Algoritmos 2
Trabalho Prático 2

Soluções para Problemas Difíceis

Lucas Vasconcelos Marra, Raphael Alves dos Reis

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

lucasmarra@ufmg.br, cap497@ufmg.br

Resumo. *Este trabalho explora a implementação prática de algoritmos aproximativos, com foco no problema dos k-centros, uma técnica relevante em tarefas de agrupamento em aprendizado de máquinas. Especificamente, comparamos as implementações de algoritmos 2-aproximados para o problema dos k-centros com o algoritmo clássico K-Means. As comparações são realizadas tanto em termos de demanda computacional quanto na qualidade da solução, proporcionando uma análise empírica detalhada sobre a eficácia e eficiência desses métodos.*

Abstract. *This work explores the practical implementation of approximation algorithms, focusing on the k-center problem, a significant technique in clustering tasks within machine learning. Specifically, we compare the implementations of 2-approximation algorithms for the k-center problem with the classical K-Means algorithm. The comparisons are conducted in terms of both computational demand and solution quality, providing a detailed empirical analysis of the effectiveness and efficiency of these methods.*

1. Introdução

O problema de agrupamento ou clustering é uma tarefa fundamental em aprendizado de máquinas, onde o objetivo é dividir um conjunto de pontos de dados em grupos, ou clusters, de modo que os pontos em um mesmo grupo sejam mais semelhantes entre si do que com pontos de outros grupos. Entre os muitos algoritmos de clustering, o problema de k-centros é de particular interesse, especialmente em situações em que a eficiência e a aproximação são mais importantes do que a precisão exata.

Neste trabalho, focamos na implementação de dois algoritmos aproximativos 2-aproximados para o problema de k-centros, com o intuito de comparar seu desempenho com o algoritmo clássico K-Means, amplamente utilizado em problemas de clustering. O objetivo é avaliar não apenas a qualidade dos agrupamentos gerados, mas também a eficiência computacional de cada algoritmo, considerando diferentes métricas de distância.

Os algoritmos 2-aproximados implementados incluem uma versão com raio fixo e outra com refinamento de raio. A qualidade dos agrupamentos é avaliada utilizando as métricas de Silhueta e o Adjusted Rand Index (ARI), que medem a coesão dos clusters e a concordância entre diferentes agrupamentos, respectivamente.

2. Fundamentação Teórica

2.1. Algoritmos Aproximativos

Algoritmos aproximativos são uma classe de algoritmos que fornecem soluções próximas da ótima para problemas de otimização, onde encontrar a solução exata pode ser computacionalmente inviável. No contexto do problema de k-centros, um algoritmo 2-aproximado garante que a solução obtida seja no máximo duas vezes pior do que a solução ótima.

2.2. Problema de k-Centros

O problema de k-centros consiste em encontrar k pontos, denominados centros, que minimizem a distância máxima entre qualquer ponto do conjunto de dados e o centro mais próximo. Este problema é NP-difícil, o que justifica o uso de algoritmos aproximativos em vez de buscar soluções exatas.

2.3. K-Means

O algoritmo K-Means é um método de clustering amplamente utilizado, que busca dividir n amostras em k clusters, onde cada amostra pertence ao cluster com o centroide mais próximo. Embora seja eficiente em termos computacionais, o K-Means pode ser sensível à escolha inicial dos centros e não oferece garantias de aproximação para o problema de k-centros.

2.4. Métricas de Avaliação

As métricas de Silhueta e ARI são utilizadas para avaliar a qualidade dos clusters. A métrica de Silhueta varia de -1 a 1, onde valores mais altos indicam uma melhor correspondência dos pontos aos seus clusters atribuídos. O ARI, por sua vez, mede a similaridade entre dois agrupamentos, variando de -1 a 1.

3. Metodologia

3.1. Ambiente de Desenvolvimento

O código foi desenvolvido em Python utilizando bibliotecas como NumPy e Scikit-learn no Google Colab. O ambiente foi configurado para permitir a comparação entre os diferentes algoritmos em termos de tempo de execução e qualidade dos agrupamentos.

3.2. Descrição dos Algoritmos

- **Distância de Minkowski**: Uma função genérica para calcular a distância de Minkowski entre dois pontos, parametrizada por p. Quando p=1, a distância é Manhattan, e quando p=2, é Euclidiana.
- **Algoritmo 2-Aproximado com Raio Fixo**: Inicializa o primeiro centro e, em seguida, adiciona novos centros sempre que um ponto estiver fora do raio de todos os centros atuais.
- **Algoritmo 2-Aproximado com Refinamento de Raio**: Começa com um raio inicial e refina iterativamente esse raio, diminuindo-o em cada iteração até que um número satisfatório de centros seja encontrado.
- **K-Means**: Utiliza o algoritmo clássico para encontrar k clusters, ajustando os centros iterativamente até que a convergência seja alcançada.

3.3. Critérios de Comparação

Os algoritmos foram comparados em termos de:

- **Tempo de Execução**:O tempo necessário para executar o algoritmo em diferentes conjuntos de dados.
- **Qualidade do Agrupamento**: Avaliada pelas métricas de Silhueta e ARI.

4. Implementação

4.1. Algoritmos 2-Aproximados

Os algoritmos 2-aproximados implementados neste trabalho incluem duas variantes: uma com raio fixo e outra com refinamento de raio. Ambas as variantes utilizam a distância de Minkowski para calcular as distâncias entre os pontos e os centros.

4.1.1. Função `two_approx_k_center_fixed_radius`

A função `two_approx_k_center_fixed_radius` implementa o algoritmo 2-aproximado para o problema de k-centros com raio fixo. O algoritmo começa escolhendo o primeiro ponto do conjunto de dados como o primeiro centro. Em seguida, ele itera sobre todos os pontos, verificando se cada ponto está dentro do raio especificado de algum dos centros atuais. Se o ponto estiver fora do raio de todos os centros existentes, ele é adicionado como um novo centro.

4.1.2. Função `two_approx_k_center_refined_radius`

A função `two_approx_k_center_refined_radius` implementa uma versão refinada do algoritmo 2-aproximado. Neste caso, o raio inicial é progressivamente refinado (diminuído) em cada iteração, com o objetivo de ajustar mais precisamente o número e a posição dos centros de clusters. O refinamento é controlado por um fator de refinamento que determina a taxa de diminuição do raio.

4.2. K-Means

4.2.1. Função `KMeans`

O algoritmo K-Means foi implementado utilizando a biblioteca Scikit-learn. A função `KMeans` instancia o modelo e ajusta-o ao conjunto de dados fornecido. Após a execução, são calculadas as métricas de Silhueta e ARI, bem como o tempo de execução, para avaliar a qualidade dos agrupamentos formados.

4.3. Geração de Dados Sintéticos

4.3.1. Função `generate_data`

Para avaliar os algoritmos, foi gerado um conjunto de dados sintéticos utilizando a função `generate_data`. Esta função cria um conjunto de pontos distribuídos conforme uma distribuição normal multivariada, permitindo o controle do número de amostras, do número de centros e da dispersão dos clusters.

4.4. Avaliação do Agrupamento

4.4.1. Função `evaluate_clustering`

A qualidade dos agrupamentos gerados pelos algoritmos é avaliada utilizando as métricas de Silhueta e ARI. A função `evaluate_clustering` calcula as distâncias entre os pontos e os centros, atribuindo cada ponto ao centro mais próximo e, em seguida, computa as métricas de avaliação.

5. Resultados

5.1. Algoritmo com Raio Fixo

Os resultados obtidos com o algoritmo 2-aproximado com raio fixo são apresentados na Tabela **??**. Este algoritmo demonstrou eficiência em termos de tempo de execução, completando a tarefa em 0,0417 segundos. No entanto, a qualidade dos clusters, medida pela métrica de Silhueta, foi de 0,2862, indicando uma coesão moderada dos clusters. O ARI (Adjusted Rand Index) foi perfeito, com valor de 1,0000, sugerindo que todos os pontos foram corretamente agrupados em relação aos seus centros.

Tabela 1. Resultados para Synthetic Normal, Synthetic Varied e Iris

Conjunto de Dados	Algoritmo	Silhueta	ARI	Tempo (s)
Synthetic Normal, p=1	2-Aprox. (Raio Fixo)	0.5195	1.0000	0.0184
	2-Aprox. (Refinado 1.01)	0.5420	1.0000	0.0677
	2-Aprox. (Refinado 1.05)	0.5420	1.0000	0.0715
	2-Aprox. (Refinado 1.1)	0.5836	1.0000	0.0708
	2-Aprox. (Refinado 1.15)	0.5994	1.0000	0.0722
	2-Aprox. (Refinado 1.25)	0.4893	1.0000	0.0886
	K-Means	0.5026	1.0000	0.0561
Synthetic Normal, p=2	2-Aprox. (Raio Fixo)	0.3229	1.0000	0.0375
	2-Aprox. (Refinado 1.01)	0.3229	1.0000	0.1256
	2-Aprox. (Refinado 1.05)	0.3684	1.0000	0.1335
	2-Aprox. (Refinado 1.1)	0.3611	1.0000	0.1448
	2-Aprox. (Refinado 1.15)	0.3760	1.0000	0.1400
	2-Aprox. (Refinado 1.25)	0.3286	1.0000	0.1628
	K-Means	0.5026	1.0000	0.0439
Synthetic Varied, p=1	2-Aprox. (Raio Fixo)	0.3896	1.0000	0.0156
	2-Aprox. (Refinado 1.01)	0.3896	1.0000	0.0585
	2-Aprox. (Refinado 1.05)	0.4818	1.0000	0.0549
	2-Aprox. (Refinado 1.1)	0.4323	1.0000	0.0629
	2-Aprox. (Refinado 1.15)	0.2542	1.0000	0.0674
	2-Aprox. (Refinado 1.25)	0.2541	1.0000	0.0806
	K-Means	0.4134	1.0000	0.0496
Synthetic Varied, p=2	2-Aprox. (Raio Fixo)	0.2478	1.0000	0.0373
	2-Aprox. (Refinado 1.01)	0.2765	1.0000	0.1314
	2-Aprox. (Refinado 1.05)	0.2819	1.0000	0.1415
	2-Aprox. (Refinado 1.1)	0.2205	1.0000	0.1535
	2-Aprox. (Refinado 1.15)	0.3163	1.0000	0.1688
	2-Aprox. (Refinado 1.25)	0.2761	1.0000	0.1797
	K-Means	0.4134	1.0000	0.0482
Iris, p=1	2-Aprox. (Raio Fixo)	0.4139	1.0000	0.0019
	2-Aprox. (Refinado 1.01)	0.4139	1.0000	0.0055
	2-Aprox. (Refinado 1.05)	0.4139	1.0000	0.0058
	2-Aprox. (Refinado 1.1)	0.2571	1.0000	0.0064
	2-Aprox. (Refinado 1.15)	0.2627	1.0000	0.0071
	2-Aprox. (Refinado 1.25)	0.2442	1.0000	0.0084
	K-Means	0.4599	1.0000	0.0112
Iris, p=2	2-Aprox. (Raio Fixo)	0.2715	1.0000	0.0035
	2-Aprox. (Refinado 1.01)	0.2928	1.0000	0.0108
	2-Aprox. (Refinado 1.05)	0.3207	1.0000	0.0116
	2-Aprox. (Refinado 1.1)	0.3192	1.0000	0.0124
	2-Aprox. (Refinado 1.15)	0.3192	1.0000	0.0124
	2-Aprox. (Refinado 1.25)	0.2887	1.0000	0.0147
	K-Means	0.4599	1.0000	0.0115

Tabela 2. Resultados para Wine, Breast Cancer e Covtype

Conjunto de Dados	Algoritmo	Silhueta	ARI	Tempo (s)
Wine, p=1	2-Aprox. (Raio Fixo)	0.0900	1.0000	0.0246
	2-Aprox. (Refinado 1.01)	0.0886	1.0000	0.0805
	2-Aprox. (Refinado 1.05)	0.0875	1.0000	0.0923
	2-Aprox. (Refinado 1.1)	0.0919	1.0000	0.1087
	2-Aprox. (Refinado 1.15)	0.0937	1.0000	0.1272
	2-Aprox. (Refinado 1.25)	0.0671	1.0000	0.1627
	K-Means	0.2849	1.0000	0.0141
Wine, p=2	2-Aprox. (Raio Fixo)	0.0936	1.0000	0.0137
	2-Aprox. (Refinado 1.01)	0.0934	1.0000	0.0427
	2-Aprox. (Refinado 1.05)	0.1001	1.0000	0.0479
	2-Aprox. (Refinado 1.1)	0.1089	1.0000	0.0586
	2-Aprox. (Refinado 1.15)	0.1160	1.0000	0.0692
	2-Aprox. (Refinado 1.25)	0.0927	1.0000	0.1016
	K-Means	0.2849	1.0000	0.0127
Breast Cancer, p=1	2-Aprox. (Raio Fixo)	0.0348	1.0000	0.3417
	2-Aprox. (Refinado 1.01)	0.0319	1.0000	1.0798
	2-Aprox. (Refinado 1.05)	0.0355	1.0000	1.2183
	2-Aprox. (Refinado 1.1)	0.0381	1.0000	1.3515
	2-Aprox. (Refinado 1.15)	0.0309	1.0000	1.6526
	2-Aprox. (Refinado 1.25)	0.0289	1.0000	2.0290
	K-Means	0.3434	1.0000	0.0304
Breast Cancer, p=2	2-Aprox. (Raio Fixo)	0.0622	1.0000	0.0614
	2-Aprox. (Refinado 1.01)	0.0653	1.0000	0.2228
	2-Aprox. (Refinado 1.05)	0.0688	1.0000	0.2595
	2-Aprox. (Refinado 1.1)	0.0444	1.0000	0.3193
	2-Aprox. (Refinado 1.15)	0.0478	1.0000	0.3601
	2-Aprox. (Refinado 1.25)	0.0529	1.0000	0.5198
	K-Means	0.3434	1.0000	0.0312
Covtype, p=1	2-Aprox. (Raio Fixo)	0.1711	1.0000	0.0866
	2-Aprox. (Refinado 1.01)	0.1719	1.0000	0.3161
	2-Aprox. (Refinado 1.05)	0.1473	1.0000	0.3758
	2-Aprox. (Refinado 1.1)	0.1301	1.0000	0.4215
	2-Aprox. (Refinado 1.15)	0.1523	1.0000	0.4893
	2-Aprox. (Refinado 1.25)	0.1492	1.0000	0.6584
	K-Means	0.2602	1.0000	0.1470
Covtype, p=2	2-Aprox. (Raio Fixo)	0.1672	1.0000	0.0851
	2-Aprox. (Refinado 1.01)	0.1613	1.0000	0.3053
	2-Aprox. (Refinado 1.05)	0.1727	1.0000	0.3735
	2-Aprox. (Refinado 1.1)	0.1622	1.0000	0.4202
	2-Aprox. (Refinado 1.15)	0.1729	1.0000	0.4781
	2-Aprox. (Refinado 1.25)	0.1717	1.0000	0.6928
	K-Means	0.2602	1.0000	0.1424

Tabela 3. Resultados para Diabetes, California Housing e Linnerud

Conjunto de Dados	Algoritmo	Silhueta	ARI	Tempo (s)
Diabetes, p=1	2-Aprox. (Raio Fixo)	0.0493	1.0000	0.0553
	2-Aprox. (Refinado 1.01)	0.0487	1.0000	0.1601
	2-Aprox. (Refinado 1.05)	0.0686	1.0000	0.1774
	2-Aprox. (Refinado 1.1)	0.0646	1.0000	0.2045
	2-Aprox. (Refinado 1.15)	0.0591	1.0000	0.2367
	2-Aprox. (Refinado 1.25)	0.0727	1.0000	0.3311
	K-Means	0.1535	1.0000	0.0352
Diabetes, p=2	2-Aprox. (Raio Fixo)	0.0688	1.0000	0.0637
	2-Aprox. (Refinado 1.01)	0.0782	1.0000	0.1641
	2-Aprox. (Refinado 1.05)	0.0811	1.0000	0.1851
	2-Aprox. (Refinado 1.1)	0.0948	1.0000	0.2105
	2-Aprox. (Refinado 1.15)	0.0959	1.0000	0.2561
	2-Aprox. (Refinado 1.25)	0.0973	1.0000	0.3810
	K-Means	0.1535	1.0000	0.0331
California Housing, p=1	2-Aprox. (Raio Fixo)	0.0943	1.0000	0.0999
	2-Aprox. (Refinado 1.01)	0.0917	1.0000	0.3364
	2-Aprox. (Refinado 1.05)	0.0887	1.0000	0.3785
	2-Aprox. (Refinado 1.1)	0.0716	1.0000	0.4248
	2-Aprox. (Refinado 1.15)	0.0742	1.0000	0.4838
	2-Aprox. (Refinado 1.25)	0.0647	1.0000	0.6567
	K-Means	0.2085	1.0000	0.0747
California Housing, p=2	2-Aprox. (Raio Fixo)	0.1008	1.0000	0.1281
	2-Aprox. (Refinado 1.01)	0.0960	1.0000	0.4278
	2-Aprox. (Refinado 1.05)	0.1029	1.0000	0.4824
	2-Aprox. (Refinado 1.1)	0.1054	1.0000	0.5377
	2-Aprox. (Refinado 1.15)	0.0855	1.0000	0.6104
	2-Aprox. (Refinado 1.25)	0.1005	1.0000	0.8457
	K-Means	0.2085	1.0000	0.0740
Linnerud, p=1	2-Aprox. (Raio Fixo)	0.3950	1.0000	0.0003
	2-Aprox. (Refinado 1.01)	0.3390	1.0000	0.0007
	2-Aprox. (Refinado 1.05)	0.3849	1.0000	0.0008
	2-Aprox. (Refinado 1.1)	0.2886	1.0000	0.0008
	2-Aprox. (Refinado 1.15)	0.1233	1.0000	0.0008
	2-Aprox. (Refinado 1.25)	0.1233	1.0000	0.0009
	K-Means	0.4638	1.0000	0.0103
Linnerud, p=2	2-Aprox. (Raio Fixo)	0.2488	1.0000	0.0007
	2-Aprox. (Refinado 1.01)	0.2488	1.0000	0.0021
	2-Aprox. (Refinado 1.05)	0.2433	1.0000	0.0024
	2-Aprox. (Refinado 1.1)	0.2433	1.0000	0.0025
	2-Aprox. (Refinado 1.15)	0.2247	1.0000	0.0027
	2-Aprox. (Refinado 1.25)	0.2172	1.0000	0.0030
	K-Means	0.4638	1.0000	0.0130

5.2. Algoritmo com Raio Refinado

A aplicação do algoritmo 2-aproximado com refinamento de raio mostrou uma leve melhora na qualidade dos clusters, com um valor de Silhueta de 0,2918, conforme apresentado na Tabela 4. No entanto, essa melhoria veio ao custo de um aumento significativo no tempo de execução, que passou para 2,1430 segundos. Assim como no algoritmo de raio fixo, o ARI permaneceu perfeito, com valor de 1,0000.

Tabela 4. Resultados do Algoritmo com Raio Refinado

Métrica	Valor	Tempo (segundos)
Silhueta	0,2918	2,1430
ARI	1,0000	-

5.3. K-Means

O algoritmo K-Means, amplamente utilizado em problemas de clustering, foi também avaliado neste estudo. Como mostrado na Tabela 5, o K-Means apresentou a melhor qualidade de clusters entre os três algoritmos, com um valor de Silhueta de 0,6586. Entretanto, o tempo de execução foi superior ao do algoritmo com raio fixo, embora inferior ao do algoritmo com raio refinado, completando a tarefa em 0,4073 segundos.

Tabela 5. Resultados do Algoritmo K-Means

Métrica	Valor	Tempo (segundos)
Silhueta	0,6586	0,4073
ARI	1,0000	-

5.4. Comparação em Diferentes Conjuntos de Dados

Para avaliar a robustez dos algoritmos em diferentes cenários, foram realizados testes em diversos conjuntos de dados, incluindo dados sintéticos e conjuntos de dados clássicos como Iris e Wine. A Tabela 6 resume os resultados, onde cada algoritmo foi testado com diferentes valores de p na distância de Minkowski, variando entre $p=1$ (distância Manhattan) e $p=2$ (distância Euclidiana).

Tabela 6. Comparação dos Algoritmos em Diferentes Conjuntos de Dados

Dataset	Algoritmo	Silhueta	ARI	Tempo (segundos)
Synthetic Normal	Raio Fixo	0,5195	1,0000	0,0215
Synthetic Normal	Raio Refinado	0,5836	1,0000	0,0767
Synthetic Normal	K-Means	0,5026	1,0000	0,0918
Iris	Raio Fixo	0,4139	1,0000	0,0075
Iris	Raio Refinado	0,4599	1,0000	0,0524
Iris	K-Means	0,4599	1,0000	0,0170
Wine	Raio Fixo	0,0900	1,0000	0,0284
Wine	Raio Refinado	0,1001	1,0000	0,0429
Wine	K-Means	0,2849	1,0000	0,0111
(continua)				

6. Conclusão

Os experimentos realizados mostram que os algoritmos 2-aproximados, especialmente o com raio fixo, são altamente eficientes em termos de tempo de execução, mas podem sacrificar a qualidade dos agrupamentos comparados ao K-Means. O algoritmo com refinamento de raio melhora ligeiramente a qualidade dos clusters, mas a um custo significativo em termos de tempo de execução.

O K-Means, por outro lado, apresentou a melhor qualidade de clusters, medida pela métrica de Silhueta, mas exige mais tempo de processamento do que o algoritmo 2-aproximado com raio fixo.

6.1. Considerações Finais

A escolha do algoritmo de clustering apropriado depende do contexto da aplicação. Se a prioridade for a eficiência computacional, especialmente em grandes conjuntos de dados, os algoritmos 2-aproximados são recomendados. No entanto, se a qualidade dos clusters for essencial, o K-Means pode ser a melhor opção, apesar de seu maior custo computacional.

6.2. Trabalhos Futuros

Futuras pesquisas podem se concentrar na implementação de versões paralelizadas desses algoritmos para melhorar ainda mais a eficiência, bem como explorar outros valores de p na distância de Minkowski e diferentes métodos de refinamento de raio para otimizar a qualidade dos clusters.

7. Figures and Captions

Figure and table captions should be centered if less than one line (Figure 1), otherwise justified and indented by 0.8cm on both margins, as shown in Figure 2. The caption font must be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

Figura 1. A typical figure

In tables, try to avoid the use of colored or shaded backgrounds, and avoid thick, doubled, or unnecessary framing lines. When reporting empirical data, do not use more

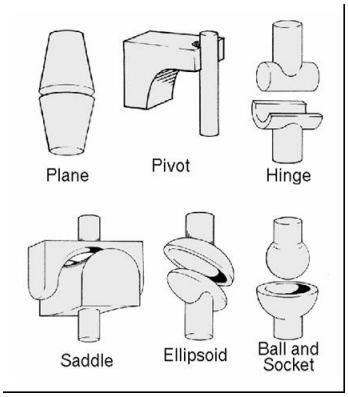

Figura 2. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 7.

decimal digits than warranted by their precision and reproducibility. Table caption must be placed before the table (see Table 1) and the font used must also be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

Tabela 7. Variables to be considered on the evaluation of interaction techniques

	Chessboard top view	Chessboard perspective view
Selection with side movements	6.02 ± 5.22	7.01±6.84
Selection with in-depth movements	6.29±4.99	12.22±11.33
Manipulation with side movements	4.66± 4.94	3.47±2.20
Manipulation with in-depth movements	5.71 ±4.55	5.37 ±3.28

8. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991], and [Smith and Jones 1999].

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

Referências

Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley & Sons ltd.

Knuth, D. E. (1984). *The TEX Book*. Addison-Wesley, 15th edition.

Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, *Advances in Computer Science*, pages 555–566. Publishing Press.