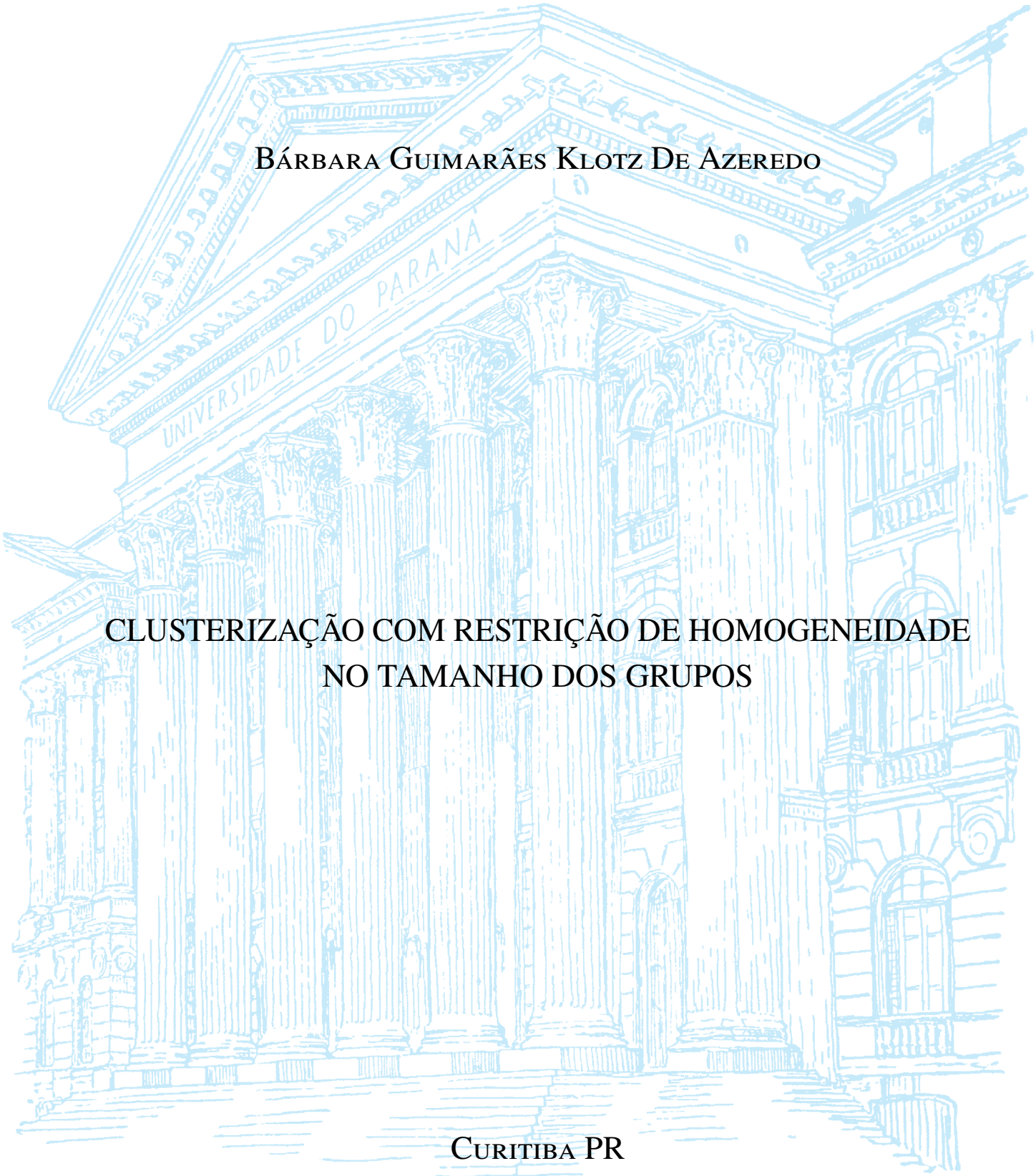
Universidade Federal do Paraná

Bárbara Guimarães Klotz De Azeredo



CLUSTERIZAÇÃO COM RESTRIÇÃO DE HOMOGENEIDADE NO TAMANHO DOS GRUPOS

Curitiba PR
2023

Bárbara Guimarães Klotz De Azeredo

CLUSTERIZAÇÃO COM RESTRIÇÃO DE HOMOGENEIDADE NO TAMANHO DOS GRUPOS

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Matemática Industrial, setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Matemática Industrial*.

Orientador: Prof. Dr. Lucas Garcia Pedroso..

Curitiba PR
2023

Agradecimentos

Gostaria de expressar minha mais profunda gratidão a toda a minha família e amigos, com especial ênfase em minha mãe, Claudia, e em meu pai, Eugênio. Sua inquestionável dedicação e constante apoio têm sido pilares fundamentais ao longo de toda a minha jornada, inundando-me de motivação e amor. Também gostaria de agradecer a todos os meus colegas cuja presença e auxílio foram indispensáveis durante os anos dedicados à minha formação acadêmica. Não posso deixar de mencionar meu orientador, o Professor Dr. Lucas Garcia Pedroso, por transmitir seus ensinamentos com paciência e clareza, além de dedicar seu tempo para me orientar neste trabalho. Agradeço igualmente a todos os professores e professoras do Departamento de Matemática da Universidade Federal do Paraná, que fizeram parte da minha jornada e contribuíram de alguma forma para o meu desenvolvimento acadêmico. Em especial, quero expressar minha admiração e gratidão ao Professor Luiz Antonio Ribeiro de Santana, cuja incansável disposição em auxiliar os discentes de Matemática Industrial com atenção e serenidade foi exemplar.

Resumo

Este trabalho consiste em uma revisão dos conceitos principais sobre clusterização, o código K-means e Aprendizagem de Máquinas não supervisionada. O objetivo é analisar um algoritmo de clusterização que contém a mesma quantidade de pontos em todos os clusters. Foram realizados experimentos numéricos em MATLAB para validar o desempenho do algoritmo em diversos datasets. Alem disso, o trabalho inclui uma comparação do algoritmo proposto e o código K-means e finaliza trazendo algumas ideias de pontos a melhorar e as conclusões obtidas.
Palavras-chave: Clusterização, K-means.

Abstract

This paper consists of a review of the main concepts of clustering, the K-means algorithm, and unsupervised Machine Learning. The objective is to analyze a clustering algorithm that ensures an equal number of points in each cluster. Numerical experiments were conducted in MATLAB to validate the algorithm's performance on various datasets. The study also includes a comparison between the proposed algorithm and the K-means method, and concludes by suggesting areas for improvement and presenting the obtained conclusions.

Keywords: Clustering, K-means.

Lista de Figuras

Sumário

Capítulo 1

Introdução

A clusterização é uma classificação não supervisionada ou segmentação não supervisionada. O objetivo é atribuir instâncias a classes que não são definidas previamente e que supostamente refletem de alguma forma a estrutura subjacente das entidades representadas pelos dados. Na literatura mais ampla, não relacionada à Aprendizagem de Máquina, é comum usar a palavra classificação ao falar de clusterização [1] [2]. A clusterização relaciona dados ao conhecimento e é uma atividade humana básica. Em [1], os autores argumentam o quão fundamental ela é para a compreensão do mundo. Ela afeta a representação e descoberta de conhecimento [4]. A clusterização desperta um grande fascínio entre matemáticos e engenheiros, o que resulta em uma vasta literatura sobre técnicas de clusterização independentes de domínio. Embora as técnicas de classificação supervisionada sejam amplamente apreciadas e utilizadas para resolver problemas reais, a clusterização é frequentemente considerada uma forma de classificação não supervisionada. No entanto, existe uma diferença fundamental entre elas: a clusterização supervisionada pode ser transformada de maneira mais direta em um problema bem definido, incorporando uma função de perda que formaliza precisamente a tarefa que se deseja realizar. Essa função de perda pode ser vista como uma abstração do problema final de utilização, sendo fundamentada racionalmente no contexto do problema real subjacente. Muitas vezes presume-se que, para qualquer situação em que a clusterização possa ser usada, existe uma única clusterização correta. O objetivo da clusterização de dados é descobrir os agrupamentos naturais de um conjunto de padrões, pontos ou objetos [3]. Existem aqueles que vão além e defendem que a resposta correta pode ser determinada exclusivamente pelos dados, sem fazer referência ao uso pretendido. Essa abordagem postula que os dados devem votar em seu tipo e complexidade de modelo preferidos. Neste artigo, é apresentado um modelo específico de clusterização que possui como característica distintiva a atribuição de uma quantidade igual de pontos a cada cluster. Essa abordagem tem potencial de utilidade em diversos problemas reais, tais como a divisão de alunos em salas de tamanho igual com base em seus perfis, a distribuição de produtos em setores de uma loja com base na similaridade dos produtos, garantindo que os setores tenham tamanhos equivalentes, entre outros usos variados. No trabalho de João V. Pamplona [5], que utilizamos como referência nesse artigo, foi proposto um algoritmo que leva em consideração a restrição de capacidade, mas a ideia desse projeto é propor algo mais específico, o cenário sem restrições de capacidade porem onde todos os grupos têm exatamente a mesma quantidade de dados.

Capítulo 2

Conceitos importantes

Este capítulo aborda algumas noções básicas que são fundamentais para compreender o trabalho em questão. Dentre esses conceitos, destacam-se a clusterização, que é um dos algoritmos mais renomados com o objetivo de agrupar dados, o K-Means, bem como a distinção entre os tipos de Aprendizado de Máquina e sua relação com o estudo apresentado. Nesse capítulo, o principal ponto de referência utilizado foi o trabalho de João Vitor Pampola [**?**, **?**] que traz noções sobre clusterizacao, código K-Means e apresenta um algoritmo com homogeneidade no tamanho dos clusters, porém com restrição de capacidade - diferentemente desse projeto.

2.1 Aprendizagem De Máquinas Não Supervisionada

O Aprendizado não Supervisionado é uma abordagem do campo de Aprendizado de Máquina em que os algoritmos são capazes de adquirir conhecimento de forma autônoma, sem depender de exemplos rotulados. Nesse contexto, o algoritmo é desafiado a compreender a estrutura subjacente dos dados, sem ter acesso às saídas esperadas. Dessa forma, o algoritmo em questão não realiza nenhum tipo de pré-processamento ou tratamento dos dados, mas busca identificar padrões e relações entre eles por conta própria. Em seguida, o algoritmo procura agrupar os dados com base em suas similaridades descobertas. Como não dispomos de rótulos para orientar o processo, não é necessário realizar testes ou avaliações específicas.

2.2 Clusterização

O processo de clusterização consiste em agrupar, categorizar ou organizar qualquer tipo de dado conforme as suas características incomuns. Basicamente separar os grupos com características semelhantes e atribuí-los em grupos, os clusters. A classificação em clusters é feita usando diversos critérios como densidade de pontos de dados, distâncias menores, várias distribuições estatísticas ou gráficos, visando armazenar os dados, processá-los e analisá-los de maneira ordenada. Existem diversos métodos de clusterização sendo alguns deles: A hierárquica, particionada, a baseada em distribuição ou baseada em densidade.

A clusterização hierárquica baseada em conectividade é um método de Aprendizado de Máquina não Supervisionado que organiza os dados em uma estrutura hierárquica de clusters. Inicia-se com uma classificação pré-definida dos clusters de alto nível e, em seguida, os dados são decompostos com base nessa estrutura para formar os clusters. Existem duas abordagens nesse método, dependendo da direção do progresso na criação dos clusters:

1. Abordagem divisiva: Começando de cima para baixo, todos os pontos de dados são considerados pertencentes a um único grande cluster. Em seguida, o objetivo é dividir esse cluster em grupos menores com base em algum critério de parada ou ponto em que não seja possível mais dividir os dados. Assim, os dados são separados em *n* clusters menores aos quais os pontos de dados agora pertencem.

2. Abordagem aglomerativa: Em contraste com a abordagem divisiva, essa prática combina iterativamente vários clusters para formar clusters maiores e, assim, atribuir os pontos de dados a esses clusters. Essa abordagem é ascendente e também utiliza critérios de parada, como número máximo de clusters, distância entre os clusters ou variação dentro do cluster, para determinar quando parar de mesclar.

A clusterização particionada baseada em centroides é um método simples de agrupamento. Sua ideia fundamental é que cada cluster é representado por um centro e os pontos de dados que estão próximos a esse centro são atribuídos ao mesmo cluster. No entanto, um desafio importante é a necessidade de definir o número de clusters de forma intuitiva ou utilizando métodos como o método do cotovelo, para iniciar a iteração do algoritmo de Aprendizado de Máquina e realizar a atribuição dos pontos de dados aos clusters. Apesar de suas limitações, o agrupamento baseado em centralidade tem se mostrado mais eficaz do que o método hierárquico quando aplicado a conjuntos de dados grandes. Além disso, devido à sua simplicidade na implementação e interpretação, esses algoritmos têm uma ampla gama de aplicações, como segmentação de mercado, segmentação de clientes, análise de imagens e recuperação de tópicos em texto.

A clusterização baseada em densidade é um método que difere dos algoritmos hierárquicos e baseados em centralidade, pois não depende explicitamente de métricas de distância entre os pontos de dados. Em vez disso, os clusters são identificados como regiões de alta densidade em um espaço de dados, separadas por áreas de menor densidade, e são definidos como conjuntos máximos de pontos conectados. Esse método é utilizado assumindo duas premissas principais: os dados não contêm ruído significativo e a forma dos clusters formados é puramente geométrica, como circular ou elíptica. No entanto, na realidade, os dados frequentemente apresentam algum nível de inconsistência ou ruído, que não pode ser ignorado. Além disso, não devemos restringir os clusters a formas fixas, pois é desejável poder identificar clusters com formas arbitrárias, a fim de não excluir nenhum ponto de dados relevante. Na prática, os algoritmos baseados em densidade podem fornecer clusters com as seguintes características:

- Formas arbitrárias.
- Tamanhos variados, sem limitações.
- Alta homogeneidade dentro dos clusters.
- Níveis consistentes de densidade.
- Capacidade de lidar com inconsistências nos dados.

Essas características tornam a clusterização baseada em densidade uma abordagem flexível e poderosa para a análise de dados, permitindo a identificação de clusters com diferentes formas e tamanhos, preservando a homogeneidade e adaptando-se a possíveis inconsistências nos dados.

A clusterização baseada em distribuição é um método que agrupa pontos de dados com base na probabilidade de pertencerem à mesma distribuição de probabilidade, como a distribuição Gaussiana ou Binomial. Esse método de clusterização está mais relacionado ao campo da

Estatística, pois lida com a maneira como conjuntos de dados são gerados e organizados, muitas vezes utilizando princípios de amostragem aleatória. Os clusters são definidos como objetos que pertencem à mesma distribuição, o que permite uma interpretação estatística dos agrupamentos.

Uma vantagem dessa abordagem é a flexibilidade, precisão e forma dos clusters criados. No entanto, um desafio importante é que esse método funciona melhor com dados sintéticos ou simulados, pois requer conhecimento prévio sobre as distribuições subjacentes dos dados.

2.3 Algoritmo K-Means

O algoritmo K-Means é uma técnica pertencente ao campo de Aprendizado de Máquina não Supervisionado, cujo objetivo é agrupar dados que não possuem classificações ou categorizações prévias. Seu propósito é particionar os dados em k clusters, onde o valor de k é determinado pelo usuário antes da execução do algoritmo. No K-Means, utiliza-se o conceito de centroide, que consiste em selecionar aleatoriamente k registros como representantes iniciais de cada cluster. Em seguida, para cada registro restante, calcula-se a similaridade entre o registro em análise e o centroide de cada cluster. O objeto de dados é atribuído ao cluster cujo centroide apresenta a menor distância, indicando maior similaridade. O centroide do cluster é recalculado a cada nova iteração. Os passos fundamentais do algoritmo K-Means são:

1. Definir o número desejado de clusters k.
2. Encontrar os centroides iniciais para cada cluster.
3. Associar cada objeto de dados ao seu centroide mais próximo. 4. Recalcular os centroides.

O método opera por meio de iterações repetidas até que os objetos de dados não alterem seus centros de clusters, indicando que eles se estabilizaram em seus respectivos clusters. No caso de empate na distância mínima entre um ponto e certos centroides, a classe é escolhida de acordo com algum critério definido pelo usuário, como atribuir ao primeiro centroide. Esse método apesar de ser amplamente utilizado tem algumas vantagens e desvantagens, sendo elas:

Vantagens:

- O algoritmo K-Means envolve poucos cálculos computacionais e comparações de distância entre os pontos de dados e os grupos. Portanto, pode ser computacionalmente mais rápido do que outros tipos de agrupamento.
- O K-Means é um algoritmo simples e pode ser facilmente implementado.

Desvantagens:

- O algoritmo não se sai bem com conjuntos de dados não globulares, ou seja, dados em formatos não esféricos.
- O número de grupos precisa ser especificado pelo usuário.
- Como o K-Means seleciona aleatoriamente os centroides iniciais para os grupos, os resultados podem variar de uma execução para outra, falta de consistência.

Algoritmo 1 - K-Means
Dados de entrada: base de dados com i instâncias e d dimensões, K grupos desejados

Inicio
Inicializa os k centroides com valores aleatórios;
enquanto critério de parada não é atingido;
para cada amostra x_i **faça**
Adiciona x_1, ao grupo do centroide $C_k$de menor distância;
fim
Atualiza os centroides C_k de acordo com a equação.
fim

2.4 Conclusão

Nesse capítulo foram apresentados alguns conceitos importantes com a intenção de preparar o leitor para um entendimento mais aprofundado dos próximos capítulos.

Capítulo 3

Algoritmo Com Homogeneidade No Tamanho Dos Grupos

Nesse capítulo será apresentado o algoritmo criado pelo Prof. Dr. Lucas Garcia Pedroso (Algoritmo 2), professor da Universidade Federal do Paraná, que tem como principal objetivo uma clusterização com restrição de homogeneidade no tamanho dos grupos. Inclui também a motivação para criação desse algoritmo e possíveis aplicações. Por fim, uma explicação detalhada do uso do algoritmo assim como o seu respectivo pseudocódigo.

3.1 Motivação

Existem inúmeros algoritmos de clusterização amplamente utilizados na internet, porém a maioria deles se concentra principalmente na natureza intrínseca dos dados e pode não atender necessariamente às necessidades específicas do usuário. Como resultado, foram introduzidas variações e modificações, como o algoritmo encontrado em [5], que inclui restrições de capacidade e a possível exigência de cada grupo de dados conter a mesma quantidade de pontos.

3.2 Possíveis Aplicações

O algoritmo em questão pode ser aplicado a qualquer tipo de conjunto de dados quando o objetivo é obter homogeneidade no tamanho dos grupos, característica que o diferencia dos algoritmos comuns de clusterização. As possibilidades de aplicação são vastas e nesta Seção exploraremos algumas delas. Uma das possibilidades mais intuitivas de aplicação do algoritmo é a divisão de um número de pessoas em um conjunto de k grupos de tamanho equivalente, com base em características comuns, para fins como alocação de salas, realização de pesquisas, entre outros. Por exemplo, considere uma empresa com três níveis de fidelidade: Ouro, Prata e Bronze. Cada grupo deve conter a mesma quantidade de pessoas. Dependendo do nível de fidelidade de cada cliente, há benefícios específicos na loja. É necessário realizar a divisão de todos os clientes em grupos de forma equitativa, levando em consideração fatores como tempo de fidelidade, gastos mensais, média de gastos ao longo de um ano, quantidade de compras, entre outros. Essa análise é realizada mensalmente, visando garantir que cada grupo tenha a mesma quantidade de pessoas. Nesse contexto, o algoritmo proposto pode ser extremamente útil para orientar a tomada de decisão nessa divisão. Outra aplicação relevante seria a divisão de produtos. Imagine um cenário fictício no qual temos cinco lojas e uma quantidade x de produtos disponíveis. É necessário que cada loja tenha a mesma quantidade de produtos,

respeitando tanto a capacidade de cada estabelecimento quanto a distribuição igualitária de produtos entre as filiais. Além disso, a alocação dos produtos visa maximizar as vendas, levando em consideração o público-alvo de cada região da cidade. Para realizar essa divisão de maneira eficiente, podem ser consideradas diversas características, tais como valor, idade do público-alvo de cada produto e outras informações relevantes. Um algoritmo baseado na abordagem proposta pode analisar uma abundância de informações sobre cada produto, buscando prever o melhor resultado possível. É fundamental destacar que a eficácia desse algoritmo depende da descrição precisa das características de cada ponto, pessoa ou produto, permitindo uma divisão mais apurada e alcançando os objetivos estabelecidos. É inegável que as possibilidades de utilização do algoritmo são inúmeras, abrangendo desde o auxílio a empresas até o campo da pesquisa Matemática, entre tantas outras áreas. No Capítulo 4, poderemos observar o algoritmo em ação, proporcionando aos leitores uma visão mais clara de sua eficácia e das possibilidades que ele oferece. Essa Seção permitirá uma compreensão mais aprofundada do algoritmo por meio de exemplos práticos, ilustrando seu potencial e seus resultados.

3.3 O Algoritmo

O algoritmo a ser analisado foi desenvolvido pelo Prof. Dr. Lucas Garcia Pedroso, motivado pelas razões mencionadas anteriormente na Seção 3.2. Inicialmente, o código foi desenvolvido utilizando a linguagem de programação MATLAB. O autor dedicou-se a projetar e implementar esse algoritmo com o objetivo de atender às necessidades específicas de divisão de grupos com tamanhos equivalentes. Segue o pseudocódigo do algoritmo para em sequência ser analisado assertivamente seus passos.

Dados de entrada: dataset $x \in \mathbb{R}^{n\times d}$, número de grupos g, centroides iniciais $c \in \mathbb{R}^{g\times d}$.
Para $k = 1, 2, ...$
$u_j \leftarrow 0$, $j = 1, ..., n$.
$\mu_i \leftarrow 0$, $i = 1, ..., g$.
Calcule a matriz de distâncias D, com $d_{ij} = ||c_i - x_j||^2$, $i = 1, ..., g$, $j = 1, ..., n$.
$\overline{D} \leftarrow D$.
Para $j = 1, ..., n$
Encontre j' que minimize $\overline{d}_{ij}$.
Encontre i' que minimize $d_{ij'}$.
$u_{j'} \leftarrow i'$.
$I \leftarrow \{1, ..., g\}$.
Enquanto $\mu_{i'} = n/g$
Encontre i'', j'' que minimizem $d_{ij} - d_{i'j}$ para $i'' \in I$ e j'' tal que $u_{j''} = i'$.
$u_{j''} \leftarrow i''$.
$i' \leftarrow i''$.
$I \leftarrow I - \{i'\}$.
$\overline{d}_{ij'} \leftarrow \infty$.
$\mu_{i'} \leftarrow \mu_{i'} + 1$.
Se $\mu_{i'} = n/g$
$\overline{d}_{i'j} \leftarrow \infty$, $j = 1, ..., n$.
Para $i = 1, ..., g$

$$c_i = \frac{\sum_{u_j=i} x_j}{n/g}.$$

Os parâmetros de entrada do algoritmo são o dataset, que possui tamanho n por d e contém dados reais, e a quantidade de grupos desejada pelo usuário. O primeiro passo consiste em definir os centroides iniciais de maneira aleatória. A partir desse ponto, inicia-se a execução propriamente dita do algoritmo. Este pseudocódigo foi redigido de forma simplificada e não inclui o critério de parada, porém o código se encerra quando não ha alterações na alocação dos pontos por clusters de uma iteração a outra. Nas primeiras linhas do pseudocódigo são definidos dois vetores, o u e o μ.

- O vetor u que guarda as atribuições dos clusters, ponto a ponto
- O vetor μ que guarda quantos pontos tem em cada grupo até o momento, lembrando que no final cada grupo tera a mesma quantidade de pontos.

O próximo passo é calcular a matriz de distâncias denominado D, sendo c_i o centroide de número i e x_j o j-ésimo ponto do dataset. A matriz $\overline{D}$ será o D auxiliar que inicialmente guarda as distâncias.
O índice j' é calculado como aquele que realiza a menor distancia na matriz $\overline{D}$. Já o i', é o grupo mais próximo desse ponto, que pode ou não estar cheio. Entraremos no enquanto $\mu_{i'} = n/g$ apenas se o grupo i' já estiver cheio. Dentro do enquanto fazemos trocas de atribuições previamente feitas, de modo a acomodar o novo ponto $x_{j'}$ ao grupo i', porém tentando aumentar a função objetivo o mínimo possível. Subtrai-se um elemento do grupo saturado e adiciona-se a outro grupo. Se esse novo grupo também estiver saturado, o processo é repetido enquanto essa condição persistir. Cada grupo pode ser visitado apenas uma vez, o que é controlado pelo conjunto I.
Apos o enquanto, para todo o elemento i em $\overline{D}_{ij'}$ recebem ∞ para evitar que o ponto $x_{j'}$ seja escolhido mais uma vez para ser atribuído.
'Se $\mu_{i'} = n/g$' significa que o grupo i' está cheio, nesse caso é colocado toda a linha correspondente a esse grupo como infinito na matriz $\overline{D}$.
O passo final consiste em calcular o centroide como o ponto médio de cada grupo, seguindo a abordagem similar ao algoritmo K-Means.

3.4 Conclusão

Neste capítulo, abordou-se o objetivo central deste projeto, que consiste na motivação para o desenvolvimento do algoritmo proposto. Ademais, foi fornecida uma descrição detalhada do algoritmo, incluindo seu funcionamento e a apresentação formal do pseudocódigo correspondente. O próximo capítulo irá descrever uma série de experimentos numéricos com o intuito de proporcionar ao leitor um maior entendimento do funcionamento do algoritmo.

Capítulo 4

Experimentos Numéricos

Este capítulo apresentará uma série de experimentos numéricos utilizando o Algoritmo 2 proposto na Seção 3.3 do presente projeto. O objetivo é aprimorar a compreensão ao observar o algoritmo em ação e compreender melhor sua finalidade, ao compará-lo com o código do algoritmo K-Means. Os resultados do código serão analisados em relação a diversos conjuntos de dados, e o leitor poderá acompanhar esses resultados por meio de gráficos gerados no MATLAB.

4.1 O algoritmo em execução

Nesse primeiro exemplo utilizaremos o Algoritmo 2 e uma escolha de quantidade de clusters (g) igual a cinco.

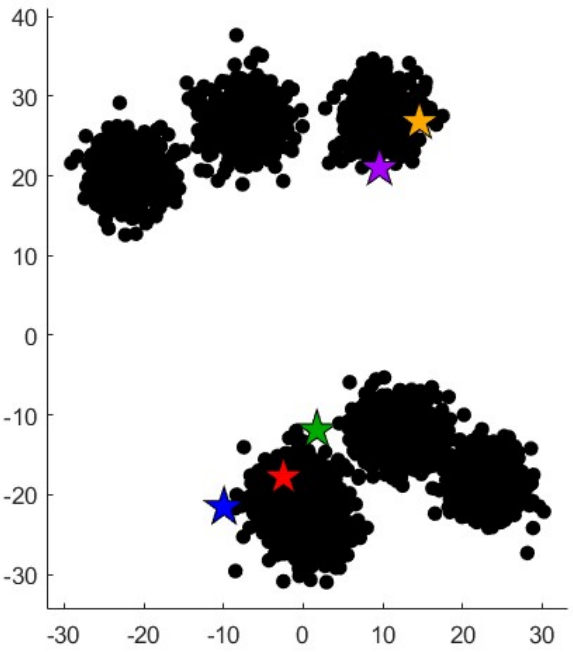
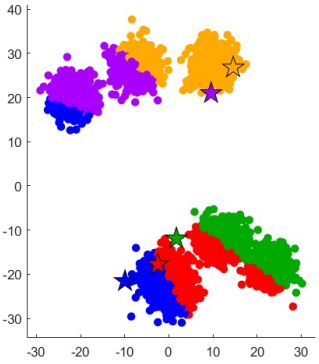
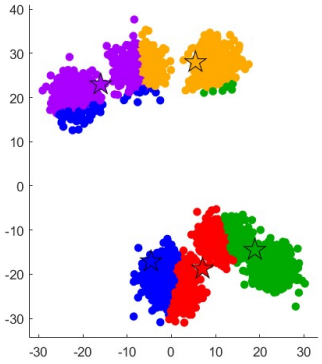


Figura 4.1: Gráfico inicial do dataset e escolha aleatória de clusters para g = 5.

A partir desse ponto iremos visualizar o algoritmo em execução e como a clusterização é feita em cada iteração.

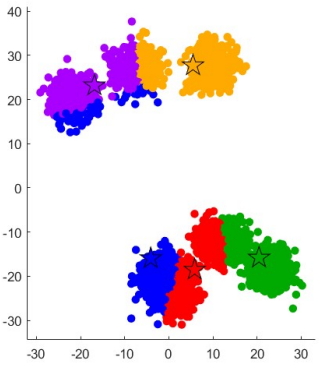


(a) Iteração 1.

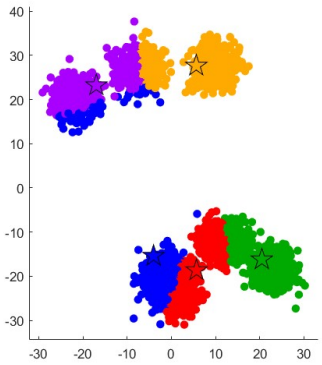


(b) Iteração 2.

Figura 4.2: Primeiras iterações do algoritmo com g = 5.



(a) Iteração 3.



(b) Iteração 4.

Figura 4.3: Próximas iterações do algoritmo com $g = 5$.

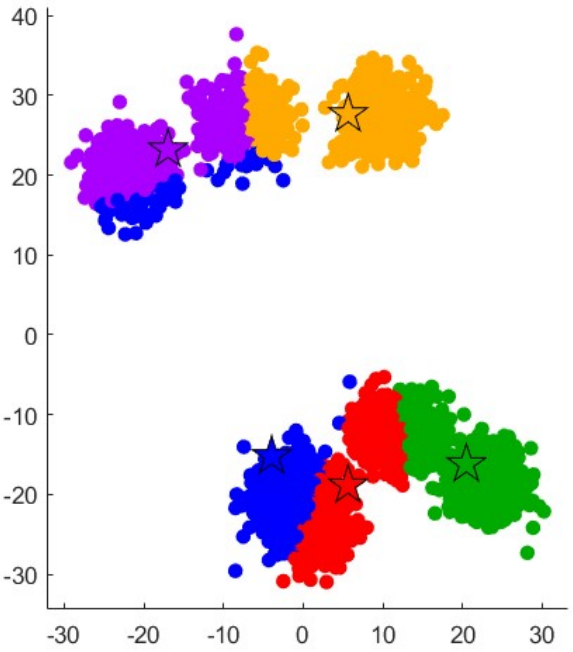


Figura 4.4: Iteração final para g = 5.

No final temos o resultado que a clusterização foi feita em 8 iterações sendo a primeira, a inicial, mostrada na Figura 4.1. Cada um dos cinco clusters contém 700 pontos após a iteração final. O próximo exemplo usaremos um dataset distinto e selecionamos $g = 4$.

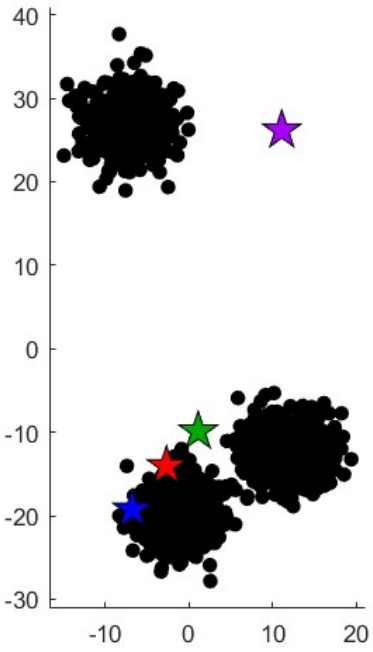


Figura 4.5: Gráfico inicial do dataset e escolha aleatória de clusters para g = 4.

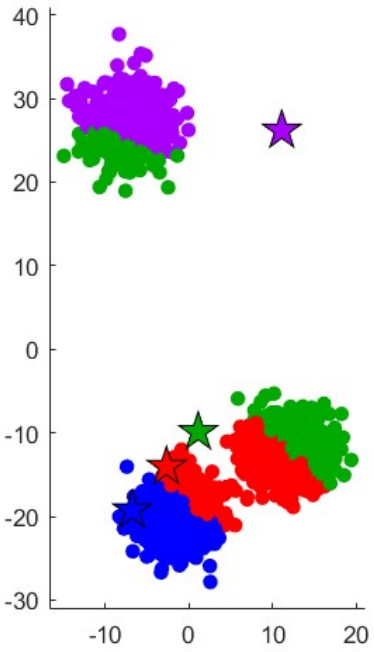


Figura 4.6: Iteração 1 para g = 4.

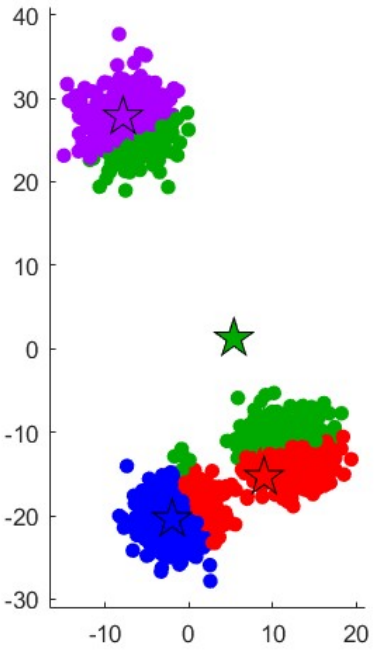


Figura 4.7: Iteração final para $g = 4$.

No final temos o resultado que a clusterização foi feita em 6 iterações sendo a primeira, a inicial, mostrada na Figura 4.7. Cada um dos quatro clusters contém 375 pontos após a iteração final.

4.2 Comparação com o K-Means

Esta Seção visa observar as diferenças nos resultados entre o algoritmo de clusterização com restrição de homogeneidade no tamanho dos grupos e o algoritmo K-Means. Será utilizada a mesma quantidade de grupos em ambos os exemplos para permitir uma comparação adequada. É importante ressaltar que o objetivo do algoritmo K-Means não é manter uma quantidade igual de pontos por cluster, ao contrário do algoritmo em questão.

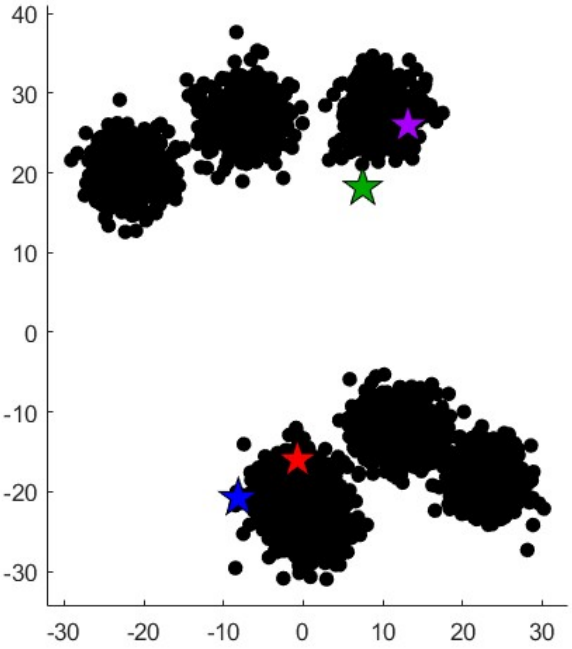


Figura 4.8: Gráfico inicial do dataset e escolha aleatória de clusters para $g = 4$.

(a) Iteração final para $g = 4$ do algoritmo com homegeneidade.

(b) Iteração final para $g = 4$ do algoritmo K-Means.

Figura 4.9: Resultado final da clusterização do algoritmo com homogeneidade e do K-Means

Ambos os exemplos foram realizados com um parâmetro g igual a 4. No entanto, o algoritmo de homogeneidade encerrou após 5 iterações, enquanto o algoritmo K-Means levou

6 iterações para finalizar. No algoritmo de homogeneidade, todos os clusters terminaram com 875 pontos, enquanto no K-Means a distribuição foi a seguinte: o primeiro cluster com 1009 pontos, o segundo com 991 pontos, o terceiro com 972 pontos e o quarto com 528 pontos. Agora faremos mais um exemplo de comparação entre os códigos dessa vez com $g = 2$.

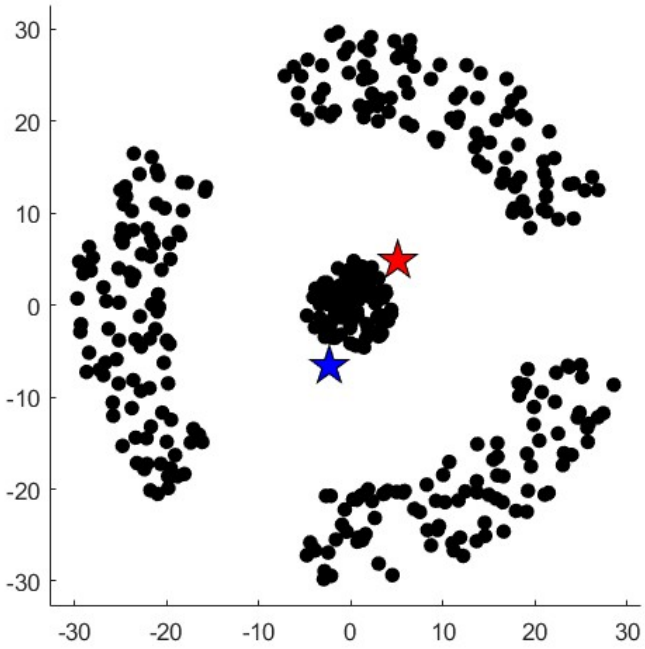
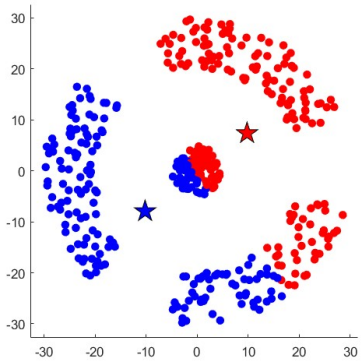
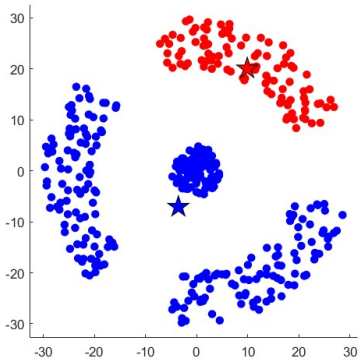


Figura 4.10: Gráfico inicial do dataset e escolha aleatória de clusters para $g = 2$.



(a) Iteração final para $g = 2$ do algoritmo com homogeneidade.



(b) Iteração final para $g = 2$ do algoritmo K-Means.

Figura 4.11: Resultado final da clusterização do algoritmo com homogeneidade e do K-Means.

Ambos os exemplos foram realizados com um parâmetro g igual a 2. No entanto, o algoritmo de homogeneidade encerrou após 9 iterações, enquanto o algoritmo K-Means levou 11 iterações para finalizar. No algoritmo de homogeneidade, todos os clusters terminaram com 200 pontos, enquanto no K-Means a distribuição foi a seguinte: o primeiro cluster com 300 pontos e o segundo com 100 pontos.

4.3 Conclusão

Neste capítulo, pudemos observar de forma explícita a diferença do algoritmo com homogeneidade na quantidade de pontos por grupo. Nos dois exemplos analisados, foi evidente que o algoritmo teve um desempenho melhor em termos de iterações necessárias. No exemplo 1, o algoritmo K-Means precisou de uma iteração adicional, enquanto no exemplo 2 foram necessárias duas iterações a mais em comparação ao algoritmo com homogeneidade. Isso demonstra que o algoritmo com homogeneidade apresenta uma performance superior, considerando a quantidade de iterações necessárias.

Capítulo 5

Conclusão e Trabalhos Futuros

5.1 Conclusão

O algoritmo estudado nesse projeto se mostrou muito promissor, na prática. Pudemos analisar as motivações para a criação do mesmo e alguns exemplos de possíveis uso. O comparativo com o algoritmo K-Means também evidenciou a eficácia e o diferencial do algorítimo proposto.

5.2 Trabalhos Futuros

Apesar dos resultados promissores obtidos neste projeto, surgiram algumas lacunas que podem ser exploradas em trabalhos futuros. Um dos pontos principais seria comparar o algoritmo proposto na Seção 3.3 com o algoritmo proposto por [5]. Seria interessante realizar uma pesquisa mais aprofundada em outros métodos de clusterização publicados que também visam a homogeneidade na quantidade de pontos por grupo, a fim de realizar uma comparação abrangente dos resultados. Além disso, um objetivo adicional seria concluir a tradução do código para Python, que já foi iniciada durante o processo de escrita deste trabalho.

Referências Bibliográficas

[1] e S. Star., G. C. B. *Sorting Things Out: Classification and its Consequences*. MIT Press, 1999.

[2] Farris., J. S. *Classification Among the Mathematicians*. Taylor Francis, Ltd., 1981.

[3] Jain., K. *50 years beyond K-means.* Pattern Recognition Letters, 2010.

[4] Kwasnik., B. *The role of classification in knowledge representation and discovery*. Library Trends, 1999.

[5] Pamplona, J. V. Uma abordagem para o problema de clusterização com restrição de capacidade. Master's thesis, Pós-Graduação em Métodos Numéricos Em Engenharia - Universidade Federal do Paraná, Curitiba - PR, Setembro 2022.