Soluções para Problemas Difíceis: K-centros e algoritmos aproximativos

Laila Melo Vaz Lopes, Thiago Martins Lima Assis

[1]DCC – Universidade Federal de Minas Gerais

Abstract. *This report corresponds to a practical assignment for the Algorithms 2 course, focusing on the experimental analysis of approximation algorithms for the k-centers problem. Here, we will present a description of the problem and the implementation of the algorithms, the results of the experiments, and conclusions on the subject.*

Resumo. *Esse relatório corresponde a um trabalho prático da disciplina Algoritmos 2, sobre análise experimental de algoritmos aproximativos para o Problema dos K-Centros. Aqui serão apresentados descrição do problema e da implementação dos algoritmos, resultados dos experimentos e conclusões a respeito do tema.*

1. Contextualização e Motivação

O **Problema dos K-Centros** consiste em, dado um conjunto de pontos, encontrar K centros de forma que todos os pontos fiquem a uma distância máxima de um dos centros. Esse problema possui diversas aplicações na área de ciência de dados, principalmente nas situações em que é preciso encontrar segmentações de dados da melhor forma possível.

Dessa forma, devido ao grande volume de dados existente atualmente, se faz necessário descobrir algoritmos que resolvam esse problema de forma eficiente, entretanto, sabe-se que, devido a sua NP-dificuldade, não há formas conhecidas de encontrar uma resposta ótima em tempo polinomial. Por isso, nesse trabalho, exploraremos dois algoritmos aproximativos rápidos, que, apesar de não obterem a resposta ótima em todos os casos, garantem uma solução no máximo duas vezes pior do que a desejada, o que é suficiente para a maioria das aplicações.

Por fim, vamos verificar a qualidade das soluções de cada algoritmos comparando com o algoritmo de *K-Means* do *Scikit Learn*.

2. Métodos e Métricas

2.1. Algoritmos utilizados

Vamos implementar dois algoritmos aproximativos para o problema. O primeiro consiste em fazer uma busca binária da solução no intervalo $[0, rmax]$, em que $rmax$ representa a distância máxima entre dois pontos da entrada. Nessa implementação, a cada iteração refinamos o espaço da solução, até que achamos o raio final que é no máximo duas vezes maior que raio da solução ótima. O custo assintótico desse algoritmo é $O(n^2 logn)$.

O segundo algoritmo consiste em, a cada iteração, escolher gulosamente pontos que maximizam a distância com os centros já escolhidos para se tornarem novos centros. No final, assintoticamente temos um custo de $O(n^2)$.

2.2. Métrica de Minkowski

Nesse trabalho utilizamos duas funções de distância, que são as *distâncias de Minkowski* para $p = 1$ e $p = 2$. Essa distância é definida por

$$\text{dist}_p(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Quando $p = 1$ essa métrica recebe o nome de *distância de Manhattan*. Já quando $p = 2$ chamamos ela de *distância Euclidiana*.

2.3. Detalhes de Implementação

Ambos os algoritmos aproximativos fazem escolhas arbitrárias durante a sua execução. Desse modo, duas execuções de um dos algoritmos podem revelar resultados diferentes, mesmo sobre uma mesma entrada. Para que isso ocorresse no código projetado, utilizamos os métodos `random.shuffle` e `random.randint` do python para simular essas escolhas arbitrárias.

Outro ponto que vale destacar é que os métodos foram encapsulados em arquivos `.py`, para melhorar a modularização do código e para que os notebooks ficassem mais legíveis.

2.4. Comparativos

Para avaliar a qualidade das soluções, tanto os dois algoritmos aproximativos quanto o algoritmo do *Scikit Learn* foram testados em 30 bases de dados diferentes, sendo 10 bases com dados reais e 20 bases com dados sintéticos, todas com mais de 700 instâncias cada. Em todos os testes foram armazenados o raio alcançado e o tempo médio gasto para cada instância. Além disso, foram gerados medidas clássicas para avaliar o agrupamento dos dados: a silhueta e o índice de Rand ajustado.

A silhueta é a medida que avalia a coesão (o quão próximo os dados de um mesmo centro estão) e a separação (o quão distante os centros estão entre si). Os valores podem variar no intervalo [-1, 1], sendo que o ideal é tê-los o mais próximo possível de 1.

Já o índice de *Rand* ajustado é uma medida de similaridade da solução com a aleatoriedade. Ele mede o quão melhor a solução computada foi do que agrupar os dados aleatoriamente. Os valores variam no intervalo de $[0, 1]$, sendo que o valor 0 representa que o agrupamento é similar a um agrupamento aleatório.

2.5. Base de Dados

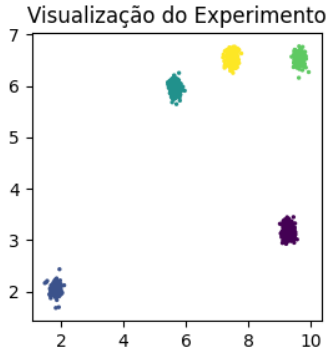
Os dados reais foram obtidos no site [2], e são ao seguintes:

- **Abalone** [3]
- **Raisin** [7]
- **Banknote Authentication** [4]
- **Rice Cammeo-and Osmancik** [8]
- **Eletrical-Stability** [5]
- **Wine Quality** [9]: utilizamos dois bancos de dados presente nesse repositório; um que só contém vinhos brancos e outro que só contém vinhos tintos.
- **Online shopping** [6]
- **Yeast** [10]
- **Optical Recognition of Handwritten Digits** [1]

3. Analisando os Experimentos Sintéticos

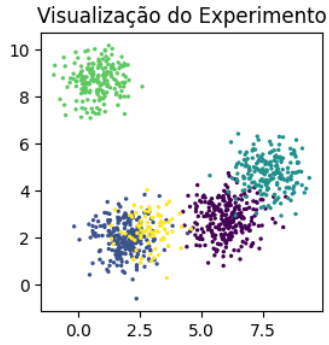
Nessa seção vamos ver como os algoritmos se comportam para um caso específico de dados sintéticos. Esses dados foram gerados da seguinte maneira: 4 centros aleatórios foram fixados e, para cada um deles, foram gerados pontos em seu redor segundo uma distribuição normal multivariada. Vamos analisar a diferença entre algoritmos quando a variância dessa distribuição cresce, ou seja, centros passam a ser mais sobrepostos.

Para cada caso de sobreposição, iremos mostrar uma imagem representando a geração dos pontos e uma tabela contendo valores úteis. Esses valores são: tempo de execução do algoritmo, raio da solução, índice de silhueta e índice de *rand* ajustado. Além disso, mostramos a média (denotada por μ) e o desvio padrão (representado por σ) desses dados ao longo de 30 repetições dos testes. A métrica utilizada nesses dados foi a com $p = 1$.



Resultados com centros pouco sobrepostos					
		Tempo	Raio	*Rand*	Silhueta
Incremental	μ	0.011	8.939	0.721	0.500
	σ	0.000	0.000	0.000	0.000
Refinamento	μ	0.224	8.952	0.463	0.418
	σ	0.004	0.019	0.167	0.099
Kmeans	μ	0.024	9.020	0.926	1
	σ	0.326	0.109	0.000	0.000

No primeiro experimento podemos notar algumas coisas interessantes: primeiramente que o raio encontrado pelos algoritmos 2-aproximativos foi menor que o do *K-means*. Entretanto, avaliando a silhueta e o *Rand* do *K-means*, percebemos uma maior coesão e estabilidade, o que indica que a solução é mais consistente, apesar do raio maior.



Resultados com centros sobrepostos					
		Tempo	Raio	*Rand*	Silhueta
Incremental	μ	0.010	13.796	0.258	0.420
	σ	0.000	0.269	0.037	0.115
Refinamento	μ	0.228	13.968	0.235	0.470
	σ	0.004	0.244	0.144	0.108
Kmeans	μ	0.123	13.900	0.508	0.781
	σ	0.014	0.198	0.000	0.005

No segundo experimento pode-se destacar a rapidez do algoritmo incremental se comparado com os seus concorrentes, o que é esperado pela sua complexidade assintótica. Mais uma vez , a coesão do algoritmo do *K-means* se destaca.

Por fim, no experimento dos centros sobrepostos permanece as observações anteriores, com a diferença de que agora o algoritmo do *K-means*, além de preservar sua coesão se comparado com os demais, também encontrou o menor raio.

Resultados com centros muito sobrepostos					
		Tempo	Raio	*Rand*	Silhueta
Incremental	μ	0.011	23.282	0.023	0.051
	σ	0.003	0.954	0.072	0.007
Refinamento	μ	0.223	20.970	0.205	0.110
	σ	0.024	2.978	0.048	0.029
Kmeans	μ	0.153	20.915	0.330	0.168
	σ	0.011	2.045	0.000	0.001

As conclusões que podemos tirar dessa análise de dados é que, independente do algoritmo, o *Rand* e a Silhueta são melhores em dados menos sobrepostos e, justamente nesses dados, o *K-means* tem a melhor perfomance de coesão, o que é esperado. Já, em dados mais agrupados, temos uma queda no Rand e na Silhueta de todos os algoritmos, e o raio encontrado pelos aproximativos se torna pior que o *K-means*

4. Outros Experimentos Sintéticos

Os conjuntos de dados que foram gerados a partir da abordagem presente em [11] podem ser encontrados no apêndice 6.

5. Analisando Experimentos Reais

Para estudar o $k - centros$ em dados reais, escolhemos ilustrar com o DataSet de *Bank Note Authentication*, que visa analisar fotos de notas autênticas e falsas, e conta com as seguintes variáveis:

1. Variância da imagem
2. Assimetria da imagem
3. Curtose da imagem
4. Entropia da imagem

Ao computar os algoritmos de cluster nesse dataset, encontramos uma silhueta moderada, com dados sobrepostos, o que indicaria uma sobreposição de características de notas falsas e verdadeiras.

Mas, ao mesmo tempo, a pontuação de Rand foi baixa, o que nos diz que nosso algoritmo não encontrou clusters muito melhores do que a aleatoriedade nos daria. Com isso, concluímos que os algoritmos não conseguem nos dar uma informação realista para esse DataSet, e que precisaríamos fazer um refinamento dos dados maior para conseguirmos aplicá-los de maneira satisfatória.

6. Conclusão

Neste trabalho, conseguimos fazer várias afirmações importantes sobre os algoritmos aproximativos, especialmente sobre as situações em que seu uso é vantajoso. Para acessar dados adicionais além dos apresentados até aqui, há um apêndice disponível ao final deste PDF, com informações complementares que reforçam as conclusões discutidas ao longo do artigo.

Por fim, destacamos a importância da existência dessa classe de algoritmos. Devido à grande quantidade de dados, percebemos que o tempo necessário para testar todas as instâncias é significativo. Esse tempo seria exponencialmente maior se fossemos utilizar o algoritmo ótimo. Portanto, os algoritmos aproximativos se mostram essenciais, fornecendo soluções em um tempo viável, mesmo que não sejam sempre ótimas, são suficientemente boas para muitas aplicações práticas.

Referências

[1] Dheeru Dua and Casey Graff. Optical Recognition of Handwritten Digits. https://archive.ics.uci.edu/dataset/80/optical+recognition+of+handwritten+digits, 2019. Accessed: 2024-08-15.

[2] Dheeru Dua and Casey Graff. UCI machine learning repository, 2019.

[3] UCI Machine Learning Repository. Abalone. https://archive.ics.uci.edu/dataset/1/abalone, 2024.

[4] UCI Machine Learning Repository. Banknote authentication. https://archive.ics.uci.edu/dataset/267/banknote+authentication, 2024.

[5] UCI Machine Learning Repository. Electrical grid stability simulated data. https://archive.ics.uci.edu/dataset/471/electrical+grid+stability+simulated+data, 2024.

[6] UCI Machine Learning Repository. Online shoppers purchasing intention dataset. https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset, 2024.

[7] UCI Machine Learning Repository. Raisin. https://archive.ics.uci.edu/dataset/850/raisin, 2024.

[8] UCI Machine Learning Repository. Rice cammeo and osmancik. https://archive.ics.uci.edu/dataset/545/rice+cammeo+and+osmancik, 2024.

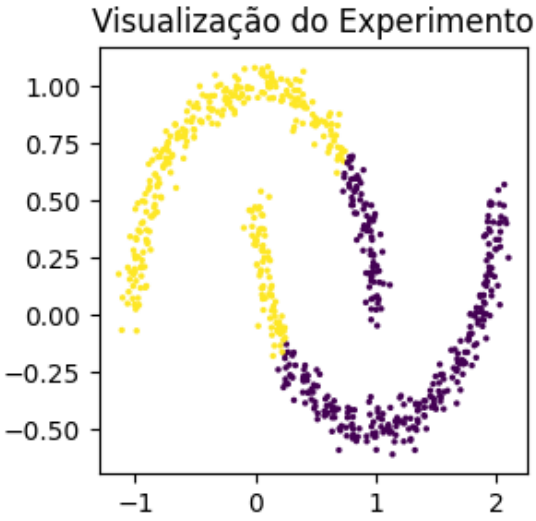
[9] UCI Machine Learning Repository. Wine quality. https://archive.ics.uci.edu/dataset/186/wine+quality, 2024.

[10] UCI Machine Learning Repository. Yeast. https://archive.ics.uci.edu/dataset/110/yeast, 2024.

[11] The scikit-learn developers. Cluster comparison, 2024. Accessed: 2024-08-15.

Apêndice

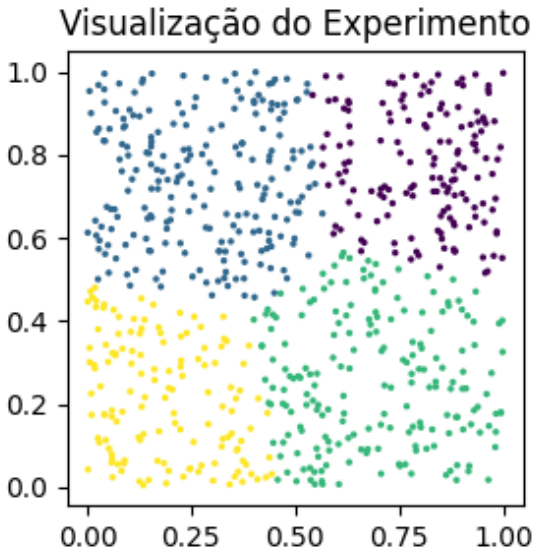

```
Teste incremental feito:
   Média     Desvio padrão
tempo:    0.0019665082295735677   0.0003504564412833385
raio:    1.8342116037076694   0.18752049214763705
silhueta:   0.3350326301608055   0.028878827539046786
rand:    0.23170298639345013   0.211299577609347

Teste kmeans feito:
   Média     Desvio padrão
tempo:    0.12625091075897216   0.008932034714370509
raio:    1.9088010205467003   0.1829259367954351
silhueta:   0.3549339247177435   0.000994401016074914
rand:    0.4236092759802887   0.0840353267329275

Teste refinamento com largura 1% feito:
   Média     Desvio padrão
tempo:    0.16998841762542724   0.008307585325133284
raio:    1.754428413571551   0.25132010399651405
silhueta:   0.33412291705811714   0.0292860903639387
rand:    0.2444558810549253   0.18353462316916158

Teste refinamento com largura 5% feito:
   Média     Desvio padrão
tempo:    0.16253225803375243   0.002346222283772193
raio:    1.753364945515921   0.18524289658499465
silhueta:   0.3413852644820544   0.022004665558692214
rand:    0.23180879060416018   0.2332137452410549

Teste refinamento com largura 10% feito:
   Média     Desvio padrão
tempo:    0.1638585885365804   0.0030594365835647976
raio:    1.777030014019982   0.24054972102693223
silhueta:   0.3372161975589846   0.024831391395233125
rand:    0.2770979780831874   0.2517001934607197
```




```
Teste refinamento com largura 15% feito:
   Média     Desvio padrão
tempo:    0.16265936692555746   0.0022790687695444594
raio:    1.831546175410968   0.20219876781913668
silhueta:   0.33054524564808857   0.048798799634467885
rand:    0.34066381658795236   0.29635321737321974

Teste refinamento com largura 20% feito:
   Média     Desvio padrão
tempo:    0.16134475072224935   0.0013638994929511391
raio:    1.678197395429365   0.21723329480114315
silhueta:   0.3338734301835188   0.03561635292669612
rand:    0.233638113600884   0.2713730603349831

Teste incremental feito:
   Média     Desvio padrão
tempo:    0.0020291805267333984   0.0003676444227524686
raio:    2.408364097992396   0.7038066564369564
silhueta:   0.31694780293970276   0.14741452116816964
rand:    0.3117425125638162   0.3119725325657234

Teste kmeans feito:
   Média     Desvio padrão
tempo:    0.12751339276631674   0.003947087118506126
raio:    2.6290509931304444   0.6237393298532001
silhueta:   0.4860830673835969   1.6653345369377348e-16
rand:    0.9160023590177375   3.3306690738754696e-16

Teste refinamento com largura 1% feito:
   Média     Desvio padrão
tempo:    0.16834239959716796   0.002511131913887711
raio:    2.041360688210332   0.3480973853382455
silhueta:   0.3680873485948962   0.10448877827758965
rand:    0.3058742107497265   0.29645841610487483

Teste refinamento com largura 5% feito:
   Média     Desvio padrão
```


```
tempo:    0.16863365968068442   0.004024302395027293
raio:    2.110539580613839   0.31207557575348277
silhueta:   0.33835126265175414   0.12208923353105344
rand:    0.2681011542112063   0.28444118685618947

Teste refinamento com largura 10% feito:
   Média     Desvio padrão
tempo:    0.16773746808369955   0.0028085009403012915
raio:    2.027580362204183   0.39362599295775846
silhueta:   0.36587929761587884   0.1192693260864988
rand:    0.37552221902545213   0.31300761483707795

Teste refinamento com largura 15% feito:
   Média     Desvio padrão
tempo:    0.1692842960357666   0.004447725713686877
raio:    2.081347183510577   0.44263814623278896
silhueta:   0.3518395278516102   0.12309124360090064
rand:    0.3225493467969917   0.2970345139429907

Teste refinamento com largura 20% feito:
   Média     Desvio padrão
tempo:    0.17150445779164633   0.009723361096253939
raio:    2.096239572461058   0.4930591370752039
silhueta:   0.3352878911936381   0.1527480622169699
rand:    0.34649096557104525   0.3255991668672913

Teste incremental feito:
   Média     Desvio padrão
tempo:    0.004494714736938477   0.000873950733875108
raio:    10.467426860772633   3.8578721750232
silhueta:   0.3517470479529894   0.0007176613609708294
rand:    0.5619019852013485   0.00016722687402519147

Teste kmeans feito:
   Média     Desvio padrão
tempo:    0.12158498764038086   0.014330504490855668
raio:    9.820086045222105   4.747793657367287
```

```
silhueta:   0.7543558996480884   3.3306690738754696e-16
rand:    0.9957081634660452   2.220446049250313e-16

Teste refinamento com largura 1% feito:
   Média     Desvio padrão
tempo:    0.17700662612915039   0.004655740697145294
raio:    10.629773849296111   3.996238685243612
silhueta:   0.4537309081038878   0.20232123137748598
rand:    0.6317090696293416   0.2390711756129731

Teste refinamento com largura 5% feito:
   Média     Desvio padrão
tempo:    0.18106516202290854   0.0110243308456234
raio:    10.338944827809227   4.545029850550584
silhueta:   0.49148430387173214   0.22302221878040877
rand:    0.6597777130094202   0.2744510496795102

Teste refinamento com largura 10% feito:
   Média     Desvio padrão
tempo:    0.18784870306650797   0.009363875594611762
raio:    10.081315132819581   4.2037614221054955
silhueta:   0.44929416527808175   0.24892626801434484
rand:    0.6312032936979933   0.2840877473502745

Teste refinamento com largura 15% feito:
   Média     Desvio padrão
tempo:    0.19186205863952638   0.00911629887049156
raio:    11.261022630498561   4.3466985485167395
silhueta:   0.4440872063407661   0.23247606157853606
rand:    0.6029414977463268   0.28182968573885747

Teste refinamento com largura 20% feito:
   Média     Desvio padrão
tempo:    0.18509276707967123   0.009032800681132797
raio:    11.058721830377701   4.173331319723632
silhueta:   0.44850370581862664   0.21622343124811347
rand:    0.6099018144298692   0.22233175555312815

Teste incremental feito:
   Média     Desvio padrão
tempo:    0.008658099174499511   0.00255677287704538
raio:    0.9835987208882125   0.06117186920929305
silhueta:   0.19822597548114843   0.07309398078032357
rand:    0.2851130235377425   0.12445492703119833

Teste kmeans feito:
   Média     Desvio padrão
tempo:    0.13751566410064697   0.008652428921107593
raio:    0.9547862957515549   0.07023746140135187
silhueta:   0.411769354788424   0.00028523555256928677
rand:    0.8463869478395393   0.0035160657787225596

Teste refinamento com largura 1% feito:
   Média     Desvio padrão
tempo:    0.17835274537404378   0.005125118437353624
raio:    0.890465268820511   0.10571225012275415
```




```
silhueta:   0.23483578287159051   0.09230728643043307
rand:    0.4478445152495351   0.14850137335519653

Teste refinamento com largura 5% feito:
   Média     Desvio padrão
tempo:    0.1774393876393636   0.005312480795242559
raio:    0.932114155156189   0.13861830934442038
silhueta:   0.23214251335963443   0.08816978597626106
rand:    0.3798257838562394   0.10298378310389059

Teste refinamento com largura 10% feito:
   Média     Desvio padrão
tempo:    0.17397146224975585   0.005124286419862058
raio:    0.876170701341423   0.12344198208863566
silhueta:   0.26422815214664497   0.06686726088419992
rand:    0.40732663899216026   0.08854763872933893

Teste refinamento com largura 15% feito:
   Média     Desvio padrão
tempo:    0.17062257130940756   0.004496761346677059
raio:    0.888764953568380   0.1184101439263502
silhueta:   0.23465999377423136   0.0778010086863031
rand:    0.408044872958395   0.11072656354405376

Teste refinamento com largura 20% feito:
   Média     Desvio padrão
tempo:    0.1711702028910319   0.004362721831228185
raio:    0.894935103078085   0.11049604776946081
silhueta:   0.2631300096677958   0.08309366402037896
rand:    0.39935014581950035   0.11814493833965052

Teste incremental feito:
   Média     Desvio padrão
tempo:    0.002871847152709961   0.000333815281583638
raio:    33.19391304444708   0.40413065277013077
silhueta:   0.5942183373853187   0.0
```


```
rand:    0.6570031360004798   0.0

Teste kmeans feito:
   Média     Desvio padrão
tempo:    0.1267333745956421   0.0036978316531072605
raio:    29.684363496728096   3.386885564850758
silhueta:   0.663188679766331   0.00010551898013925005
rand:    0.9838700117621111   0.002719802805178745

Teste refinamento com largura 1% feito:
   Média     Desvio padrão
tempo:    0.12939629554748536   0.0034333524839085944
raio:    24.648031826957119   6.953128195675585
silhueta:   0.5866145223105259   0.11100564030817138
rand:    0.5954665388626552   0.09988766586046435

Teste refinamento com largura 5% feito:
   Média     Desvio padrão
tempo:    0.1261241833368937   0.004534395755195712
raio:    27.993348380138467   6.312093246218156
silhueta:   0.4398262208753882   0.21081355390513137
rand:    0.4999927152785749   0.14029899336750182

Teste refinamento com largura 10% feito:
   Média     Desvio padrão
tempo:    0.1264898220698039   0.0032138360072044465
raio:    25.456027018127077   7.255948690836811
silhueta:   0.5096150376848521   0.19814832037552663
rand:    0.5445148639117267   0.15269428321272993

Teste refinamento com largura 15% feito:
   Média     Desvio padrão
tempo:    0.1313690423965454   0.004819820252834578
raio:    26.300240557834169   7.971944832830111
silhueta:   0.4569864125359471   0.2834686909920847
rand:    0.47769251488740694   0.18415013014814563
```

Teste refinamento com largura 20% feito:

	Média	Desvio padrão
tempo:	0.1302361249923706	0.002805238540588992
raio:	26.398128417534210	6.891371497753807
silhueta:	0.4975571017546024	0.1939992966688405
rand:	0.5244552613954974	0.12230337039531486