Trabalho Prático 2 - Algoritmos 2

Etelvina C. S. Sá Oliveira[1], **Vitor L. de Faria**[1]

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

etelvina.oliveira2003@gmail.com, vtifaria@gmail.com

Abstract. *This work aims to analyze two 2-approximation algorithms for the k-centers problem and the KMeans algorithm, considering computational demand and solution quality.*

Resumo. *O presente trabalho tem como objetivo realizar a análise de dois algoritmos 2-aproximativos para o problema dos k-centros e o KMeans, levando em consideração aspectos de demanda computacional e qualidade da solução.*

1. Introdução

Este trabalho prático aborda a implementação e análise de algoritmos para o problema dos k-centros, um desafio central em tarefas de agrupamento em aprendizado de máquina. Foram implementados dois algoritmos 2-aproximativos e o algoritmo clássico K-Means, com o objetivo de comparar demanda computacional e qualidade das soluções.

O problema dos k-centros consiste em escolher k centros de forma a minimizar a maior distância entre um ponto e seu centro mais próximo. Os algoritmos aproximativos utilizados oferecem soluções com um raio de no máximo duas vezes o ótimo. Além disso, o K-Means foi usado como referência devido à sua eficácia conhecida em tarefas de agrupamento.

Os experimentos foram realizados utilizando datasets reais do UCI Machine Learning Repository e datasets sintéticos, analisando o desempenho dos algoritmos em termos de tempo de execução e qualidade do agrupamento. A análise visa entender a aplicabilidade prática desses algoritmos em diferentes contextos.

2. Métodos e métricas

No trabalho em questão foram implementados três algoritmos que realizam a tarefa de agrupamento, sendo dois deles 2-aproximativos e um algoritmo clássico para o problema de agrupamento.

O primeiro algoritmo aproximativo implementado (k-centros1) se baseia no refinamento de intervalos, em que usamos um algoritmo auxiliar que verifica se é possível gerar um agrupamento com o raio determinado, a cada iteração selecionamos um ponto arbitrário do conjunto de pontos adicionando ele ao conjunto de centros e removemos todos os pontos que estão a uma distância de até 2*r do último ponto considerado. Esse processo é realizado até que todos os pontos tenham sido avalidos e o conjunto de pontos esteja vazio. Por fim, verificamos se o número de centros encontrado é menor ou igual ao número de centros desejado. Com a aplicação desse algoritmo auxiliar partimos de

um intervalo que vai de 0 até a maior distância entre dois pontos do dataset, que é o intervalo que garantidamente contém o raio ótimo, e o refinamos através de uma busca binária em que a cada iteração verificamos se é possível encontrar um agrupamento de tamanho k com o raio dado pela média do limite superior e inferior do intervalo. Nessa implementação recebemos um parâmetro que define a porcentagem do intervalo real que o intervalo refinado deve atingir (parâmetro de refinamento).

O segundo algoritmo aproximativo implementado (k-centros2) inicialmente verifica se o número de centros é maior que o número de pontos, retornando caso seja verdadeiro o conjunto de pontos em si, senão a cada iteração escolhemos um ponto arbitrário do conjunto de pontos e enquanto o numero de pontos no conjunto de centros é menor que o número de centros desejado escolhemos o ponto que maximiza a distância entre os pontos que atualmente estão no conjunto de centros.

Em cada um dos algoritmos aproximativos foi utilizada como medida de distância a distância de Minkwski, que é definida da seguinte forma:

$$D(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Quando atribuímos p = 1 ela é equivalente a distância de Manhattan e quando atribuímos p = 2 ela é equivalente a distância Euclidiana.

O terceiro algoritmo utilizado foi o K-Means, conhecido por sua simplicidade e eficácia em agrupamento. O k-means é um algoritmo que treina um modelo para agrupar objetos semelhantes. Para isso, ele mapeia cada observação no conjunto de dados de entrada para um ponto no espaço de n dimensões (em que n é o número de atributos da observação).

Foram utilizadas três métricas para a avaliação dos algoritmos: o tempo de execução, o raio da solução, a silhueta e o índice de Rand ajustado. No caso das métricas de silhueta e índice de Rand ajustado foi utilizada a implementação disponibilizada pela biblioteca scikit-learn (https://scikit-learn.org/stable/index.html).

O raio da solução é dado como a maior distância de um ponto a um ponto ao centro, sendo uma medida da dispersão dos pontos dentro do cluster.

O Coeficiente de Silhueta é calculado usando a distância média intracluster e a distância média do cluster mais próximo para cada amostra. O melhor valor é 1 e o pior valor é -1. Valores próximos a 0 indicam clusters sobrepostos. Valores negativos geralmente indicam que uma amostra foi atribuída ao cluster errado, pois um cluster diferente é mais similar.

O Índice Rand calcula uma medida de similaridade entre dois agrupamentos considerando todos os pares de amostras e contando os pares que são atribuídos no mesmo agrupamento ou em agrupamentos diferentes nos agrupamentos previstos e verdadeiros.O índice Rand ajustado tem um valor próximo de 0,0 para rotulagem aleatória independentemente do número de clusters e amostras e exatamente 1,0 quando os clusters são idênticos. O índice Rand ajustado é limitado abaixo por -0,5 para clusters especialmente discordantes.

3. Bases de dados

Para a realização do trabalho utilizamos os dados de 3 fontes principais a UCI Machine Learning Repository, a geração sintética 1 por meio do scikit, e a geração sintética 2 em duas dimensões utilizando a distribuição normal multivariada. Todas as bases de dados consideradas possuem um número de instâncias maior que 700.

3.1. UCI Machine Learning Repository

Foram utilizados 10 conjuntos de dados da base de dados do UCI Machine Learning Repository, esses foram escolhidos baseados no critério de possuírem a maioria dos labels em formato numérico, para que assim precisasse de pouco pré tratamento na parte do código, os conjuntos foram:

3.1.1. Raisin

Imagens das variedades de passas Kecimen e Besni cultivadas na Turquia foram obtidas com CVS. Um total de 900 grãos de passas foram usados, incluindo 450 pedaços de ambas as variedades. Essas imagens foram submetidas a vários estágios de pré-processamento e 7 características morfológicas foram extraídas. Essas características foram classificadas usando três técnicas diferentes de inteligência artificial.

3.1.2. Image Segmentation

As instâncias foram extraídas aleatoriamente de um banco de dados de 7 imagens externas. As imagens foram segmentadas manualmente para criar uma classificação para cada pixel. Cada instância é uma região 3x3.

3.1.3. Maternal Health Risk

Dados foram coletados de diferentes hospitais, clínicas comunitárias e unidades de saúde materna de áreas rurais de Bangladesh por meio do sistema de monitoramento de risco baseado em IoT.

3.1.4. Mammographic Mass

Discriminação de massas mamográficas benignas e malignas com base nos atributos BI-RADS e na idade do paciente.

3.1.5. Optical Recognition of Handwritten Digits

Usamos programas de pré-processamento disponibilizados pelo NIST para extrair bitmaps normalizados de dígitos manuscritos de um formulário pré-impresso. De um total de 43 pessoas, 30 contribuíram para o conjunto de treinamento e 13 diferentes para o conjunto de teste. Bitmaps 32x32 são divididos em blocos não sobrepostos de 4x4 e o

número de pixels é contado em cada bloco. Isso gera uma matriz de entrada de 8x8 onde cada elemento é um inteiro no intervalo de 0 a 16. Isso reduz a dimensionalidade e dá invariância a pequenas distorções.

3.1.6. Statlog (Vehicle Silhouettes)

Objetos 3D dentro de uma imagem 2D pela aplicação de um conjunto de extratores de características de forma às silhuetas 2D dos objetos.

3.1.7. Yeast

Atributo previsto: Sítio de localização da proteína. (não numérico). As referências abaixo descrevem um predecessor deste conjunto de dados e seu desenvolvimento. Elas também fornecem resultados (não validados de forma cruzada) para classificação por um sistema especialista baseado em regras com essa versão do conjunto de dados.

3.1.8. Blood Transfusion Service Center

Para demonstrar o modelo de marketing RFMTC (uma versão modificada do RFM), este estudo adotou o banco de dados de doadores do Blood Transfusion Service Center na cidade de Hsin-Chu em Taiwan. O centro passa seu ônibus de serviço de transfusão de sangue para uma universidade na cidade de Hsin-Chu para coletar sangue doado a cada três meses. Para construir um modelo FRMTC, selecionamos 748 doadores aleatoriamente do banco de dados de doadores. Esses 748 dados de doadores, cada um incluía R (Recência - meses desde a última doação), F (Frequência - número total de doações), M (Monetário - total de sangue doado em cc), T (Tempo - meses desde a primeira doação) e uma variável binária representando se ele/ela doou sangue em março de 2007 (1 significa doação de sangue; 0 significa não doação de sangue).

3.1.9. Abalone

Prevendo a idade do abalone a partir de medições físicas. A idade do abalone é determinada cortando a concha através do cone, colorindo-a e contando o número de anéis através de um microscópio – uma tarefa chata e demorada. Outras medições, que são mais fáceis de obter, são usadas para prever a idade. Mais informações, como padrões climáticos e localização (portanto, disponibilidade de alimentos) podem ser necessárias para resolver o problema. Dos dados originais, exemplos com valores ausentes foram removidos (a maioria com o valor previsto ausente), e os intervalos dos valores contínuos foram dimensionados para uso com uma ANN (dividindo por 200).

3.1.10. Diabetic Retinopathy Debrecen

Este conjunto de dados contém recursos extraídos do conjunto de imagens Messidor para prever se uma imagem contém sinais de retinopatia diabética ou não. Todos os recursos

representam uma lesão detectada, um recurso descritivo de uma parte anatômica ou um descritor de nível de imagem.

3.2. Sintética 1

A primeira abordagem para a geração dos 10 conjuntos de dados sintéticos baseou-se no material "Comparing different clustering algorithms on toy datasets"usando scikit-learn. Foram gerados datasets como noisy circles e noisy moons, que incluem formas não lineares e ruído, além de blobs com diferentes variâncias e um dataset anisotrópico resultante de uma transformação linear em blobs. Também foram criados conjuntos baseados em quantis gaussianos. Cada dataset foi salvo em formato CSV, facilitando sua utilização nos experimentos de clustering.

3.3. Sintética 2

Na segunda abordagem, foram gerados 10 conjuntos de dados sintéticos em duas dimensões utilizando a distribuição normal multivariada. Para cada conjunto, foram criados centros de distribuição com médias aleatórias, variando o desvio padrão para controlar a sobreposição entre os grupos, desde inexistente até altamente sobrepostos. A função generate multivariate data foi utilizada para amostrar pontos em torno desses centros, resultando em uma distribuição variada de pontos para cada centro. Cada conjunto foi salvo em formato CSV, com os rótulos dos centros incluídos para identificar a origem dos pontos.

4. Descrição dos experimentos

Os experimentos foram conduzidos da seguinte forma: no caso dos algoritmos aproximativos, a escolha dos centros é arbitrária e pode influenciar na qualidade da solução, desse modo foram realizadas 30 experimentos para cada conjunto de dados. Além disso, para cada uma das execuções calculamos cada uma das métricas mencionadas para possibilitar a comparação. Para auxiliar a execução dos algoritmos aproximados foi calcula uma matriz de distâncias de todos os pontos, com o objetivo de otimizar a execução dos algoritmos, visto que é necessário realizar seu cálculo apenas uma vez para cada base de dados.

4.1. K-centros 1

Nos experimentos utilizando o algoritmo K-centros 1 utilizamos como parâmetro de refinamento os valores 5%, 10%, 15%, 20% e 25%. Desse modo, foram realizadas 30 iterações para cada um desses parâmetros e com cada uma das medidas de distância.

Assim, inicialmente utilizamos uma função que calcula a distância máxima entre dois pontos do dataset e refinamos o intervalo que ele tenha x% do tamanho do intervalo original.

4.2. K-centros 2

Os experimentos utilizando o K-centros 2 foram realizadas de forma semelhante ao K-centros 1, em que foram realizadas 30 iterações para cada um dos conjuntos de dados.

4.3. K-means

Os experimentos com o algoritmo K-Means foram organizados de forma estruturada no código, começa com a função mainKmeans, que coordena todo o processo de execução.

A função carregar datasets é responsável por carregar os datasets sintéticos previamente gerados. Dependendo do tipo de dataset (sintético 1 ou 2), ela carrega os arquivos CSV correspondentes.

Para cada dataset carregado, a função processardatasetskmeans é chamada. Esta função aplica o K-Means nos dados,depois determina o número de clusters com base no número de classes únicas (y.nunique()).

A execução do K-Means é realizada pela função kmeansexec, que roda o algoritmo 30 vezes para cada dataset, variando a inicialização dos centroides. Durante cada execução, são calculadas as métricas de desempenho, como Silhueta, Índice de Rand Ajustado, Raio Máximo, e o tempo de execução.

Além dos datasets sintéticos, a função datasetUCI processa os datasets reais obtidos do UCI Machine Learning Repository. Ela carrega os dados, ajusta-os conforme necessário (remove colunas ou lida com valores ausentes), e executa o K-Means,no final, salva os resultados em arquivos CSV específicos.

Após o processamento de todos os datasets, os resultados são organizados em dataframes e salvos em arquivos CSV. Isso permite uma análise posterior dos resultados para comparar o desempenho do K-Means em diferentes tipos de dados.

5. Apresentação dos Resultados

Tabela dos resultados consolidados para o algoritmo do k-means:

tabela_consolidada_Kmeans (1)

Algoritmo	Tempo Execução (s)	Silhueta	Índice de Rand Ajustado	Raio Máximo	Dataset
K-Means	0.0572 ± 0.0209	0.6542 ± 0.0001	0.1665 ± 0.0005	411985.0973 ± 8.0829	
K-Means	0.0933 ± 0.0445	0.3326 ± 0.0055	0.4055 ± 0.0186	2768.7261 ± 19.1158	
K-Means	0.4528 ± 0.1275	0.2722 ± 0.0045	0.0452 ± 0.0014	2.6730 ± 0.0820	
K-Means	0.0316 ± 0.0080	0.7025 ± 0.0000	0.0795 ± 0.0000	15223.1532 ± 0.0000	
K-Means	0.0330 ± 0.0135	0.5427 ± 0.0000	0.1367 ± 0.0000	111.0518 ± 0.0000	
K-Means	0.0471 ± 0.0197	0.3543 ± 0.0003	-0.0013 ± 0.0001	2.5921 ± 0.0184	dataset_0.csv
K-Means	0.0306 ± 0.0105	0.4081 ± 0.0000	-0.0014 ± 0.0000	2.3925 ± 0.1473	dataset_1.csv
K-Means	0.0192 ± 0.0097	0.4898 ± 0.0000	0.2461 ± 0.0000	1.8127 ± 0.0004	dataset_2.csv
K-Means	0.0168 ± 0.0016	0.4845 ± 0.0000	0.2461 ± 0.0000	1.9329 ± 0.0000	dataset_3.csv
K-Means	0.0184 ± 0.0016	0.6429 ± 0.0000	0.7938 ± 0.0000	17.9606 ± 0.0000	dataset_4.csv
K-Means	0.0201 ± 0.0042	0.6341 ± 0.0000	0.9414 ± 0.0000	24.4613 ± 0.0000	dataset_5.csv
K-Means	0.0388 ± 0.0155	0.5034 ± 0.0000	0.5905 ± 0.0000	7.4021 ± 0.0024	dataset_6.csv
K-Means	0.0427 ± 0.0145	0.6420 ± 0.0000	0.9829 ± 0.0000	11.3476 ± 0.0000	dataset_7.csv
K-Means	0.0357 ± 0.0160	0.6844 ± 0.0000	0.9957 ± 0.0000	14.2793 ± 0.0000	dataset_8.csv
K-Means	0.0123 ± 0.0019	0.3646 ± 0.0000	0.0475 ± 0.0000	4.5288 ± 0.0000	dataset_9.csv
K-Means	0.0362 ± 0.0061	0.4459 ± 0.0133	0.1204 ± 0.0009	1943.4479 ± 5.4125	
K-Means	0.0416 ± 0.0151	0.3753 ± 0.0000	0.1264 ± 0.0000	178.3239 ± 0.0000	
K-Means	0.0195 ± 0.0029	0.7669 ± 0.0000	0.9896 ± 0.0000	14.8107 ± 0.0000	dataset_0_dist_mult.csv
K-Means	0.0281 ± 0.0025	0.4852 ± 0.0003	0.7409 ± 0.0010	15.8632 ± 0.0000	dataset_1_dist_mult.csv
K-Means	0.0381 ± 0.0136	0.6015 ± 0.0000	0.8455 ± 0.0000	11.4450 ± 0.0000	dataset_2_dist_mult.csv
K-Means	0.0566 ± 0.0158	0.4226 ± 0.0018	0.5783 ± 0.0110	11.5106 ± 0.0000	dataset_3_dist_mult.csv
K-Means	0.0519 ± 0.0116	0.4394 ± 0.0023	0.6899 ± 0.0027	18.2694 ± 0.0061	dataset_4_dist_mult.csv
K-Means	0.0216 ± 0.0063	0.6804 ± 0.0000	0.9755 ± 0.0000	19.1114 ± 0.0000	dataset_5_dist_mult.csv
K-Means	0.0374 ± 0.0054	0.4556 ± 0.0001	0.7559 ± 0.0014	17.3312 ± 0.0143	dataset_6_dist_mult.csv
K-Means	0.0380 ± 0.0045	0.4414 ± 0.0000	0.6859 ± 0.0007	18.5989 ± 0.0000	dataset_7_dist_mult.csv
K-Means	0.0399 ± 0.0052	0.3493 ± 0.0103	0.4726 ± 0.0064	17.4235 ± 0.0871	dataset_8_dist_mult.csv
K-Means	0.0346 ± 0.0042	0.4209 ± 0.0003	0.6684 ± 0.0027	19.8590 ± 0.0293	dataset_9_dist_mult.csv
K-Means	0.0700 ± 0.0262	0.4407 ± 0.0007	-0.0016 ± 0.0000	2255.7181 ± 0.1906	
K-Means	0.9101 ± 0.4208	0.1871 ± 0.0023	0.6719 ± 0.0025	1884.7318 ± 1.0507	
K-Means	0.1492 ± 0.0534	0.1762 ± 0.0069	0.1415 ± 0.0101	3.5747 ± 0.0435	

Figura 1. k-means

Tabelas dos resultados consolidados para o algoritmo os algoritmos k-centros 2 (reias e sinteticas):

tabela_consolidada_Reais_kcentros2

Algoritmo	Tempo Execução (s)	Silhueta	Índice de Rand Ajustado	Raio Máximo
K-Centros 2	0.0005 ± 0.0001	0.6393 ± 0.0000	0.0158 ± 0.0000	156470.4476 ± 0.0000
K-Centros 2	0.0316 ± 0.0038	0.3350 ± 0.0000	-0.0090 ± 0.0000	0.9103 ± 0.0000
K-Centros 2	0.0005 ± 0.0001	0.8504 ± 0.0000	0.0368 ± 0.0000	6048.0000 ± 0.0000
K-Centros 2	0.0015 ± 0.0023	0.6819 ± 0.0000	0.0001 ± 0.0000	49.2138 ± 0.0000
K-Centros 2	0.0016 ± 0.0025	0.6819 ± 0.0000	0.0001 ± 0.0000	56.0000 ± 0.0000
K-Centros 2	0.0020 ± 0.0001	0.1024 ± 0.0000	0.0338 ± 0.0000	371.1860 ± 0.0000
K-Centros 2	0.0053 ± 0.0038	0.3886 ± 0.0000	0.0022 ± 0.0000	90.0000 ± 0.0000
K-Centros 2	0.0023 ± 0.0014	0.1081 ± 0.0000	0.0818 ± 0.0000	957.7717 ± 0.0000
K-Centros 2	0.0459 ± 0.0129	0.4233 ± 0.0000	0.0092 ± 0.0000	1.5500 ± 0.0000
K-Centros 2	0.0027 ± 0.0014	0.1242 ± 0.0000	0.0355 ± 0.0000	335.2357 ± 0.0000
K-Centros 2	0.0005 ± 0.0000	0.8504 ± 0.0000	0.0368 ± 0.0000	6000.0887 ± 0.0000
K-Centros 2	0.0027 ± 0.0011	0.1330 ± 0.0000	0.0296 ± 0.0000	596.0000 ± 0.0000
K-Centros 2	0.0005 ± 0.0001	0.6366 ± 0.0000	0.0131 ± 0.0000	114332.7300 ± 0.0000
K-Centros 2	0.0006 ± 0.0001	0.5459 ± 0.0000	-0.0019 ± 0.0000	562.8169 ± 0.0000
K-Centros 2	0.0006 ± 0.0000	0.5672 ± 0.0000	-0.0013 ± 0.0000	227.8358 ± 0.0000
K-Centros 2	0.0046 ± 0.0028	0.2909 ± 0.0000	-0.0069 ± 0.0000	60.9672 ± 0.0000

Figura 2. k-centros2 reais

tabela_consolidada_sinteticos_kcentros2

Algoritmo	Tempo Execução (s)	Silhueta	Índice de Rand Ajustado	Raio Máximo
K-Centros 2	0.0053 ± 0.0005	0.2985 ± 0.0000	0.3614 ± 0.0000	14.5563 ± 0.0000
K-Centros 2	0.0023 ± 0.0013	0.1411 ± 0.0000	0.3267 ± 0.0000	18.9975 ± 0.0000
K-Centros 2	0.0081 ± 0.0006	0.6392 ± 0.0000	0.4481 ± 0.0000	11.3726 ± 0.0000
K-Centros 2	0.0014 ± 0.0001	0.4874 ± 0.0000	0.6875 ± 0.0000	14.0229 ± 0.0000
K-Centros 2	0.0048 ± 0.0017	-0.0070 ± 0.0000	0.1937 ± 0.0000	11.1692 ± 0.0000
K-Centros 2	0.0050 ± 0.0005	0.4447 ± 0.0000	0.5085 ± 0.0000	13.3017 ± 0.0000
K-Centros 2	0.0076 ± 0.0012	0.2143 ± 0.0000	0.1335 ± 0.0000	18.3425 ± 0.0000
K-Centros 2	0.0069 ± 0.0021	0.2588 ± 0.0000	0.1981 ± 0.0000	11.6828 ± 0.0000
K-Centros 2	0.0033 ± 0.0003	0.6053 ± 0.0000	0.5194 ± 0.0000	6.1819 ± 0.0000
K-Centros 2	0.0024 ± 0.0002	0.4351 ± 0.0000	0.3266 ± 0.0000	7.3422 ± 0.0000

Figura 3. k-centros2 sinteticas

6. Resultados

Faremos a análise dos resultados considerando cada uma das métricas obtidas a partir de nossos experimentos, no geral, desvio padrão baixo indica que os resultados obtidos são consistentes e não apresentam grandes flutuações em torno do valor obtido.

6.1. Tempo de execução

Após a execução dos experimentos foi possível observar que em média o K-Centros 1 apresenta o menor tempo de execução, o que pode ser justificado pela simplicidade do algoritmo na realização do agrupamento. O K-Centros 2 apresenta um tempo de execução maior, visto que, a cada iteração em que é realizado um refinamento, é necessário rodar uma nova clusterização e verificar se ela é valida. Já o K-Means possui um tempo de execução intermediário entre os dois algoritmos apresentados.

É válido mencionar que não foi considerado no cálculo do tempo de execução da matriz de distâncias, o que poderia elevar o tempo de execução dos algoritmos aproximativos, dando uma vantagem maior para o K-Means nesse sentido.

Não foram observados grandes impactos no uso de diferentes distâncias no tempo de execução. Pode-se observar que o tempo de execução é menor com o aumento da porcentagem do refinamento no algoritmo K-Clusters 1.

6.2. Raio da solução

Em relação ao raio da solução para cada uma das bases de dados pode-se que o raio da solução encontrado é menor para o algoritmo K-Clusters 1, o que era esperado, visto que a proposta do algoritmo é fazer refinamentos no intervalo inicial até que o intervalo atual corresponda a x% do intervalo inicial. O resultado é ainda melhor considerando um valor de porcentagem menor em que irá ocorrer um maior número de refinamentos.

O K-Cluster 2 apresenta um resultado melhor resultado se comparado o KMeans, também pode estar relacionado com a proposta do algoritmo de a cada iteração reduzir a maior distancia de um ponto aos centros obtidos até o momento.

Além disso, também foi possível observar que ao considerar a distância euclidiana obtemos valores de raio menor que ao considerar a distância de Manhattan.

6.3. Silhueta

Em relação a métrica de silhueta pode-se observar que em alguns casos não são observadas diferenças significativas entre os algoritmos, no entanto, o K-Centros 1 tende a apresentar valores melhores em comparação com os demais algoritmos, o que indica que foram gerados clusters mais separados, com menos sobreposição utilizando essa abordagem. Além disso, temos resultados melhores com a distância euclidiana, em comparação com a distância de Manhattan. Também foi possível observar que a redução da porcentagem do parâmetro de refinamento produz resultado melhores em relação à métrica em questão.

6.4. Indice de Rand ajustado

Em relação a métrica de índice de Rand ajustado pode-se observar que os algoritmos apresentam um melhor desempenho nos dados sintético em comparação aos dados reais. O que mostra que em alguns casos os algoritmos obtiveram como resultados clusters diferentes dos reais.

Nesse caso o K-Clusters 1 apresenta melhores resultados com um valor de porcentagem de refinamento menor e com o uso da distância de Manhattan. Os resultados obtidos são melhores do que

Referências

[1] sklearn.metrics.silhouette_score, *Scikit-learn Documentation*, 2024. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html. Acessado em: 15 de agosto de 2024.

[2] Notas técnicas do algoritmo k-means, *Amazon Web Services Documentation*, 2024. Disponível em: https://docs.aws.amazon.com/pt_br/sagemaker/latest/dg/algo-kmeans-tech-notes.html#:~:text=O%20k%2Dmeans%20%C3%A9%20um,n%C3%BAmero%20de%20atributos%20da%20observa%C3%A7%C3%A3o). Acessado em: 15 de agosto de 2024.

[3] sklearn.metrics.adjusted_rand_score, *Scikit-learn Documentation*, 2024. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html. Acessado em: 15 de agosto de 2024.

Referências