	1	8	1065	3.00±0.00	0.26±0.04	0.08±0.04	0.02±0.00
flare	2	8	1065	2.02±0.17	0.35±0.05	0.08±0.06	0.03±0.01
magic04	1	2	19019	1105.53±60.63	0.61±0.13	0.02±0.02	4.66±0.83
magic04	2	2	19019	531.68±42.13	0.62±0.12	0.01±0.01	4.55±0.79
poker-hand-training-true	1	10	25009	27.52±0.63	0.06±0.01	0.00±0.00	7.81±1.28
poker-hand-training-true	2	10	25009	12.50±0.25	0.09±0.01	-0.00±0.00	8.03±0.57
transfusion	1	2	748	3497.47±836.14	0.75±0.07	0.05±0.02	0.01±0.00
transfusion	2	2	748	3681.28±930.35	0.78±0.08	0.05±0.02	0.01±0.00
winequality-white	1	7	6497	103.61±8.09	0.33±0.07	-0.00±0.00	0.77±0.03
winequality-white	2	7	6497	70.98±4.77	0.38±0.07	-0.00±0.00	0.76±0.02
yeast	1	10	1483	1.00±0.04	0.15±0.05	0.09±0.04	0.04±0.01
yeast	2	10	1483	0.53±0.02	0.21±0.06	0.10±0.04	0.05±0.01

neste trabalho. No que diz respeito à base de dados gerada por meio de uma distribuição normal multivariada, nota-se que os índices de qualidade tendem a diminuir drasticamente com o aumento do valor do desvio padrão (as linhas inferiores da tabela correspondem às simulações com maiores valores de desvio padrão). Entretanto, o algoritmo do scikit-learn parece ser o menos afetado por essa variação, enquanto o algoritmo guloso é o mais impactado.

6. Conclusão

Neste trabalho, investigamos o Problema dos k-Centros, avaliando dois algoritmos aproximativos 2-aproximados. Através de experimentos em bases de dados reais e sintéticas, observamos que o primeiro algoritmo, baseado em busca binária demonstrou uma maior precisão na obtenção de soluções com menores valores de r. No entanto, isso veio ao custo de um maior tempo de execução relativo. Por outro lado, o segundo algoritmo, que utiliza uma abordagem gulosa, destacou-se por sua simplicidade e velocidade de execução, mantendo uma precisão aceitável.

7. Referências

1. Kleinberg, J., Tardos, E. Algorithm Design. Pearson Education, 2006. Seções 11.1 e 11.2.
2. Mahajan, M., Nimbhorkar, P., Varadarajan, K. The planar k-means problem is NP-hard. *The Institute of Mathematical Sciences, Chennai*. [2009]
3. Wikipedia. K-means clustering. `https://en.wikipedia.org/wiki/K-means_clustering`.
4. Wikipedia. K-medoids. `https://en.wikipedia.org/wiki/K-medoids`.
5. Wikipedia. Minkowski distance. `https://en.wikipedia.org/wiki/Minkowski_distance`.

Tabela 2. Segundo algoritmo (guloso) - para bases de dados reais

Instância	p	Centro	Tamanho	Raio ± std	Silhouette ± std	Rand ± std	Tempo ± std
ObesityDataSet_raw_and_data_sinthetic	1	7	2111	36.26±1.51	0.29±0.04	0.25±0.03	0.07±0.00
ObesityDataSet_raw_and_data_sinthetic	2	7	2111	24.18±1.27	0.37±0.04	0.28±0.02	0.07±0.00
Raisin_Dataset	1	2	900	178742.15±15576.12	0.63±0.03	0.07±0.12	0.01±0.00
Raisin_Dataset	2	2	900	125636.00±13845.76	0.64±0.02	0.06±0.11	0.02±0.00
abalone	1	28	4176	0.60±0.02	0.21±0.04	0.06±0.00	0.30±0.03
abalone	2	28	4176	0.30±0.01	0.23±0.05	0.07±0.00	0.30±0.03
data_banknote_authentication	1	2	1371	28.04±2.63	0.44±0.04	0.07±0.03	0.03±0.00
data_banknote_authentication	2	2	1371	15.50±1.53	0.45±0.03	0.07±0.02	0.03±0.00
flare	1	8	1065	3.27±0.45	0.29±0.05	0.08±0.06	0.02±0.00
flare	2	8	1065	2.33±0.11	0.34±0.04	0.10±0.06	0.03±0.00
magic04	1	2	19019	1166.62±47.54	0.67±0.04	0.01±0.01	4.33±0.21
magic04	2	2	19019	575.18±22.92	0.69±0.04	0.00±0.01	4.20±0.10
poker-hand-training-true	1	10	25009	28.47±0.57	0.08±0.01	0.00±0.00	6.45±0.43
poker-hand-training-true	2	10	25009	13.61±0.39	0.11±0.02	0.00±0.00	6.96±0.42
transfusion	1	2	748	5450.97±648.72	0.85±0.01	0.03±0.01	0.01±0.00
transfusion	2	2	748	5327.79±601.74	0.86±0.00	0.03±0.00	0.01±0.00
winequality-white	1	7	6497	120.12±11.29	0.29±0.09	-0.00±0.01	0.72±0.02
winequality-white	2	7	6497	82.16±5.98	0.36±0.04	-0.00±0.00	0.71±0.01
yeast	1	10	1483	1.09±0.05	0.13±0.05	0.07±0.05	0.04±0.00
yeast	2	10	1483	0.58±0.02	0.18±0.05	0.06±0.05	0.04±0.00

6. UCI Machine Learning Repository. https://archive.ics.uci.edu/.
7. scikit-learn. K-Means Clustering. https://scikit-learn.org/stable/modules/clustering.html#k-means.
8. scikit-learn. Plot cluster comparison. https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py.
9. Wikipedia. Multivariate normal distribution. https://en.wikipedia.org/wiki/Multivariate_normal_distribution.
10. Wikipedia. Silhouette (clustering). https://en.wikipedia.org/wiki/Silhouette_(clustering).
11. Wikipedia. Rand index. https://en.wikipedia.org/wiki/Rand_index.

Bases de dados utilizadas:

UCI Machine Learning Repository. Banknote Authentication Data Set. https://archive.ics.uci.edu/dataset/267/banknote+authentication.

UCI Machine Learning Repository. MAGIC Gamma Telescope Data Set. https://archive.ics.uci.edu/dataset/159/magic+gamma+telescope.

UCI Machine Learning Repository. Raisin Data Set. https://archive.ics.uci.edu/dataset/850/raisin.

UCI Machine Learning Repository. Blood Transfusion Service Center Data Set. https://archive.ics.uci.edu/dataset/176/blood+transfusion+service+center.

Tabela 3. Algoritmo do scikit-learn - para bases de dados reais

Instância	p	Centro	Tamanho	Raio ± std	Silhouette ± std	Rand ± std	Tempo ± std
ObesityDataSet_raw_and_data_sinthetic	1	7	2111	59.99±0.00	0.33±0.00	0.32±0.00	0.17±0.01
ObesityDataSet_raw_and_data_sinthetic	2	7	2111	36.61±0.00	0.45±0.00	0.31±0.00	0.10±0.01
Raisin_Dataset	1	2	900	234887.22±0.00	0.61±0.00	0.34±0.00	0.06±0.00
Raisin_Dataset	2	2	900	135902.06±0.00	0.66±0.00	0.16±0.00	0.03±0.00
abalone	1	28	4176	2.22±0.00	0.26±0.00	0.06±0.00	0.91±0.04
abalone	2	28	4176	0.43±0.00	0.27±0.00	0.05±0.00	0.33±0.02
data_banknote_authentication	1	2	1371	38.39±0.00	0.00±0.00	0.00±0.00	0.03±0.00
data_banknote_authentication	2	2	1371	15.47±0.00	0.43±0.00	0.05±0.00	0.04±0.01
flare	1	8	1065	11.00±0.00	0.00±0.00	0.00±0.00	0.03±0.00
flare	2	8	1065	4.29±0.00	0.53±0.00	0.01±0.00	0.03±0.01
magic04	1	2	19019	1282.59±0.00	0.32±0.00	0.01±0.00	30.78±1.71
magic04	2	2	19019	666.99±0.00	0.43±0.00	0.06±0.00	4.06±0.10
poker-hand-training-true	1	10	25009	30.00±0.00	0.05±0.00	0.00±0.00	19.15±1.66
poker-hand-training-true	2	10	25009	10.85±0.00	0.15±0.00	0.00±0.00	6.98±0.49
transfusion	1	2	748	10594.00±0.00	0.55±0.00	0.06±0.00	0.03±0.00
transfusion	2	2	748	7978.51±0.00	0.72±0.00	0.07±0.00	0.02±0.00
winequality-white	1	7	6497	492.59±0.00	0.33±0.00	0.01±0.00	2.00±0.01
winequality-white	2	7	6497	316.15±0.00	0.37±0.00	0.00±0.00	0.70±0.01
yeast	1	10	1483	1.87±0.00	0.08±0.00	0.08±0.00	0.07±0.00
yeast	2	10	1483	0.76±0.00	0.19±0.00	0.15±0.00	0.06±0.01

UCI Machine Learning Repository. Estimation of Obesity Levels Based on Eating Habits and Physical Condition Data Set. https://archive.ics.uci.edu/dataset/544/estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition.

UCI Machine Learning Repository. Wine Quality Data Set. https://archive.ics.uci.edu/dataset/186/wine+quality

UCI Machine Learning Repository. Solar Flare Data Set. https://archive.ics.uci.edu/dataset/89/solar+flare.

UCI Machine Learning Repository. Poker Hand Data Set. https://archive.ics.uci.edu/dataset/158/poker+hand.

UCI Machine Learning Repository. Yeast Data Set. https://archive.ics.uci.edu/dataset/110/yeast.

UCI Machine Learning Repository. Abalone Data Set. https://archive.ics.uci.edu/dataset/1/abalone.

Tabela 4. Primeiro algoritmo (por busca binária de r) - para bases de dados sintéticas geradas por meio do scikit-learn

Instância	p	Centro	Tamanho	Raio ± std	Silhouette ± std	Rand ± std	Tempo ± std
center_3_saida_cluster_7.csv	1	3	800	2.11±0.21	0.54±0.07	0.60±0.09	0.02±0.01
center_3_saida_cluster_7.csv	2	3	800	1.72±0.18	0.56±0.04	0.60±0.07	0.01±0.00
center_3_saida_cluster_8.csv	1	3	800	1.95±0.18	0.42±0.06	0.42±0.07	0.01±0.00
center_3_saida_cluster_8.csv	2	3	800	1.49±0.17	0.42±0.06	0.46±0.08	0.01±0.00
center_3_saida_cluster_9.csv	1	3	800	2.06±0.19	0.53±0.10	0.80±0.12	0.02±0.01
center_3_saida_cluster_9.csv	2	3	800	1.59±0.16	0.52±0.08	0.81±0.09	0.03±0.00
center_4_saida_cluster_10.csv	1	4	800	2.06±0.22	0.33±0.03	0.47±0.09	0.01±0.00
center_4_saida_cluster_10.csv	2	4	800	1.51±0.14	0.34±0.03	0.48±0.10	0.01±0.00
center_4_saida_cluster_5.csv	1	4	800	1.94±0.19	0.33±0.02	0.43±0.08	0.01±0.00
center_4_saida_cluster_5.csv	2	4	800	1.51±0.15	0.33±0.02	0.42±0.07	0.01±0.00
center_4_saida_cluster_7.csv	1	4	800	1.85±0.20	0.54±0.05	0.34±0.07	0.02±0.00
center_4_saida_cluster_7.csv	2	4	800	1.43±0.15	0.54±0.09	0.36±0.07	0.01±0.00
center_5_saida_cluster_8.csv	1	5	800	1.50±0.13	0.39±0.02	0.46±0.07	0.01±0.00
center_5_saida_cluster_8.csv	2	5	800	1.11±0.12	0.38±0.03	0.48±0.08	0.01±0.00
center_5_saida_cluster_9.csv	1	5	800	1.69±0.15	0.39±0.08	0.66±0.07	0.01±0.00
center_5_saida_cluster_9.csv	2	5	800	1.30±0.11	0.41±0.08	0.65±0.09	0.01±0.00
center_6_saida_cluster_8.csv	1	6	800	1.35±0.09	0.38±0.03	0.46±0.07	0.02±0.00
center_6_saida_cluster_8.csv	2	6	800	1.03±0.09	0.39±0.03	0.49±0.07	0.01±0.00
center_7_saida_cluster_7.csv	1	7	800	1.34±0.06	0.49±0.05	0.74±0.06	0.02±0.01
center_7_saida_cluster_7.csv	2	7	800	1.05±0.05	0.48±0.05	0.76±0.07	0.03±0.01

Tabela 5. Segundo algoritmo (guloso) - Primeiro algoritmo (por busca binária de r) - para bases de dados sintéticas geradas por meio do scikit-learn

Instância	p	Centro	Tamanho	Raio ± std	Silhouette ± std	Rand ± std	Tempo ± std
center_3_saida_cluster_7.csv	1	3	800	2.70±0.32	0.49±0.12	0.53±0.11	0.02±0.01
center_3_saida_cluster_7.csv	2	3	800	2.19±0.21	0.51±0.10	0.53±0.10	0.01±0.00
center_3_saida_cluster_8.csv	1	3	800	2.41±0.25	0.36±0.09	0.30±0.14	0.01±0.00
center_3_saida_cluster_8.csv	2	3	800	1.79±0.16	0.42±0.06	0.39±0.10	0.01±0.00
center_3_saida_cluster_9.csv	1	3	800	2.67±0.22	0.44±0.09	0.58±0.15	0.02±0.01
center_3_saida_cluster_9.csv	2	3	800	2.17±0.19	0.53±0.09	0.72±0.18	0.03±0.00
center_4_saida_cluster_10.csv	1	4	800	2.55±0.27	0.34±0.05	0.53±0.16	0.01±0.00
center_4_saida_cluster_10.csv	2	4	800	1.91±0.14	0.35±0.04	0.55±0.15	0.01±0.00
center_4_saida_cluster_5.csv	1	4	800	2.19±0.13	0.33±0.03	0.40±0.09	0.01±0.00
center_4_saida_cluster_5.csv	2	4	800	1.55±0.06	0.36±0.02	0.45±0.07	0.01±0.00
center_4_saida_cluster_7.csv	1	4	800	2.15±0.22	0.49±0.08	0.30±0.04	0.01±0.00
center_4_saida_cluster_7.csv	2	4	800	1.78±0.15	0.53±0.05	0.32±0.06	0.01±0.00
center_5_saida_cluster_8.csv	1	5	800	1.74±0.12	0.32±0.04	0.44±0.09	0.01±0.00
center_5_saida_cluster_8.csv	2	5	800	1.36±0.10	0.37±0.04	0.52±0.09	0.01±0.00
center_5_saida_cluster_9.csv	1	5	800	2.01±0.14	0.35±0.09	0.59±0.11	0.01±0.00
center_5_saida_cluster_9.csv	2	5	800	1.60±0.15	0.36±0.07	0.60±0.10	0.01±0.00
center_6_saida_cluster_8.csv	1	6	800	1.58±0.08	0.33±0.05	0.51±0.07	0.01±0.00
center_6_saida_cluster_8.csv	2	6	800	1.24±0.10	0.36±0.05	0.55±0.09	0.01±0.00
center_7_saida_cluster_7.csv	1	7	800	1.53±0.08	0.45±0.09	0.68±0.11	0.01±0.00
center_7_saida_cluster_7.csv	2	7	800	1.21±0.07	0.47±0.05	0.70±0.08	0.02±0.01

Tabela 6. Algoritmo do scikit-learn - Primeiro algoritmo (por busca binária de r) - para bases de dados sintéticas geradas por meio do scikit-learn

Instância	p	Centro	Tamanho	Raio ± std	Silhouette ± std	Rand ± std	Tempo ± std
center_3_saida_cluster_7.csv	1	3	800	2.77±0.00	0.62±0.00	0.85±0.00	0.05±0.02
center_3_saida_cluster_7.csv	2	3	800	2.19±0.00	0.62±0.00	0.80±0.00	0.01±0.00
center_3_saida_cluster_8.csv	1	3	800	2.13±0.00	0.49±0.00	0.51±0.00	0.03±0.00
center_3_saida_cluster_8.csv	2	3	800	1.50±0.00	0.50±0.00	0.56±0.00	0.01±0.00
center_3_saida_cluster_9.csv	1	3	800	2.40±0.00	0.65±0.00	0.99±0.00	0.05±0.01
center_3_saida_cluster_9.csv	2	3	800	1.80±0.00	0.65±0.00	1.00±0.00	0.03±0.00
center_4_saida_cluster_10.csv	1	4	800	2.16±0.00	0.31±0.00	0.39±0.00	0.03±0.00
center_4_saida_cluster_10.csv	2	4	800	1.18±0.00	0.42±0.00	0.96±0.00	0.01±0.00
center_4_saida_cluster_5.csv	1	4	800	1.95±0.00	0.37±0.00	0.47±0.00	0.02±0.00
center_4_saida_cluster_5.csv	2	4	800	1.34±0.00	0.38±0.00	0.43±0.00	0.01±0.00
center_4_saida_cluster_7.csv	1	4	800	2.20±0.00	0.61±0.00	0.45±0.00	0.03±0.00
center_4_saida_cluster_7.csv	2	4	800	1.69±0.00	0.61±0.00	0.44±0.00	0.01±0.00
center_5_saida_cluster_8.csv	1	5	800	2.02±0.00	0.41±0.00	0.47±0.00	0.03±0.00
center_5_saida_cluster_8.csv	2	5	800	1.22±0.00	0.43±0.00	0.43±0.00	0.02±0.00
center_5_saida_cluster_9.csv	1	5	800	2.02±0.00	0.49±0.00	0.73±0.00	0.03±0.00
center_5_saida_cluster_9.csv	2	5	800	1.42±0.00	0.45±0.00	0.61±0.00	0.01±0.00
center_6_saida_cluster_8.csv	1	6	800	1.90±0.00	0.47±0.00	0.51±0.00	0.04±0.00
center_6_saida_cluster_8.csv	2	6	800	1.22±0.00	0.48±0.00	0.51±0.00	0.02±0.00
center_7_saida_cluster_7.csv	1	7	800	1.93±0.00	0.42±0.00	0.69±0.00	0.03±0.01
center_7_saida_cluster_7.csv	2	7	800	1.45±0.00	0.50±0.00	0.69±0.00	0.03±0.01

Tabela 7. Primeiro algoritmo (por busca binária de r) - para bases de dados sintéticas geradas com distribuição normal multivarias

Instância	p	Centro	Tamanho	Raio ± std	Silhouette ± std	Rand ± std	Tempo ± std
data_0	1	5	1855	0.63±0.05	0.63±0.07	0.91±0.05	0.06±0.00
data_0	2	5	1855	0.48±0.04	0.64±0.06	0.92±0.06	0.08±0.03
data_1	1	5	1975	0.81±0.05	0.45±0.11	0.65±0.08	0.08±0.02
data_1	2	5	1975	0.62±0.05	0.46±0.09	0.63±0.08	0.08±0.03
data_2	1	10	3880	0.82±0.05	0.34±0.04	0.54±0.06	0.24±0.03
data_2	2	10	3880	0.63±0.03	0.36±0.05	0.55±0.05	0.25±0.03
data_3	1	2	700	1.06±0.15	0.35±0.13	0.44±0.22	0.02±0.01
data_3	2	2	700	0.83±0.09	0.38±0.09	0.51±0.24	0.02±0.01
data_4	1	2	704	1.29±0.14	0.58±0.12	0.88±0.14	0.02±0.01
data_4	2	2	704	0.95±0.09	0.58±0.13	0.87±0.14	0.01±0.00
data_5	1	6	2304	1.15±0.08	0.26±0.05	0.38±0.05	0.11±0.02
data_5	2	6	2304	0.88±0.05	0.28±0.06	0.41±0.05	0.12±0.03
data_6	1	5	1905	1.25±0.09	0.34±0.06	0.50±0.07	0.07±0.02
data_6	2	5	1905	0.97±0.07	0.34±0.06	0.49±0.06	0.06±0.00
data_7	1	3	1158	1.46±0.13	0.32±0.08	0.41±0.14	0.02±0.00
data_7	2	3	1158	1.11±0.09	0.31±0.09	0.39±0.13	0.02±0.00
data_8	1	7	2667	1.30±0.08	0.22±0.03	0.30±0.03	0.14±0.02
data_8	2	7	2667	1.03±0.09	0.24±0.04	0.32±0.04	0.15±0.03
data_9	1	10	3640	1.09±0.06	0.22±0.02	0.23±0.02	0.21±0.02
data_9	2	10	3640	0.86±0.04	0.23±0.02	0.23±0.02	0.22±0.03

Tabela 8. Segundo algoritmo (guloso) - para bases de dados sintéticas geradas com distribuição normal multivarias

Instância	p	Centro	Tamanho	Raio ± std	Silhouette ± std	Rand ± std	Tempo ± std
data_0	1	5	1855	0.95±0.08	0.63±0.06	0.84±0.10	0.05±0.00
data_0	2	5	1855	0.68±0.05	0.65±0.04	0.87±0.10	0.07±0.03
data_1	1	5	1975	1.00±0.08	0.42±0.06	0.54±0.08	0.08±0.03
data_1	2	5	1975	0.82±0.06	0.46±0.04	0.56±0.07	0.07±0.03
data_2	1	10	3880	0.92±0.05	0.25±0.05	0.40±0.05	0.22±0.03
data_2	2	10	3880	0.73±0.03	0.24±0.04	0.37±0.04	0.22±0.03
data_3	1	2	700	1.35±0.16	0.33±0.08	0.26±0.19	0.02±0.01
data_3	2	2	700	1.05±0.15	0.32±0.09	0.23±0.20	0.01±0.00
data_4	1	2	704	1.55±0.15	0.64±0.07	0.83±0.15	0.02±0.01
data_4	2	2	704	1.23±0.13	0.63±0.09	0.80±0.18	0.01±0.00
data_5	1	6	2304	1.31±0.11	0.22±0.04	0.32±0.04	0.10±0.02
data_5	2	6	2304	1.04±0.06	0.23±0.03	0.32±0.03	0.11±0.03
data_6	1	5	1905	1.46±0.12	0.26±0.05	0.35±0.06	0.06±0.01
data_6	2	5	1905	1.17±0.07	0.26±0.04	0.35±0.04	0.05±0.01
data_7	1	3	1158	1.73±0.17	0.28±0.05	0.24±0.11	0.02±0.00
data_7	2	3	1158	1.40±0.12	0.27±0.07	0.25±0.12	0.02±0.00
data_8	1	7	2667	1.44±0.08	0.16±0.03	0.24±0.04	0.13±0.02
data_8	2	7	2667	1.17±0.06	0.17±0.04	0.24±0.05	0.13±0.03
data_9	1	10	3640	1.18±0.06	0.16±0.02	0.18±0.02	0.19±0.03
data_9	2	10	3640	0.96±0.05	0.15±0.03	0.18±0.03	0.19±0.02

Tabela 9. Algoritmo do scikit-learn - para bases de dados sintéticas geradas com distribuição normal multivarias

Instância	p	Centro	Tamanho	Raio ± std	Silhouette ± std	Rand ± std	Tempo ± std
data_0	1	5	1855	0.85±0.00	0.59±0.00	0.71±0.00	0.13±0.01
data_0	2	5	1855	0.41±0.00	0.71±0.00	0.99±0.00	0.07±0.03
data_1	1	5	1975	0.76±0.00	0.53±0.00	0.82±0.00	0.21±0.04
data_1	2	5	1975	0.56±0.00	0.52±0.00	0.68±0.00	0.08±0.03
data_2	1	10	3880	0.90±0.00	0.46±0.00	0.69±0.00	0.43±0.07
data_2	2	10	3880	0.67±0.00	0.43±0.00	0.70±0.00	0.23±0.04
data_3	1	2	700	1.01±0.00	0.51±0.00	0.79±0.00	0.03±0.01
data_3	2	2	700	0.82±0.00	0.51±0.00	0.81±0.00	0.02±0.01
data_4	1	2	704	1.24±0.00	0.71±0.00	1.00±0.00	0.03±0.01
data_4	2	2	704	0.87±0.00	0.71±0.00	0.99±0.00	0.01±0.00
data_5	1	6	2304	1.17±0.00	0.40±0.00	0.50±0.00	0.24±0.04
data_5	2	6	2304	0.90±0.00	0.35±0.00	0.48±0.00	0.12±0.03
data_6	1	5	1905	1.25±0.00	0.45±0.00	0.68±0.00	0.20±0.03
data_6	2	5	1905	0.86±0.00	0.45±0.00	0.68±0.00	0.06±0.00
data_7	1	3	1158	1.58±0.00	0.44±0.00	0.61±0.00	0.05±0.00
data_7	2	3	1158	1.13±0.00	0.44±0.00	0.62±0.00	0.02±0.00
data_8	1	7	2667	1.79±0.00	0.33±0.00	0.45±0.00	0.32±0.05
data_8	2	7	2667	1.24±0.00	0.36±0.00	0.48±0.00	0.14±0.04
data_9	1	10	3640	1.49±0.00	0.31±0.00	0.33±0.00	0.53±0.08
data_9	2	10	3640	0.99±0.00	0.33±0.00	0.29±0.00	0.20±0.03