UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
MESTRADO EM INFORMÁTICA
DEPARTAMENTO DE INFORMÁTICA

IGOR DE MOURA VENTORIM

BIRCHSCAN: Um método de aproximação do DBSCAN para grandes conjuntos de dados

Vitória–ES

2021

IGOR DE MOURA VENTORIM

BIRCHSCAN: Um método de aproximação do DBSCAN para grandes conjuntos de dados

Versão Corrigida

Versão original encontra-se no PPGI/UFES.

Dissertação apresentada ao Mestrado em Informática (PPGI) da Universidade Federal do Espírito Santo (Ufes), como parte das exigências para a obtenção do título de Mestre em Informática.

Área de Concentração: Ciência da computação.

Orientador: Prof. Dr. Flávio Miguel Varejão
Coorientador: Dr. Diego Luchi

Vitória–ES

2021

Ficha catalográfica disponibilizada pelo Sistema Integrado de Bibliotecas - SIBI/UFES e elaborada pelo autor

---

Ventorim, Igor de Moura, 1995-
V466b BIRCHSCAN: Um método de aproximação do DBSCAN para grandes conjuntos de dados / Igor de Moura Ventorim. - 2021.
43 f. : il.

Orientador: Flávio Miguel Varejão.
Coorientador: Diego Luchi.
Dissertação (Mestrado em Informática) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Amostragem. 2. Análise de agrupamento. 3. Aprendizado não supervisionado. I. Varejão, Flávio Miguel. II. Luchi, Diego. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 004

Às vezes, pensamos que uma pessoa tem que fazer coisas brilhantes para mudar o mundo. Porém, na ciência é possível ver que, na verdade, são várias pessoas. O brilhantismo está em cada uma ao fazer pequenas contribuições e assim criarem um mundo melhor. Dedico este trabalho a todas as pessoas que, independente das adversidades, continuam buscando ser uma pessoa melhor.

Agradecimentos

Agradeço a Deus por tudo que me permitiu realizar até o presente momento.

Aos meus pais Amadeu, Maria Eunice(in memoriam) e ao meu irmão Iccaro, por tudo que fizeram por mim até hoje.

À minha esposa Ludmila, por sempre me incentivar, apoiar e estar ao meu lado em os todos momentos.

Ao meu professor orientador Flávio Miguel Varejão, por ter me aceitado como seu aluno e por ter me guiado ao longo deste trabalho.

Ao meu coorientador Diego Luchi, mesmo com toda a correria do dia a dia, sempre se disponibilizou a ajudar no que fosse possível para o andamento do trabalho.

Aos amigos e familiares, que estiveram comigo nesta jornada.

RESUMO

O algoritmo DBSCAN é um método clássico de agrupamento baseado em densidade. Este algoritmo permite identificar grupos de diferentes formatos, com a capacidade de gerenciar padrões ruidosos. O DBSCAN apresenta bons resultados, porém possui uma alta complexidade computacional em decorrência de diversos cálculos de distâncias no processo de agrupamento. Essa baixa eficiência computacional limita sua aplicação em grandes conjuntos de dados. Este trabalho apresenta um novo método de agrupamento cuja primeira etapa é a construção de elementos representativos para aplicar o DBSCAN a um conjunto reduzido de exemplos. O método proposto permite agrupar grandes conjuntos de dados com resultados próximos do resultado do DBSCAN em todo o conjunto de dados. A partir dos experimentos realizados, é observado que a técnica proposta apresenta bons resultados e consistência quando comparada a outros algoritmos com proposta semelhante.

Palavras-chave: Amostragem. análise de agrupamento. aprendizado não supervisionado.

ABSTRACT

The DBSCAN algorithm is a classic density-based clustering method. This algorithm allows to identify clusters of different shapes, with the ability to identify noisy patterns in the data. DBSCAN presents good results, however it has a high computational complexity due to several distance calculations in the clustering process. This low computational efficiency limits its application to large data sets. This work presents a new method of grouping whose first stage is the construction of representative elements to apply DBSCAN to a reduced set of examples The proposed method allows grouping large datasets with approximate results to the DBSCAN result applied in the entire dataset. From the experiments performed, it is observed that the proposed technique presents good results and consistency when compared to other algorithms with a similar proposal.

Keywords: Sampling. cluster analysis. unsupervised learning.

LISTA DE FIGURAS

Figura 1 – Fluxo dos procedimentos realizados para o desenvolvimento da dissertação 30
Figura 2 – Taxonomia de análise de agrupamento adaptado de (FAHAD et al., 2014) 36
Figura 3 – Exemplo do conceito diretamente alcançável por densidade. 38
Figura 4 – Visualização do conceito alcançável por densidade exibida na conexão dos pontos vermelhos. 39
Figura 5 – Resultado de execução do DBSCAN com raio ε e $minPTS = 4$ com a apresentação dos conceitos de ruído, diretamente alcançável, vizinhança ε, alcançável por densidade e grupo. 40
Figura 6 – Estrutura da CF-Tree do algoritmo BIRCH. Figura baseada na imagem apresentada em Wan e Yang (2013). 54
Figura 7 – Visão geral do BIRCH. Baseada na figura apresentada em ZHANG, RAMAKRISHNAN e LIVNY (1996). 55
Figura 8 – Visualização da distribuição dos centroides BIRCH 58

LISTA DE TABELAS

Tabela 1 – Conjuntos de dados usados nos experimentos 62
Tabela 2 – Tabela de contingência usada para computar o ARI 67
Tabela 3 – Resultados do Experimento 1. Número de execuções do DBSCAN por conjunto de dados, média e desvio padrão dos valores ARI em porcentagem e classificação dos algoritmos. Os maiores valores médios por conjunto de dados estão em negrito. 69
Tabela 4 – Teste Posthoc de Nemenyi 70
Tabela 5 – Tempos de execução em segundos do Experimento 1 70
Tabela 6 – Resultados do Experimento 2. Valores dos parâmetros, tempos de execução em segundos e ARI 72
Tabela 7 – Resultados do Experimento 3. Métodos aplicados em conjuntos de dados muito grandes. Valores dos parâmetros especificados e tempos de execução em segundos. 74

Lista de abreviaturas e siglas

DBS	Density Biased Sampling
BBS	Biased Box Sampling
FFT	Farthest First Traversal
DBSCAN	Density-based spatial clustering of applications with noise
BIRCH	Balanced iterative reducing and clustering using hierarchies
RI	Rand-Index
ARI	Adjusted Rand-Index

SUMÁRIO

1 INTRODUÇÃO 23
1.1 Motivação 23
1.2 Revisão de literatura 26
1.3 Objetivos 29
1.3.1 Objetivo geral 29
1.3.2 Objetivos específicos 30
1.4 Metodologia 30
1.5 Estrutura da dissertação 32

2 ALGORITMOS DE AGRUPAMENTO DE DADOS BASEADOS EM AMOSTRAGEM 33
2.1 Análise de agrupamento 33
2.1.1 Categorias dos métodos de análise de agrupamento 35
2.1.2 DBSCAN 37
2.2 Algoritmos de agrupamento baseados em amostragem 44
2.2.1 DENDIS 46
2.2.2 DIDES 48
2.2.3 ProTraS 49
2.2.4 Rough*-DBSCAN 51

3 A ABORDAGEM BIRCHSCAN 53
3.1 BIRCH 53
3.2 BIRCHSCAN 56

4 EXPERIMENTOS COMPUTACIONAIS 61
4.1 Metodologia Experimental 61
4.1.1 Bases de dados 61
4.1.2 Experimentos 62
4.1.3 Métrica de avaliação 66
4.2 Resultados 67
4.2.1 Experimento 1: Avaliação dos métodos de amostragem para diferentes valores dos parâmetros do DBSCAN 68
4.2.2 Experimento 2: Avaliação de qualidade e tempo em conjuntos de dados de grandes dimensões 71
4.2.3 Experimento 3: Avaliação de tempo em grandes conjuntos de dados . . 73

5 CONCLUSÃO 75
5.1 Contribuições 75
5.2 Trabalhos Futuros 77
Referências 79

1

Introdução

Com o grande avanço da tecnologia ao longo dos últimos anos, estamos passando por uma transformação digital que muitos chamam de quarta revolução industrial. Presenciamos uma era movida por dados, estes que estão cada vez mais volumosos, variados e valiosos. Consequentemente com mais dados sendo gerados, maiores tendem a se tornar os conjuntos de dados a serem trabalhados, no entanto, nosso poder computacional não acompanha o mesmo ritmo, sendo necessário técnicas, ferramentas e algoritmos de exploração e análise de dados mais eficientes e eficazes.

Entre diversas técnicas que trabalham com conjuntos de dados, que são atualmente relevantes e possuem aplicação em diferentes áreas, está a análise de agrupamento. Esta técnica é utilizada com sucesso e ainda é tema de pesquisas atuais, sendo aplicada em diferentes contextos, como avaliação da poluição do ar (GOVENDER; SIVAKUMAR, 2020), seleção de movimento do solo (ZHANG et al., 2020), avaliação de documentos em redes sociais (CURISKIS et al., 2020), segmentação de imagens (CARDOSO; BARROS; BARBOSA, 2020), entre outros.

1.1 Motivação

A análise de agrupamentos é uma tarefa comum de aprendizado de máquina que , dado um critério de similaridade, visa agrupar objetos semelhantes em um mesmo grupo. O algoritmo DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) proposto por ESTER (1996) é um algoritmo clássico e um dos métodos de agrupamento de maior sucesso na literatura. Seu funcionamento é baseado na densidade espacial que continua sendo o tema de pesquisas recentes (OHADI et al., 2020; CHEN et al., 2020; LI et al., 2020; LI, 2020; JANG; JIANG, 2019; CHEN YEWANG, 2018). O DBSCAN possui vantagens(apresentadas na seção 2.1.2), como por exemplo, detecção de ruídos e identificação de formatos arbitrários. Portanto, de acordo com suas características, o método aparece como uma proposta atraente para muitos problemas de agrupamento

quando comparado com algoritmos clássicos como Single-Linkage (SIBSON, 1973) e k-médias (MACQUEEN, 1967).

Apesar de ser um algoritmo de agrupamento bem-sucedido e aplicado em muitos domínios (SHERIDAN et al., 2020; FURQON; MUFLIKHAH, 2016; SHEN et al., 2016; NISA; ANDRIANTO; MARDHIYYAH, 2014), o DBSCAN executa uma operação cara chamada "consulta do vizinho mais próximo", que determina quantos elementos existem dentro de uma distância inferior a ε (Parâmetro DBSCAN) de um elemento específico. Cada vez que esta operação é acionada, o algoritmo realiza uma varredura completa do conjunto de dados, levando à alta complexidade computacional do algoritmo - $O(n^2)$ - e limitando a capacidade de lidar com grandes conjuntos de dados (JANG; JIANG, 2019; LI, 2020).

Existem duas formas principais para reduzir custo computacional do DBSCAN. A primeira se concentra em tornar a consulta do vizinho mais próximo mais eficiente (LI et al., 2020; LI, 2020; CHEN YEWANG, 2018; KUMAR; REDDY, 2016) ou aproximar os resultados da consulta (OTAIR et al., 2013). É importante notar que o tempo de execução de alguns métodos aumenta rapidamente conforme a dimensionalidade do conjunto de dados aumenta (CHÁVEZ et al., 2001; ARLIA; COPPOLA, 2001; KUO; SLOAN, 2005). A segunda abordagem utiliza técnicas de amostragem para encontrar uma solução aproximada a partir de um conjunto de informações para representar todo o conjunto de dados. Consequentemente, reduz o tempo de execução do DBSCAN. Alguns métodos de amostragem geram a amostra estatisticamente (WANG; HAMILTON, 2003a), enquanto outros selecionam com base nos resultados de um algoritmo de agrupamento com menor complexidade computacional (TANG; PI; HE, 2016; EL-SONBATY; ISMAIL; FAROUK, 2004a; VISWANATH P.; BABU, 2009).

Algumas técnicas, que possuem o propósito de tornar a consulta do vizinho mais próximo eficiente, nem sequer proporcionam um ganho de desempenho significativo ao se trabalhar com conjuntos de dados de alta dimensionalidade (CHÁVEZ et al., 2001; ARLIA; COPPOLA, 2001), no entanto, outras como a amostragem mostram-se bastante promissoras como pode ser visto em Al-mamory e Kamil (2016).

Uma estratégia semelhante a amostragem é a sumarização, esta consiste em criar um novo subconjunto(resumo) dos dados que represente as mais importantes informações do conteúdo original (SAINT-PAUL; RASCHIA; MOUADDIB, 2005).

Em ambas as técnicas, amostragem e sumarização, a quantidade de informação a ser processada pelo método e o esforço computacional diminuem. Em meio ao cenário de crescimento das bases de dados, essas técnicas mostram-se cada vez mais úteis, sendo que quanto mais consegue-se conter elementos de todos os grupos do conjunto de dados, melhores podem ser os resultados obtidos (APPEL, 2007; NGOAN; DUC et al., 2018; ZHAO; LIANG; DANG, 2019; CHEN et al., 2019). Em concordância com outros trabalhos

da literatura (ZHOU et al., 2000; NANOPOULOS; THEODORIDIS; MANOLOPOULOS, 2001; NANOPOULOS; MANOLOPOULOS; THEODORIDIS, 2002; WANG et al., 2011), o objeto de estudo dessa dissertação consiste em desenvolver um método que viabilize o uso do DBSCAN em grandes bases de dados. Para isto, é utilizado um algoritmo de análise de agrupamento de menor custo computacional com o objetivo de gerar um conjunto de dados de menor cardinalidade com elementos representativos. Este processo é semelhante a um processo de amostragem, no entanto, os elementos representativos não estão necessariamente presentes no conjunto de dados original. Este propósito implica em encontrar uma solução aproximada do DBSCAN em conjuntos de dados grandes com custo computacional reduzido. Nestes casos, um pequeno conjunto dos dados é definido para representar o conjunto de dados inteiro, reduzindo o tempo de execução do algoritmo.

A principal preocupação em relação ao processo de amostragem na análise de agrupamento está relacionada a possibilidade de um grupo não estar presente na amostra. A possibilidade de não conter representantes de todos os grupos na amostra é preocupante, pois impacta em perda de informação de um grupo pequeno em relação ao conjunto de dados. Esse risco é acentuado em conjuntos de dados que possuem grupos com cardinalidades discrepantes.

Para os casos em que se tem o conhecimento prévio dos grupos dos elementos, a realização do processo de amostragem, de modo a garantir a representatividade de cada grupo na amostra, é uma tarefa muito mais simples. Este tipo de caso é análogo a diversos problemas de classificação, no qual é comum a utilização da amostragem estratificada uniforme. Neste tipo de amostragem, sorteia-se igual número de elementos em cada estrato. Portanto, a amostra denota uma homogeneidade maior do que de toda a população, sendo assim, esse tipo de amostragem garante que contenha o mesmo número de representantes de cada grupo na amostra e que todos os grupos serão representados.

Na maioria dos casos, o uso da amostragem estratificada não é possível de ser realizado ao lidarmos com análise de agrupamento, isto ocorre devido ao fato de que, por definição, não há nenhuma informação referente aos grupos dos conjuntos de dados.

Diversos trabalhos na literatura estabelecem o uso da amostragem aleatória uniforme, afim de obter uma redução da quantidade de elementos do conjunto de dados (WANG; HAMILTON, 2003b; GUHA; RASTOGI; SHIM, 1998). É possível que haja uma tentativa de ajustar o tamanho da amostra com a intenção de que todos os grupos estejam representados, como feito em Bejarano et al. (2011). Porém, para esse tipo de amostragem, em casos de conjuntos de dados que contenham grupos com baixa cardinalidade, espera-se que o tamanho da amostra torne-se demasiadamente grande, para que exista uma alta probabilidade de todos os grupos estarem presentes na amostra.

Na amostragem aleatória uniforme, não existe um limite inferior para definir o tamanho da amostra, sendo difícil obter um conjunto reduzido do conjunto de dados que

contenha representantes de todos os grupos com uma margem de confiança alta (KOLLIOS et al., 2001). Além disso, devido não haver um conhecimento prévio sobre a cardinalidade de cada grupo, a definição do tamanho da amostra não é uma tarefa trivial.

Motivado pelos problemas mencionados e com a finalidade de buscar meios de solucioná-los, foram propostos diversos trabalhos sobre amostragem tendenciosa no contexto de análise de agrupamento. Esse tipo de amostragem tem como propósito incluir também os elementos de partições de baixa cardinalidade, garantindo que elementos de todas as partições estejam contidos na amostra.

Este trabalho segue a mesma abordagem realizada em LUCHI, RODRIGUES e VAREJÃO (2019), que apresenta uma nova técnica para reduzir a cardinalidade do conjunto de dados aplicando o algoritmo de agrupamento Leader* (LUCHI; RODRIGUES; VAREJÃO, 2019) para gerar uma amostra para rodar o DBSCAN. Esta proposta de trabalho substitui o algoritmo Leader* pelo algoritmo de agrupamento BIRCH (ZHANG; RAMAKRISHNAN; LIVNY, 1996) gerando um novo conjunto de dados a partir do resultado do BIRCH. O BIRCH é bastante eficiente ao trabalhar com grandes conjuntos de dados, ele explora a ideia de que o espaço de dados não está ocupado uniformemente e que uma região densa pode ser representada como um único elemento. Além disso, ao contrário do Leader*, o algoritmo BIRCH é bastante resiliente à ordem em que os dados são apresentados ao algoritmo.

1.2 Revisão de literatura

Nesta seção é apresentado um a breve revisão bibliográfica do tema amostragem tendenciosa para análise de agrupamento desde o surgimento na literatura até os dias atuais.

O primeiro trabalho encontrado na literatura especificamente sobre este tema foi proposto por Palmer e Faloutsos (2000) denominado como DBS (*Density Biased Sampling*). Este trabalho surge fazendo uma crítica sustentada até hoje em relação a amostragem aleatória uniforme para análise de agrupamento, a crítica diz respeito a este tipo de amostragem perder pequenos agrupamentos. O DBS é um método de amostragem tendenciosa por densidade que realiza a amostragem através de sorteio na qual a chance de um elemento compor a amostra é inversamente proporcional a densidade da região em que se encontra.

Após a publicação, o DBS obteve determinada atenção na área fazendo com que, posteriormente, surgissem outros trabalhos de amostragem tendenciosa voltados para análise de agrupamento (KOLLIOS et al., 2003; KERDPRASOP; KERDPRASOP; SATTAYATHAM, 2005; NANOPOULOS; THEODORIDIS; MANOLOPOULOS, 2006; APPEL et al., 2007a; APPEL et al., 2007b). Diversos trabalhos utilizavam a mesma

abordagem do DBS para compor a amostra, porém outras abordagens realizaram a seleção dos elementos com um critério baseado em: densidade, distância ou um híbrido de ambos.

Em amostragens tendenciosas por densidade, a probabilidade de um elemento ser selecionado para amostra depende da densidade local do conjunto de dados (KOLLIOS et al., 2003). Para os métodos que utilizam esse tipo de abordagem, por exemplo, em Palmer e Faloutsos (2000), estes necessitam de uma estratégia para estimar a densidade de modo que o custo computacional não o torne inviável. Para estes métodos, no decorrer dos anos, diferentes estratégias foram utilizadas para realizar a estimativa de densidade. Uma das estratégias de estimativa de densidade comum em diversos trabalhos é a utilização de indexadores espaciais sobre um reticulado do conjunto de dados.

Em Palmer e Faloutsos (2000) são utilizados índices espaciais em um reticulado sobre os dados. Para este método a indexação espacial é realizada através de uma estrutura de dados tabela *hash*. A tabela é construída a partir de uma função que realiza o mapeamento do conjunto de dados inicialmente representado em um reticulado no espaço para a ordem linear de uma estrutura de dados tabela *hash*. A tabela construída é composta com as densidades das células do reticulado. Neste caso, o reticulado é utilizado somente uma vez, no momento do mapeamento para a tabela *hash*.

A estratégia de estimativa de densidade adotada em Appel et al. (2007a), Appel et al. (2007b) ao propor o BBS(Biased Box Sampling) é a utilização de indexadores espaciais para dividir o conjunto de dados em células e armazenar em uma estrutura de dados de árvore. O algoritmo é composto por três etapas: a primeira corresponde a geração da árvore, a segunda reduz a profundidade da árvore nas regiões menos densas e a terceira etapa realiza a amostragem. A densidade de cada região é representada pelos nós folhas que estão em diferentes níveis de profundidade, sendo definidos a partir da avaliação de um teste de limiar. Os nós folhas são utilizados para compor a amostra.

Os autores de Kollios et al. (2003) utilizam uma técnica de estimativa de densidade baseada em estatísticas e, em particular, na teoria de kernel. Apesar desta técnica apresentar uma boa qualidade de estimativa, a mesma apresenta um custo computacional aceitável somente ao trabalhar com conjuntos de dados de baixa dimensionalidade. Além disso, necessita de uma configuração de parâmetros para ser utilizada. Porém, o método se mostrou promissor na identificação de ruídos.

Além das técnicas de estimativa de densidade apresentadas, existem diversas outras para conjuntos de dados multidimensionais, como: computação de histogramas multidimensionais (POOSALA; IOANNIDIS, 1997; CHAUDHURI; MOTWANI; NARASAYYA, 1999; KORN; JOHNSON; JAGADISH, 1999), utilização de transformadas (VITTER; WANG; IYER, 1998; MATIAS; VITTER; WANG, 1998) e estimadores de kernel (SCOTT, 2015; SILVERMAN, 1986), as quais, no entanto, não foram testadas em conjunto com métodos de análise de agrupamento.

Visto que a estimativa da densidade costuma ser custosa, em uma outra vertente, existem métodos que utilizam critérios de distância para a construção da amostra. Esta abordagem é relevante devido a distância também ser um critério importante na definição de agrupamentos, dado que essa métrica é utilizada para medir a similaridade e proximidade dos objetos. Para essas técnicas é comum encontrar a utilização de algoritmos como Leader (apresentado na seção 2.2) ou outro algoritmo que realize computação de distância para geração de amostras

Em Sarma e Viswanath (2009) é apresentado uma variação do algoritmo Leader que permite gerar uma amostra capaz de alcançar os mesmos resultados de agrupamento do método k-médias, porém, com um menor custo computacional. Neste caso, a variação do algoritmo Leader utiliza o critério de distância para a composição da amostra.

Um outro trabalho utilizando esta abordagem é apresentado em Dolnicar e Leisch (2004). Este trabalho se refere a segmentação de mercados utilizando agrupamentos e propõe um método que gera diversas amostras pequenas de modo aleatório uniforme, e então, um algoritmo de agrupamento baseado em distância é utilizado nessas amostras. Posteriormente, são extraídos os centros dos grupos retornados pelo algoritmo de agrupamento aplicado nas amostras, compondo um novo conjunto de dados reduzido que finalmente é aplicado em um algoritmo de agrupamento.

O algoritmo k-médias também pode ser considerado como um técnica de amostragem baseada em distância, o mesmo foi utilizado na geração de amostras para uma máquina de vetor de suporte em Xiao et al. (2014). No entanto, sua versão canônica pode não ser recomendada para trabalhar em conjunto com métodos de análise de agrupamento, devido os problemas de escalabilidade do próprio algoritmo como visto em Arthur e Vassilvitskii (2006).

Por fim, visando tirar proveito dos benefícios de ambas abordagens anteriores, temos os métodos híbridos, que utilizam critérios tanto de distância como de densidade para a seleção dos elementos que irão compor a amostra. Para os métodos que utilizam essa abordagem, é comum observar o propósito de calcular a densidade de um número menor de elementos devido ao alto custo computacional demandado por essa tarefa. Uma estratégia adotada é a criação de uma amostra utilizando critérios de distância e posteriormente o descarte dos elementos de regiões de baixa densidade, como feito no algoritmo DIDES (ROS FRÉDÉRIC; GUILLAUME, 2017) (descrito na seção 2.2.2).

Os métodos propostos por Yang e Wu (2005), Feldman e Langberg (2011) são bons exemplos de metodologias híbridas. Em Yang e Wu (2005) é proposto uma modificação do algoritmo *mountain method* adaptado para agrupamentos, o qual é apresentado como um método de agrupamento aproximado utilizando as métricas de distância e densidade. Na mesma direção em Feldman e Langberg (2011) é proposto um *framework* para construção de agrupamentos aproximados, que obteve amostragens significativamente menores e

correspondeu em tempos de execução melhores para conjuntos de dados da literatura.

Baseado no método de Feldman e Langberg (2011) surgiram os métodos DENDIS e DIDES. O DIDES (descrito na secão 2.2.2) demonstrou resultados satisfatórios, tendo apresentado desempenho de tempo de geração da amostra somente superior ao da amostragem aleatória uniforme, mas com um alto nível de qualidade quando comparado com os resultados do k-médias aplicado na amostra e no conjunto de dados (ROS FRÉDÉRIC; GUILLAUME, 2017). Em conjunto com o DIDES, os autores publicaram o DENDIS (ROS FRÉDÉRIC; GUILLAUME, 2016) (descrito na secão 2.2.1), que obteve resultados superiores ao DIDES. Por fim, os autores propuseram o ProTraS (ROS FRÉDÉRIC; GUILLAUME, 2018) (descrito na seção 2.2.3), que compõe a amostragem baseado em uma probabilidade definida a partir da distância e da densidade, o qual apresentou resultados ainda melhores que seus antecessores e demonstrou boa qualidade de aproximação de agrupamento quando comparado ao DBSCAN e k-médias.

Nos trabalhos apresentados é possível notar que a amostragem tendenciosa tornou-se uma etapa de pré-processamento para métodos de análise de agrupamento em geral. Porém, em VISWANATH P.; BABU (2009) foi desenvolvido um método para atuar especificamente em conjunto com o algoritmo de agrupamento DBSCAN, que apresentou resultados promissores. Em LUCHI, RODRIGUES e VAREJÃO (2019) é apresentada uma variante deste método chamada ROUGH*-DBSCAN (descrito na seção 2.2.4) a qual obteve resultados superiores ao do ROUGH-DBSCAN.

Nesse trabalho apresentamos o BIRCHSCAN (descrito na secão 3.2), um método que utiliza um algoritmo de análise de agrupamento de menor custo computacional para encontrar um novo conjunto de elementos representativos específicos para o DBSCAN, a fim de alcançar um resultado aproximado do próprio DBSCAN quando aplicado no conjunto de dados inteiro com menor custo computacional. Este método mostrou bom desempenho comparado aos métodos DIDES, DENDIS, ProTraS e ROUGH*-DBSCAN.

1.3 Objetivos

Uma das estratégias para a execução de métodos de análise de agrupamento em grandes conjuntos de dados é a utilização de métodos de amostragem combinados aos algoritmos de agrupamento. A seguir são descritos os objetivos desta dissertação.

1.3.1 Objetivo geral

Conduzindo-se na mesma direção de VISWANATH P.; BABU (2009) e LUCHI, RODRIGUES e VAREJÃO (2019), este trabalho tem como objetivo principal desenvolver um

método que permita encontrar um resultado aproximado de um algoritmo de análise de agrupamento específico, no caso o DBSCAN, com um custo computacional inferior para grandes conjuntos de dados.

1.3.2 Objetivos específicos

- Definir como gerar os elementos representantes, que serão aplicados ao DBSCAN.
- Obter a qualidade de aproximação e tempo de execução do método proposto comparado a outros métodos no estado da arte que possuem o mesmo propósito.
- Realizar a execução do método proposto em conjuntos de dados grandes em que o método de análise de agrupamento definido não consegue ser executado.

1.4 Metodologia

Nesta seção é descrita a metodologia utilizada nesta pesquisa, apresentando todas as etapas realizadas para elaboração deste trabalho.

Na Figura 1 podem ser vistos todos os procedimentos realizados para o desenvolvimento dessa dissertação.

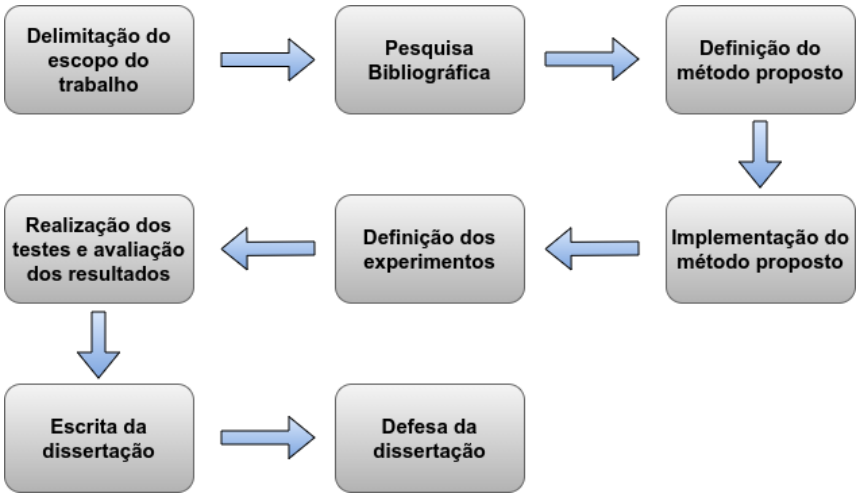

Figura 1 – Fluxo dos procedimentos realizados para o desenvolvimento da dissertação

O trabalho iniciou-se com a definição do tema a ser trabalhado, suas delimitações e escopo. Após a definição foi realizada uma revisão bibliográfica dos métodos de amostragem tendenciosa aplicados a análise de agrupamento.

No decorrer da revisão, foram descobertos métodos promissores que tiveram como intuito reduzir o alto custo computacional do DBSCAN quando aplicado em grandes

conjuntos de dados. Esse alto custo computacional é um problema atual. Nos trabalhos de VISWANATH P.; BABU (2009) e LUCHI, RODRIGUES e VAREJÃO (2019) foi possível observar que o uso de um algoritmo de agrupamento de menor custo computacional aplicado em grandes conjuntos de dados para gerar uma amostra, pode ser uma alternativa viável para resolver o problema.

O método foi definido inspirado em estudos realizados sobre o algoritmo de agrupamento BIRCH e nos trabalhos dos métodos de amostragem tendenciosa de VISWANATH P.; BABU (2009) e LUCHI, RODRIGUES e VAREJÃO (2019). A escolha do BIRCH foi devido a este na literatura ser apresentado como um algoritmo rápido ao se trabalhar com grandes conjuntos de dados, e que possibilita realizar uma boa sumarização dos dados. O objetivo do método proposto é permitir que se tenha um resultado aproximado do DBSCAN com ganho de desempenho quando comparado ao DBSCAN aplicado no conjunto de dados inteiro.

Após a definição do método proposto, foi realizada a etapa de implementação do método. Para a implementação foi utilizado a linguagem de programação Python, em conjunto com bibliotecas nativas da linguagem, disponíveis para se trabalhar com manipulação de dados e aprendizado de máquina. Ademais, foram realizados diversos testes empíricos e ajustes de parâmetros com a finalidade de atestar a viabilidade inicial do método.

A abordagem proposta foi comparada experimentalmente com os métodos mais recentes e destacados encontrados na revisão bibliográfica. Os seguintes algoritmos no estado da arte fizeram parte dos experimentos: DENDIS (ROS FRÉDÉRIC; GUILLAUME, 2017), DIDES (ROS FRÉDÉRIC; GUILLAUME, 2016), ProTraS (ROS FRÉDÉRIC; GUILLAUME, 2018) e ROUGH*-DBSCAN (LUCHI; RODRIGUES; VAREJÃO, 2019).

Três diferentes cenários de avaliação foram considerados. Para o primeiro experimento foram selecionados conjuntos de dados de baixa ou média cardinalidade e analisou-se o desempenho dos métodos comparando seus resultados com os resultados do DBSCAN aplicado em todo o conjunto de dados. O objetivo deste experimento foi comparar os resultados obtidos entre os métodos, realizando testes para diferentes valores de parâmetros do DBSCAN. Este experimento compara de modo abrangente o quão próximos os resultados dos algoritmos de amostragem correspondem aos resultados do DBSCAN.

Para os demais experimentos o teste de diversos valores de parâmetros do DBSCAN seriam muito custosos computacionalmente, pois são conjuntos de dados grandes e que consomem bastante tempo ao realizar o agrupamento. Deste modo, foram definidos apenas quatro configurações de valores de parâmetros para serem executadas, selecionadas de modo que os métodos não retornem todos os exemplos como ruídos e também visando obter diferentes agrupamentos em cada execução.

Para o segundo experimento foram selecionados conjuntos de dados com um número alto de elementos e dimensões, nos quais o DBSCAN ainda consegue ser executado. Este experimento mede a escalabilidade dos métodos comparando o tempo de execução do DBSCAN como uma linha de base. Deste modo, o objetivo deste experimento foi identificar a redução do tempo na execução dos métodos avaliados quando comparados ao DBSCAN, avaliando também em conjunto a qualidade dos resultados.

Para o terceiro, e último experimento, foi utilizado um conjunto de bases de dados nas quais não é viável executar o DBSCAN. O objetivo deste experimento foi realizar uma comparação temporal de execução entre os métodos mesmo não avaliando a qualidade dos agrupamentos gerados, e também verificar se os métodos conseguem executar em conjuntos de dados muito grandes.

Para avaliação dos métodos, além da mensuração de tempo, foi utilizada a métrica de avaliação *Adjusted Rand-Index* (descrita na seção 4.1.3) para avaliar a qualidade do resultado dos métodos comparado ao resultado do DBSCAN. A partir dos resultados da métrica de avaliação foram realizados os testes estatísticos de Friedman e post-hoc de Nemenyi.

1.5 Estrutura da dissertação

Este trabalho é organizado da seguinte forma. O Capítulo 2 faz uma discussão sobre métodos de análise de agrupamento e apresenta o método DBSCAN. Assim como, apresenta o conceito de algoritmos de agrupamento baseados em amostragem em conjunto com os algoritmos presentes na literatura, que foram selecionados para a etapa de experimentos. O Capítulo 3 apresenta o método BIRCHSCAN desenvolvido neste trabalho. No Capítulo 4, são apresentados os experimentos computacionais realizados e os resultados encontrados. Por fim, as conclusões e trabalhos futuros podem ser vistos no Capítulo 5.

2

Algoritmos de agrupamento de dados baseados em amostragem

Este capítulo apresenta uma breve revisão e categorização dos métodos de análise de agrupamento. O capítulo também apresenta o método DBSCAN que é o foco dos estudos realizados neste trabalho. Considerando que o método proposto aqui é inspirado em métodos de amostragem, o capítulo finaliza apresentando os algoritmos de agrupamento baseados em amostragem que se encontram no estado da arte e foram utilizados nos experimentos.

2.1 Análise de agrupamento

A análise de agrupamentos é o nome dado ao conjunto de técnicas idealizadas para reunir elementos em grupos formados a partir de uma métrica de similaridade ou dissimilaridade preestabelecida. Portanto, elementos que pertencem ao mesmo grupo possuem maior semelhança entre si, do que elementos que estão em grupos distintos (HUANG, 1997). O objetivo é se ter homogeneidade dentro do grupo e heterogeneidade entre grupos, em outras palavras, que as semelhanças dos elementos dentro de um grupo seja máxima enquanto a semelhança entre elementos de diferentes grupos seja mínima.

A técnica é bastante utilizada pois permite identificar objetos que possuem características em comum e separá-los, sendo abordada em muitos contextos por diversos pesquisadores devido a grande utilidade na análise exploratória de dados. Existem inúmeras aplicações práticas em diversas áreas de pesquisa, entre elas estão áreas como medicina, psicologia, biologia, sociologia, marketing, processamento de imagens entre outras (AHMAD; STARKEY, 2018; HUANG, 1997; VELMURUGAN, 2014). Todavia, é bastante comum observar seu estudo majoritariamente nos campos de mineração de dados, reconhecimento de padrões, suporte de decisão e aprendizado de máquina (YUNOH et al., 2017).

Diferentemente de uma tarefa de classificação, que é definida como aprendizado supervisionado, devido aos elementos já possuírem rótulos predefinidos, na análise de agrupamentos os elementos de entrada não possuem nenhum rótulo, sendo assim, este tipo de técnica é definida como uma tarefa de aprendizado não supervisionado (JAIN; MURTY; FLYNN, 1999; GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

Em Hruschka e Ebecken (2003) é apresentada uma definição formal a respeito da forma mais comum de análise de agrupamento. Um conjunto de n elementos $X = \{X_1, X_2, \ldots, X_n\}$, onde X_i^p é um vetor de dimensão p que pode ser agrupado em k grupos disjuntos $C = \{C_1, C_2, \ldots, C_k\}$ com as seguintes premissas:

$$C_1 \cup C_2 \cup \ldots \cup C_k = X \tag{2.1}$$

$$C_i \neq \{\} \tag{2.2}$$

$$C_i \cap C_j = \{\} \tag{2.3}$$

Baseado nestas definições observa-se que a união de todos os grupos compõe o conjunto de dados inteiro, cada grupo contém pelo menos um elemento e não é permitido que um elemento esteja presente em mais de um grupo. Segundo Hair et al. (2009) ainda se tem as seguintes questões fundamentais que surgem ao se trabalhar com análise de agrupamento:

1. Como medir a similaridade/dissimilaridade entre os objetos?
2. Como formar os agrupamentos?
3. Quantos grupos formar?
4. Como validar os agrupamentos formados?

A primeira questão é normalmente enfrentada baseada nas diversas medidas de proximidade existentes na literatura. No contexto de análise de agrupamento, é comum o uso de três medidas de proximidade, as medidas correlacionais, as medidas baseadas em distância e as medidas de associação (HAIR et al., 2009). A similaridade entre os objetos corresponde em quanto maior o valor observado mais parecidos são os objetos, por exemplo, um coeficiente de correlação. Por outro lado, a dissimilaridade dos objetos corresponde em quanto maior o valor observado mais diferentes são os objetos, por exemplo, distância euclidiana.

A segunda questão corresponde ao procedimento que deve ser utilizado para realizar os agrupamentos. Neste caso, o procedimento de agrupamento é baseado na técnica

utilizada pelo algoritmo escolhido. Na literatura existem diversos métodos de análise de agrupamento, em que cada um possui suas particularidades (XU; WUNSCH, 2005). Esses métodos podem ser categorizados baseado na estratégia adotada pelo algoritmo. Na próxima seção serão apresentados os tipos de categorias de técnicas de agrupamento.

Sabendo que a quantidade de métodos de análise de agrupamento presentes na literatura é extensa, e existem diferentes técnicas (FAHAD et al., 2014; GOLDSCHMIDT; PASSOS; BEZERRA, 2015), é necessário decidir qual técnica e método é mais adequado ao seu propósito, dado que diferentes métodos podem levar a resultados diferentes.

A terceira questão também é complexa , dado que em muitos problemas não se é conhecido o número de grupos a serem formados a priori. Na literatura existem heurísticas que possibilitam a escolha do número de grupos baseado na estrutura do agrupamento formado (HAIR et al., 2009), no entanto, é necessário destacar que existem diversos métodos que requerem essa informação como parâmetro de entrada. Além disso, também é uma prática comum a realização de testes empíricos para a identificação do número de grupos.

A quarta e última questão aborda a difícil tarefa de avaliar a qualidade do processo de agrupamento, principalmente pelo fato de ser uma técnica de aprendizado não supervisionado. Para isso, são necessárias medidas de qualidade que permitem avaliar os agrupamentos formados. Existem dois índices de validação de agrupamentos, os internos e os externos (DESGRAUPES, 2013). Os índices internos, também encontrados como índices relativos, analisam a estrutura de grupos descoberta em relação a alguma propriedade intrínseca dos dados, como por exemplo, compacidade ou separabilidade. Por outro lado, os índices externos, cujo foram utilizados neste trabalho, comparam a estrutura de grupos descoberta com uma estrutura previamente conhecida.

É importante destacar que existem duas premissas bastante relevantes a serem verificadas ao decidir pela utilização de uma técnica de análise de agrupamento (AAKER; KUMAR; DAY, 2001; ALBUQUERQUE, 2005; LINDEN, 2009). A primeira verifica se existe alguma métrica de similaridade ou dissimilaridade que possa ser utilizada no processo de análise de agrupamento com o objetivo de ser uma medida válida para identificação de semelhança entre os elementos. A segunda diz respeito a verificar se existe alguma justificativa teórica lógica para estruturar os elementos em grupos.

2.1.1 Categorias dos métodos de análise de agrupamento

Dependendo do autor, os métodos de análise de agrupamento podem ser categorizados de diferentes formas. Para Goldschmidt, Passos e Bezerra (2015) os métodos de análise

de agrupamentos podem ser basicamente separados em três grupos. Algoritmos baseados em distância, baseados em distribuições de probabilidades e baseados em densidade. No entanto, de acordo com Fahad et al. (2014), os agrupamentos também podem ser separados baseado na forma dos agrupamentos gerados, sendo assim, estes são classificados nas seguintes categorias: hierárquico, por particionamento, baseado em densidade, baseado em *grid* e baseado em modelo. Na imagem 2 temos a taxonomia com exemplos de algoritmos classificados em cada categoria.

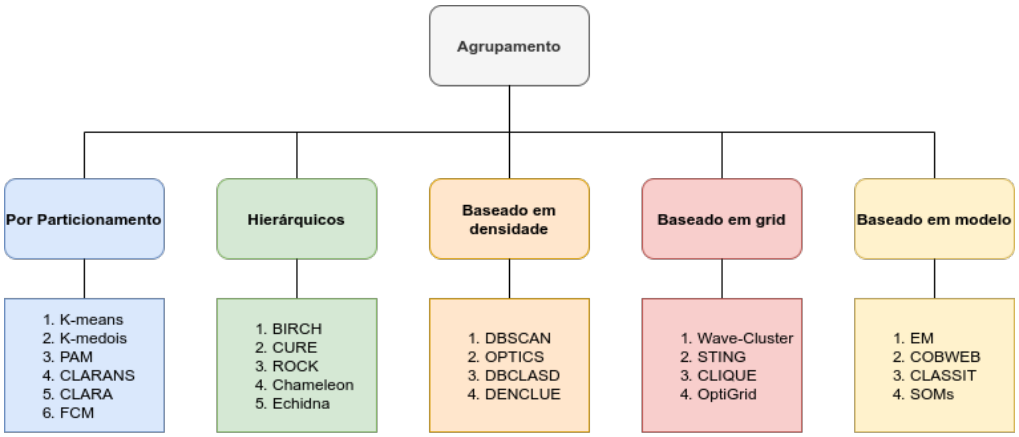

Figura 2 – Taxonomia de análise de agrupamento adaptado de (FAHAD et al., 2014)

Os métodos hierárquicos utilizam geralmente um processo recursivo, que constrói uma hierarquia de grupos de objetos no estilo de uma árvore. Sendo subdivididos em aglomerativos e divisivos (PACIFICO; LUDERMIR, 2014). Os métodos aglomerativos iniciam com cada objeto representando um grupo e no decorrer do algoritmo os grupos vão se unindo em grupos maiores. Nos métodos divisivos temos o contrário, o método inicia com todos elementos em um único grupo e no decorrer do processo são divididos em grupos menores. Uma prática comum de representar grupos formados por métodos hierárquicos é através de gráficos bidimensionais em forma de árvore conhecidos como dendrogramas. O método de análise de agrupamento BIRCH, utilizado neste trabalho, está presente nessa categoria.

Os métodos baseados em partição buscam de forma direta pela divisão do conjunto de dados em k grupos (GOLDSCHMIDT; PASSOS; BEZERRA, 2015). Estes métodos se iniciam definindo os centros dos k grupos, então os objetos do conjunto de dados são divididos entre os k grupos de acordo com que cada elemento seja atribuído ao centro do grupo mais próximo, baseado em uma métrica preestabelecida. O método mais conhecido desta categoria é o método das k-médias(k-means).

Os métodos baseados em densidade definem grupos observando a densidade de suas regiões, os grupos são formados a partir de regiões densas e ruídos se encontram em regiões de baixa densidade. Estes métodos possuem um formato particular de crescimento, incluindo somente objetos densos nos grupos. Um elemento é considerado denso baseado

nos elementos de sua vizinhança ou em alguma função de densidade. O método de análise de agrupamento DBSCAN está presente nesta categoria e será apresentado com mais detalhes na próxima seção.

Os métodos baseados em *grid* realizam uma discretização do espaço em diversas células. O objetivo é realizar cálculos de agrupamento em um espaço reduzido. O benefício destes métodos está na redução do tempo de execução, dado que os cálculos não são aplicados no conjunto de dados inteiro, mas sim em cada célula. As operações de agrupamento dependem somente da quantidade de células em cada dimensão do espaço discretizado.

Os métodos baseados em modelo buscam ajustar algum modelo matemático aos dados criando um modelo para cada um dos grupos e encontrando o melhor ajuste dos dados ao modelo. Métodos baseados em modelos matemáticos podem realizar agrupamentos a partir de uma função que seja capaz de representar a distribuição espacial dos elementos no conjunto de dados. Estes modelos supõem que os dados são gerados baseados em um mistura de distribuições de probabilidades (HAN; KAMBER; PEI, 2001).

2.1.2 DBSCAN

Esta seção apresenta o algoritmo DBSCAN (*Density Based Spatial Clustering of Application with Noise*) proposto por ESTER (1996). Este é um algoritmo de grande relevância na área de análise de agrupamento e corresponde a um dos métodos mais bem sucedidos da área, o qual é objeto de estudo desde sua publicação original até os dia atuais (OHADI et al., 2020; CHEN et al., 2020; LI et al., 2020; LI, 2020; JANG; JIANG, 2019; CHEN YEWANG, 2018).

O algoritmo foi desenvolvido para descobrir grupos e ruídos de um conjunto de dados baseado em sua cobertura espacial. O DBSCAN utiliza a densidade para a realização dos agrupamentos, isto possibilita que o algoritmo seja capaz de realizar agrupamentos de formato arbitrário. O método permite agrupamentos de alta qualidade e possui diversas vantagens quando comparado com outros algoritmos de análise de agrupamento.

Na publicação original do DBSCAN em ESTER (1996), os autores definem alguns conceitos para que seja possível uma maior compreensão do algoritmo. Na primeira definição temos o conceito de vizinhança ε de um elemento, que é definida por $N_\varepsilon(p) = \{q \in D \mid dist(p,q) \leq \varepsilon\}$. Neste caso ε representa a distância limite que um elemento q deve estar do elemento p para que seja considerado um vizinho. Deste modo, podemos dizer que ε representa o raio da vizinhança de um determinado elemento e $N_\varepsilon(p)$ simboliza a função que obtém todos os vizinhos do elemento p.

Como segunda definição tem-se o conceito de diretamente alcançável por densidade

definindo que um elemento p é diretamente alcançável por densidade de um elemento q se:

1. $p \in N_\varepsilon(q)$

2. $|N_\varepsilon(q)| \geq minPTS$ (Condição para elemento de núcleo)

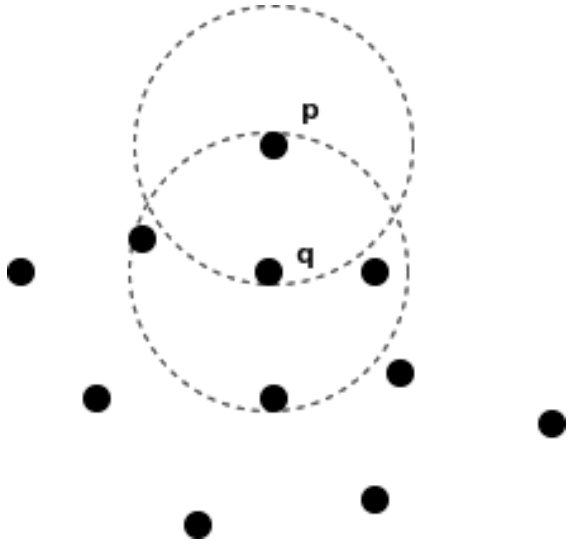

Figura 3 – Exemplo do conceito diretamente alcançável por densidade.

- p: elemento de fronteira
- q: elemento de núcleo
- p é diretamente alcançável por densidade de q
- q não é diretamente alcançável por densidade de p.

3. Existem dois tipos de elementos presentes em um grupo: os elementos de núcleo e os elementos de fronteira. Em uma Vizinhança ε de um elemento de núcleo, baseado em sua definição, existem no mínimo minPTS elementos. No entanto, isso não é verdade para elementos de fronteira. Deste modo, um elemento de fronteira é diretamente alcançável a partir de um elemento de núcleo, porém o inverso não é verdadeiro. Na Figura 3 podemos ver o elemento q, que é um elemento de núcleo por conter no mínimo 3 elementos em sua vizinhança, e q é diretamente alcançável de p. No entanto, q não contém o mínimo de 3 elementos em sua vizinhança, por isto, é um elemento de fronteira, deste modo, p não é diretamente alcançável a partir do elemento q.

Em um elemento de fronteira tem-se significativamente menos elementos de vizinhança ε do que na vizinhança de um elemento de núcleo. Pares de elementos de núcleo possuem simetria no critério de diretamente alcançável por densidade, ou seja, se p é um elemento de núcleo e q também, e se p é diretamente alcançável por densidade de q, o inverso também é válido.

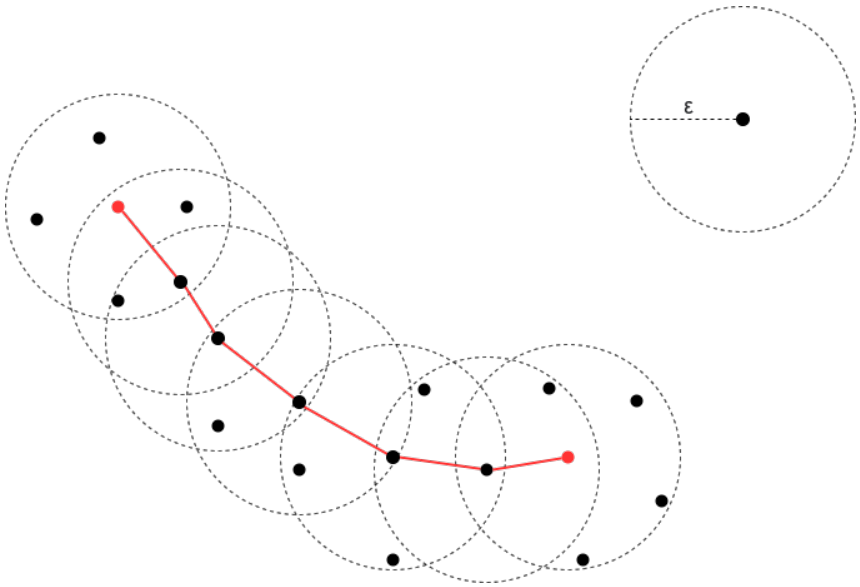

Figura 4 – Visualização do conceito alcançável por densidade exibida na conexão dos pontos vermelhos.

O terceiro conceito a ser apresentado utiliza o conceito anterior em sua definição, dado que é uma extensão de diretamente alcançável por densidade, sendo denominado alcançável por densidade e definido como: Um elemento p é alcançável por densidade de um elemento q, se existir uma cadeia de elementos $p_1, p_2, \ldots, p_n$, dado que $p_1 = p$ e $p_n = q$ tal que $p_{(i+1)}$ é diretamente alcançável por densidade de p_i.

Assim, como no conceito de diretamente alcançável por densidade, o conceito alcançável por densidade também é simétrico para dois elementos de núcleo. Na Figura 4 pode-se visualizar dois elementos de núcleo (pontos vermelhos) diretamente alcançáveis por densidade. No entanto, quando se lida com dois pontos de fronteira que pertencem ao mesmo grupo, provavelmente estes não serão alcançáveis por densidade entre si, dado que isso somente será verdade se ambos satisfazerem a condição de elemento de núcleo.

Na quarta definição se tem o conceito de conectado por densidade. Um elemento p é conectado por densidade a um elemento q, se existe um elemento em que tanto p e q são alcançáveis por densidade. Ao se considerar dois elementos de fronteiras em um grupo, existe pelo menos um elemento de núcleo que ambos os elementos de fronteira são alcançáveis por densidade.

Após as definições anteriores é possível fazer a quinta definição, que apresenta o conceito grupo baseado em densidade. Se existe um conjunto de dados D, um grupo C é um subconjunto não vazio de D se satisfazer as seguintes propriedades:

1. $\forall p, q$: Se $p \in C$ e q é um elemento alcançável por densidade de p, então $q \in C$. (Maximalidade)

2. $\forall p, q \in C$: p é conectado por densidade a q. (Conectividade)

A definição do conceito de grupo baseado em densidade permite que o algoritmo particione o conjunto de dados em grupos e encontre os membros pertencentes a cada grupo, onde todos os elementos do mesmo grupo estão conectados por densidade.

A característica de identificação de ruídos é um recurso muito importante do DBSCAN e que não é encontrada na maioria dos outros métodos de análise de agrupamento. Seja $C_1, \ldots, C_k$ grupos do conjunto de dados D tal que $i = 1, 2, \ldots, k$. Define-se um elemento p como ruído, se p pertence ao conjunto de dados D e não pertence a qualquer grupo C_i. Ou seja, $Ruído = \{p \in D \mid \forall i : p \notin C_i\}$. É comum que padrões ruidosos estejam em regiões de baixa densidade não permitindo que sejam conectados por densidade a nenhum elemento e consequentemente não sendo inclusos a nenhum grupo, sendo este um mecanismo eficaz. Entretanto, pode ocorrer de um elemento ruidoso estar em uma região densa, deste modo, o elemento será rotulado como um membro de grupo.

Na imagem 5 é possível visualizar o resultado de uma execução realizada pelo DBSCAN com a exibição de alguns dos conceitos apresentados. Há dois grupos baseados em densidade (grupo de pontos azuis e grupo de pontos verdes), dois elementos rotulados como ruído (pontos vermelhos), círculo com raio ε dos elementos de núcleo e ruídos. Elementos de núcleo conectados por densidade (linha vermelha). A partir da figura também é possível visualizar que, todos os elementos verdes são conectados por densidade, ou seja, para todos os pares de elementos verdes, existe um elemento que é alcançável por densidade para ambos os elementos do par, e o mesmo é válido para os elementos azuis.

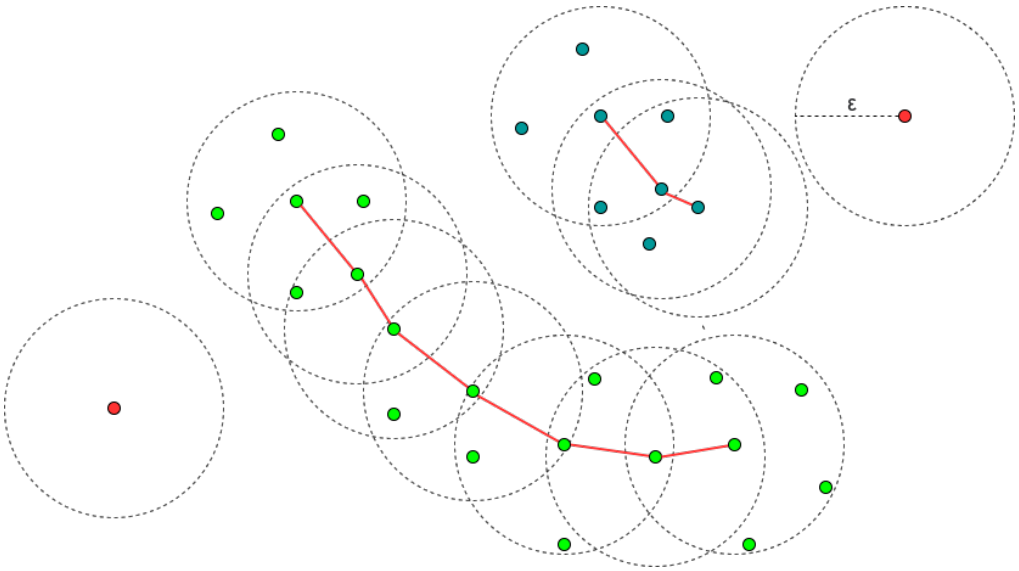

Figura 5 – Resultado de execução do DBSCAN com raio ε e $minPTS = 4$ com a apresentação dos conceitos de ruído, diretamente alcançável, vizinhança ε, alcançável por densidade e grupo.

O algoritmo 1 exibe o pseudocódigo do DBSCAN tal como apresentado em Schubert et al. (2017). O método recebe como entrada o conjunto de dados que se deseja realizar o agrupamento, o parâmetro ε, o parâmetro $minPTS$ e a função de distância, sendo a

função mais comum utilizada a distância euclidiana. O algoritmo inicia com um laço principal que itera sobre todos os elementos do conjunto de dados (linha 1) e verifica se o grupo do elemento é diferente de indefinido, caso seja, itera para o outro elemento (linha 2). Inicialmente todos os elementos possuem grupos indefinidos. Caso a primeira verificação seja falsa, isso corresponde que o elemento p é indefinido, portanto, é executado a função $N_\varepsilon(p)$ para encontrar a vizinhança ε de p (linha 4), em seguida é verificada a cardinalidade da vizinhança ε (linha 5), caso seja menor que $minPTS$ o elemento é provisoriamente rotulado como ruído e se prossegue para o próximo elemento (linhas 6 e 7).

Algorithm 1: DBSCAN

Input : D; // Conjunto de dados a ser agrupado
Input : ε; // Raio para vizinhança epsilon
Input : $minPTS$; // Limite de densidade:
Input : $dist$; // Função de distância
Output : $Clusters$// Grupos do conjunto de dados

```
1  foreach elemento p in D do
2      if cluster(p) ≠ indefinido then
3          continue;
4      Vizinhança N ← Vizinhança_ε(D, dist,p, ε);
5      if |N|< minPTS then
6          cluster(p) ← Noise;
7          continue;
8      c ← próximo rótulo de grupo;
9      cluster(p) ← c;
10     S ← N \{p};
11     foreach q in S do
12         if cluster(q)= Noise then
13             cluster(q) ← c;
14         if cluster(q) ≠ indefinido then
15             continue;
16         Vizinhança N ← Vizinhança_ε(D, dist,q, ε);
17         cluster(q) ← c;
18         if |N|< minPTS then
19             continue;
20         S ← S ∪ N ;
21     end
22 end
```

Dado que a condição de elemento de núcleo tenha sido satisfeita, então um novo grupo é gerado e p passa a pertencer a esse grupo (linhas 8 e 9). S recebe todos os elementos que são vizinhos de p (linha 10), após isso, um laço é executado para iterar por todos os elementos presentes em S (linha 11). Para cada elemento q presente em S

é verificado se o elemento havia sido rotulado provisoriamente como ruído, caso tenha sido, ele recebe o mesmo grupo de p (linha 13). Em seguida, é verificado se o grupo de q é diferente de indefinido, caso seja, se itera para o próximo elemento (linha 14). Caso q seja indefinido então, é utilizada a função $N_\varepsilon(q)$ para encontrar a vizinhança ε de q (linha 16). Após isso, q recebe o mesmo grupo de p (linha 17) e é analisado se q não é um elemento de núcleo. Caso q não seja um elemento de núcleo então, se itera para o próximo elemento (linha 19). Caso q seja um elemento de núcleo, então atribui-se a S a vizinhança ε encontrada a partir de q (linha 20).

Inicialmente o laço (linhas 11 – 21), que itera por S, é formado somente pela vizinhança ε de p, no entanto, todas as vezes que um elemento q, que ainda não foi atribuído nenhum grupo, for adicionado ao grupo de p, e q for um elemento de núcleo, a sua vizinhança ε também será atribuída em S. Nesta etapa, inicializa-se com os elementos diretamente alcançáveis por densidade de p e posteriormente são encontrados todos os elementos alcançáveis por densidade a partir de p.

Após finalizar o laço interno, o processo volta para o laço principal até que seja feito a iteração por todos os elementos do conjunto de dados. Todos os elementos que a partir da execução foram definidos provisoriamente como ruídos, e não houve mudanças em seu grupo no decorrer do algoritmo, mantêm-se como ruídos. O resultado final do algoritmo retorna o conjunto de dados com os grupos baseados em densidade e os elementos rotulados como ruído.

Pode-se identificar que o DBSCAN é um método determinístico, ou seja, o resultado do mesmo não irá variar entre diferentes execuções. No entanto, é valido destacar que o algoritmo pode ser sensível a ordem que os dados são apresentados ao mesmo, dado que se tivermos um elemento de fronteira entre dois grupos baseados em densidade, este elemento pode ser rotulado como qualquer um dos grupos, ficando vulnerável a ordem que o conjunto de dados será iterado.

O algoritmo DBSCAN possui diversas vantagens quando comparado a outros algoritmos de análise de agrupamento. Nestas vantagens, está inclusa a identificação de formatos arbitrários de grupos, sendo assim, não é necessário assumir que a partição possui um formato específico. Outra vantagem é a não necessidade de informar como parâmetro o número de grupos no momento da execução, e por fim, como apresentado anteriormente, a capacidade de identificar e separar os ruídos (ESTER, 1996).

O método possui como desvantagem a necessidade de ajuste de dois parâmetros de execução. Eles não são de estimativa trivial, principalmente quando se é necessário trabalhar com conjuntos de dados de alta dimensionalidade.

Caso seja informado um valor de $minPTS$ e ε muito pequeno o algoritmo tende a formar um número grande de grupos e em caso de $minPTS$ igual a 1, é possível que

o número de grupos seja equivalente a cardinalidade do conjunto de dados, não fazendo sentido executar um algoritmo de análise de agrupamento para definir cada elemento como um grupo. Outro problema que pode ocorrer na definição dos parâmetros é o caso do valor de ε ser muito grande fazendo com que seja gerado apenas um único grupo, devido todos os elementos do conjunto de dados serem alcançáveis por densidade a partir de um único elemento, ou muito pequeno de modo a não permitir que os elementos sejam alcançáveis por densidade e consequentemente rotular todos os elementos como ruído.

Apesar dos casos específicos apresentados, é possível notar que a estimativa incorreta dos parâmetros pode causar diversos problemas, como por exemplo, agrupar dois grupos diferentes em um único, ou até mesmo separar um grupo com elementos coerentes em subgrupos. Portanto, para encontrar resultados de qualidade é necessário que se tenha cuidado ao definir os parâmetros. Pensando nisso, diversos trabalhos foram propostos com uma forma de seleção automática de parâmetros (ZHOU; WANG; LI, 2012), ou realizando estudos para definir formas de encontrar os melhores parâmetros (RAHMAH; SITANGGANG, 2016), ou também propondo versões que não necessitam de parâmetros (CHEN YEWANG, 2018).

Embora o DBSCAN seja um algoritmo bastante utilizado devido as qualidades apresentadas, existem diversas modificações propostas com o objetivo de acelerar a execução do método. Isto ocorre devido o algoritmo possuir um alto custo computacional ao realizar a busca dos vizinhos. Em suma, pode-se dizer que a busca consiste em calcular a distância de todos os elementos até o elemento de interesse e retornar os que possuem distância inferior a ε. O alto custo computacional é observado principalmente ao se trabalhar com conjuntos de dados de alta cardinalidade ou dimensionalidade, dado que o algoritmo possui complexidade de $O(n^2)$. A maioria das propostas de modificações são focadas em tentar reduzir este alto custo computacional que está correlacionado a busca dos vizinhos.

Diversas foram as estratégias propostas focadas em acelerar a busca da vizinhança, entre elas uma bastante comum é a do uso de estruturas de dados para a indexação espacial dos elementos do conjunto de dados com o objetivo de otimizar a consulta dos elementos vizinhos, como por exemplo em Patwary et al. (2012). Baseados nessa estratégia, diversos trabalhos na literatura foram propostos utilizando diferentes estruturas de dados. No entanto, de modo geral, apesar de muitas propostas obterem desempenhos satisfatórios em conjuntos de dados com um baixo número de dimensões, essa vantagem obtida tende a desaparecer ao se trabalhar com conjuntos de alta dimensionalidade (CHÁVEZ et al., 2001).

Por outro lado, existem trabalhos que exploram variações do DBSCAN utilizando técnicas de amostragem tendenciosa. Para estes trabalhos, em vez de aplicar o método no conjunto de dados inteiro, o foco é aplicá-lo somente em um subconjunto reduzido do conjunto de dados e ter como resultado somente este subconjunto rotulado. Também

existem variantes do algoritmo que utilizam abordagens híbridas – foco desta dissertação - nas quais utiliza-se um algoritmo de análise de agrupamento de menor complexidade que é executado em uma etapa anterior a aplicação do DBSCAN.

Em El-Sonbaty, Ismail e Farouk (2004b) temos um exemplo de abordagem híbrida, onde os autores inicialmente utilizam o algoritmo CLARANS (NG; HAN, 2002) para fazer o agrupamento no conjunto de dados inteiro, posteriormente aplicam o DBSCAN em conjuntos menores retornados na primeira etapa e fazem a união dos resultados das diferentes execuções, obtendo o conjunto de dados. Esta estratégia é realizada devido ser muito menos custoso a aplicação do DBSCAN em vários conjuntos de dados menores do que a execução no conjunto completo. Isto ocorre devido a complexidade quadrática do algoritmo.

Outro trabalho que propõe a utilização de abordagem híbrida pode ser visto em VISWANATH P.; BABU (2009). Neste trabalho os autores utilizam o algoritmo de análise da agrupamento Leader que realiza uma única varredura no conjunto de dados para compor uma amostra que posteriormente é executada no DBSCAN. Ao final do algoritmo todos os elementos que não foram selecionados para compor a amostra são inseridos novamente e rotulados com o mesmo grupo de seu líder.

É válido destacar que ao executar o DBSCAN apenas em uma amostra do conjunto de dados e no conjunto de dados completo, existem diferenças além do tempo de execução. Algo que vale ser observado com atenção está relacionado aos parâmetros do algoritmo. Para isto, é necessário compreender que, existem parâmetros que são utilizados no conjunto de dados completo que podem ser inviáveis ao serem aplicados na amostra, fazendo com que diversos elementos sejam rotulados como ruído. Isto pode ocorrer, dado que a densidade de elementos da amostra tende a ser muito menor que a do conjunto de dados inteiro. Portanto, é importante ressaltar que ao se trabalhar com amostras a estimativa dos parâmetros requer ainda mais atenção.

Além dos trabalhos mencionados, foram propostos diversos outros métodos de amostragem tendenciosa para a análise de agrupamento, que utilizam especificamente o DBSCAN. Os métodos no estado da arte selecionados para os experimentos deste trabalho serão apresentados na próxima seção.

2.2 Algoritmos de agrupamento baseados em amostragem

A técnica de amostragem aleatória é a abordagem mais popular e intuitiva para compreender uma amostra de elementos, no entanto, dois problemas principais surgem ao usar este

método para selecionar amostras que serão aplicadas em algoritmos de agrupamento. Os problemas estão relacionados à cardinalidade e à densidade do agrupamento, respectivamente. A primeira, e mais importante questão é, que os grupos com baixas cardinalidades podem facilmente ser perdidos na amostra, uma vez que a abordagem aleatória pressupõe que cada elemento do conjunto de dados tem uma probabilidade igual de ser escolhido. A segunda questão surge pelo fato de que regiões densas ficarão super-representadas, enquanto o oposto ocorre em regiões esparsas. O fato de que o conhecimento prévio sobre a cardinalidade e densidade dos grupos normalmente não estão disponíveis aumenta os problemas associados à amostragem aleatória (KOLLIOS et al., 2003).

As técnicas de amostragem tendenciosa são muito atraentes, porque visam resolver ambos os problemas. Elas mantêm uma boa quantidade de amostras em grupos esparsos, enquanto removem amostras desnecessárias em regiões que já estão bem representadas (PALMER; FALOUTSOS, 2000). Os algoritmos de amostragem tendenciosa podem ser divididos em duas categorias principais: métodos baseados em distância e baseados em densidade (a combinação dos dois conceitos também pode ser adotada).

Os algoritmos Farthest-First Traversal (FFT) (ROSENKRANTZ; STEARNS; LEWIS II, 1977) e Leader (HARTIGAN, 1975) são duas abordagens amplamente utilizadas por métodos de amostragem tendenciosa. Em suas versões canônicas, eles usam a métrica de distância entre elementos em sua execução. No entanto, eles também são usados para cálculo de densidade em vários métodos de amostragem, que combinam conceitos baseados em distância e baseados em densidade (ROS FRÉDÉRIC; GUILLAUME, 2016; ROS FRÉDÉRIC; GUILLAUME, 2017; ROS FRÉDÉRIC; GUILLAUME, 2018; VISWANATH P.; BABU, 2009; LUCHI; RODRIGUES; VAREJÃO, 2019).

O algoritmo FFT, proposto principalmente como heurística para o problema do caixeiro viajante (ROSENKRANTZ; STEARNS; LEWIS II, 1977), é um algoritmo iterativo que adiciona novos dados à amostra a cada iteração até atingir um critério de parada. Em sua versão canônica, o elemento mais distante dos já selecionados é o elemento a ser incluído na amostra, e o critério de parada é o tamanho da amostra.

O princípio FFT é amplamente utilizado entre as abordagens de amostragem com base na distância, uma vez que não requer nenhum parâmetro de distância. O algoritmo FFT é inicializado selecionando o primeiro elemento, que é escolhido aleatoriamente ou calculado como o elemento mais distante no conjunto de dados. Após a etapa de inicialização, o algoritmo executa duas etapas até atingir o tamanho de amostra necessário. Primeiro, itera sobre cada elemento não pertencente à amostra e associa, a cada um deles, o elemento mais próximo presente na amostra. Então, na segunda etapa, o algoritmo seleciona o item com a maior distância associada e o anexa à amostra. Observe que a escolha do primeiro elemento define se FFT é um algoritmo estocástico (escolha aleatória) ou determinístico (escolha calculada).

Outro algoritmo popular usado para realizar a amostragem tendenciosa é o Leader (HARTIGAN, 1975), um algoritmo de agrupamento incremental que realiza uma única varredura no conjunto de dados. O algoritmo começa com o primeiro elemento como líder e, sequencialmente, itera sobre todos os outros elementos do conjunto de dados. Se um elemento está a uma distância menor do que τ de qualquer líder existente, ele se torna um seguidor, caso contrário, ele se torna um líder. Após uma única varredura em todo o conjunto de dados, o conjunto de líderes, que estão pelo menos a uma distância τ separados um do outro, pertencem a amostra. Observe que, ao contrário do FFT, o Leader requer o parâmetro de distância τ. Por outro lado, o tamanho da amostra não precisa ser definido.

Três algoritmos de agrupamento baseados em FFT foram propostos recentemente: DENDIS (ROS FRÉDÉRIC; GUILLAUME, 2016), DIDES (ROS FRÉDÉRIC; GUILLAUME, 2017) e ProTraS (ROS FRÉDÉRIC; GUILLAUME, 2018). Esses algoritmos usam distância, densidade ou uma combinação de ambas as características para gerar as amostras. Outro trabalho recente (LUCHI; RODRIGUES; VAREJÃO, 2019) propôs uma abordagem baseada no algoritmo Leader para obter uma versão melhorada do algoritmo Rough-DBSCAN (VISWANATH P.; BABU, 2009). Os autores chamaram essa proposta de Rough*-DBSCAN. Nas seções a seguir, esses algoritmos de amostragem tendenciosa são descritos resumidamente.

2.2.1 DENDIS

A sigla DENDIS é um acrônimo de DENsidade e DIStância. O método é uma abordagem híbrida proposta em ROS FRÉDÉRIC; GUILLAUME (2016) que combina densidade e distância. Ele possui duas etapas para a construção da amostra e utiliza a densidade como critério de parada. Seu objetivo é construir uma amostra que seja a menor possível e disponha do mesmo padrão de distribuição do conjunto de dados. Os elementos presentes na amostra são considerados representantes, tal que cada elemento é associado ao representante mais próximo. No final do algoritmo é esperado que a amostra S seja a menor possível e possua o mesmo padrão de distribuição do conjunto de dados D.

Baseado no algoritmo FFT, o DENDIS também é um algoritmo iterativo, entretanto, com dois propósitos ao adicionar um novo representante em cada iteração. O primeiro consiste em garantir que as áreas de alta densidade sejam representadas na amostra, com a menor quantidade de elementos possíveis visando escapar de representações sobrepostas. O segundo compreende em cobrir o espaço de modo homogêneo. Para satisfazer o critério de densidade, o novo representante é escolhido do conjunto de maior cardinalidade entre os conjuntos de elementos associados a cada representante, e para satisfazer o propósito de cobertura espacial o elemento escolhido é o mais distante dos representantes do conjunto.

O fato do novo elemento ser do conjunto de maior cardinalidade, prioriza que áreas

de alta densidade sejam representadas, e sendo o mais distante dos representantes, permite que a amostra cresça de modo homogêneo garantindo uma boa distribuição espacial entre os elementos, inclinando-se a repetir o mesmo formato das partições do conjunto de dados. A quantidade de elementos associados a cada representante decresce a cada iteração, devido que elementos associados a outros representantes são associados ao novo representante.

O método possui um único parâmetro de entrada, chamado granularidade. Este parâmetro representa o quão espaçada a amostra será, ou seja, o valor determinado para o parâmetro impacta diretamente no tamanho da amostra, tal que quanto menor a granularidade, maior o número de representantes na amostra. A granularidade gr é combinada com a cardinalidade do conjunto de dados n para encontrar um limiar $wt = n \times gr$ que será utilizado para definir o número de elementos que serão associados a um determinado representante. O critério de parada de seleção de representantes é atendido quando o grupo de maior cardinalidade não atende o limiar wt preestabelecido.

O método utiliza duas etapas para a construção da amostra. Na primeira se seleciona os representantes para compor a amostra. Na segunda se realiza um pós processamento que tem como objetivo não permitir ruídos como representantes da amostra. São utilizados dois tratamentos. No primeiro, caso o conjunto de elementos associados ao representante possua cardinalidade inferior a wt, o representante é descartado da amostra. O segundo é baseado na distância, dado que os representantes se encontram distribuídos de modo homogêneo, a distância média entre os representantes é utilizada na verificação, caso o representante tenha uma distância superior a distância média, substitui-se o representante pelo elemento associado mais próximo do centroide computado considerando os representantes.

O método inicia selecionando de modo aleatório qualquer elemento do conjunto de dados como primeiro representante, ou de modo determinístico, escolhendo o elemento mais próximo de um elemento virtual computado como o mínimo de cada dimensão. Após a inicialização é realizado a seleção dos elementos representantes concluindo a primeira etapa do algoritmo, posteriormente, na segunda etapa é realizado o pré-processamento para tratamento dos ruídos.

Uma vantagem dos métodos baseados no algoritmo FFT é a distribuição espacial dos elementos na amostra, que é obtida na seleção de novos representantes na primeira etapa. Outra vantagem é realizar o tratamento de ruídos na amostra na segunda etapa do método. Além disso, uma vantagem característica do DENDIS é o fato de priorizar a presença de regiões densas na amostra. Um elemento fraco a ser observado está relacionado a dificuldade de estimativa do parâmetro gr, dado que uma estimativa ruim do parâmetro pode fornecer resultados de baixa qualidade.

2.2.2 DIDES

O DIDES é um método híbrido proposto em ROS FRÉDÉRIC; GUILLAUME (2017) semelhante ao DENDIS, que também possui duas etapas para construção de uma amostra S, combinando densidade e distância para selecionar os representantes. A diferença entre os métodos está na seleção dos representantes, tratamento dos ruídos e no critério de parada, sendo que o DIDES finaliza a seleção dos representantes utilizando um critério baseado em distância, enquanto o DENDIS utiliza um critério baseado em densidade. Ambos pretendem garantir a cobertura espacial e representatividade do conjunto de dados a um baixo custo computacional.

O parâmetro granularidade gr é o mesmo para DIDES e DENDIS, sendo este responsável por determinar o espaçamento da amostra. Os dois métodos utilizam o parâmetro combinado com a cardinalidade do conjunto de dados n para definir um limiar $wt = n \times gr$ que é utilizado no critério de parada e no tratamento de ruídos.

Na primeira etapa do DIDES, os representantes são selecionados com base na distância. Cada novo representante é o que possui maior distância (dmax) para todos os outros representantes selecionados, análogo a versão canônica do algoritmo FFT. O procedimento de inserção de representantes em S é realizado iterativamente até atender um critério de parada adaptativo. O critério é considerado adaptativo devido alterar seu valor em cada iteração ao considerar todas as cardinalidades dos representantes presentes em S. Em conjunto com a amostra S é gerado um subconjunto St que contém os representantes que atendem dois critérios: a cardinalidade dos elementos associados ao representante é inferior ao limiar wt e o dmax do representante é maior que a média do dmax de todos os representantes de St. O critério de parada é satisfeito e a etapa é finalizada quando o dmax do novo representante a ser inserido na amostra S, é inferior a média do dmax de todos os representantes de St.

Na segunda etapa é realizado um pós-processamento utilizando a densidade dos elementos associados aos representantes. Neste estágio são identificados e tratados os ruídos selecionados como representantes, fazendo com que apenas permaneçam os elementos das áreas densas presentes em S. Inicialmente calcula-se a média das cardinalidades dos elementos associados aos representantes presentes em St, e compara-se com a cardinalidade dos elementos associados aos representantes da amostra, caso seja inferior a média computada, o representante é descartado de S. Após a checagem individual, a densidade é considerada em nível de grupo, sendo que um grupo é definido como o conjunto de elementos alcançáveis entre si, dentro de uma determinada distância. A distância é a de 1 vizinho mais próximo entre representantes, no entanto, só são considerados do mesmo grupo se esta distância for inferior a duas vezes o dmax do representante. Um grupo contém o número de elementos representantes e o número total de elementos associados

aos representantes $G(t)$. Caso $G(t)$ seja inferior ao limiar wt o grupo é de baixa densidade sendo provavelmente composto por ruídos, por isto, o grupo é removido.

Apesar dos métodos DIDES e DENDIS serem bastante semelhantes, existem diferenças nítidas entre os métodos. A primeira diferença é encontrada na seleção dos representantes, dado que o DENDIS escolhe o representante do conjunto de elementos associados de maior cardinalidade, enquanto o DIDES seleciona o elemento mais distante dos representantes selecionados. A escolha do DENDIS prioriza selecionar os elementos das áreas mais densas, a do DIDES prioriza a cobrir todo o espaço. Outra diferença corresponde ao critério de parada. O DENDIS verifica se o novo elemento possui densidade relevante para compor a amostra, e o DIDES verifica se a distância do elemento a ser incluído na amostra contribuíra na distribuição espacial da amostra. A última diferença encontra-se no tratamento de ruídos, ambos tentam deixar a amostra o mais o homogênea possível, no entanto, o DENDIS substitui os representantes que se encontram muito distantes por algum elemento do conjunto de elementos associados, por outro lado, o DIDES realiza uma verificação de densidade em grupo, removendo os que são considerados grupos de baixa densidade.

Dado que o DIDES e DENDIS são semelhantes, suas vantagens e desvantagens também são. O DIDES também possui como vantagem uma etapa de tratamento de ruídos na amostra e um boa distribuição espacial característica do FFT, no entanto, como particularidade prioriza que a amostra seja distribuída espacialmente ao em vez de elementos densos. De modo equivalente ao DENDIS, este também possui como desvantagem a dificuldade de estimativa do parâmetro gr, dado que a estimativa do parâmetro compromete a qualidade da amostra.

2.2.3 ProTraS

Assim como DENDIS e DIDES, o algoritmo ProTraS (ROS FRÉDÉRIC; GUILLAUME, 2018) é um método híbrido baseado no algoritmo FFT que utiliza critérios de densidade e distância na construção da amostra do conjunto de dados. No entanto, diferentemente dos métodos anteriores, o ProTraS é baseado em teoria probabilística e não possui uma etapa exclusiva para tratamento de ruídos.

O método utiliza um conceito chamado custo de amostragem no critério de parada e objetiva a sua redução ao selecionar novos elementos para a amostra. Na seleção de representantes, escolhe-se o elemento mais distante do grupo com a maior probabilidade de redução do custo. Para computar a probabilidade de redução do custo de cada grupo é necessário a cardinalidade(w) e o elemento mais distante(d) do conjunto de elementos associados ao representante do grupo. Além disso, também são necessárias as seguintes informações da amostra: maior cardinalidade($max(w)$) dos conjuntos de elementos associ-

ados aos representantes e o elemento mais distante de todos os representantes($max(d)$). A probabilidade é computada pela seguinte combinação: $\frac{w \times d}{max(d \times w)}$. O custo de amostragem é calculado em cada iteração, considerando cada representante da amostra, como o somatório $w \times d$ de todos os representantes dividido pelo número de elementos do conjuntos de dados.

O único parâmetro do método é o parâmetro custo, o qual semelhantemente ao parâmetro granularidade do DIDES e DENDIS, faz com que o tamanho da amostra varie conforme sua estimativa. Os autores nomeiam o parâmetro como ε, usado no processo de amostragem como critério de parada. Para evitar qualquer confusão com o parâmetro do DBSCAN homônimo, este trabalho refere-se ao parâmetro do ProTraS como γ.

O ProTraS é composto por quatro etapas. A primeira seleciona um novo elemento no grupo com maior probabilidade de redução de custo. A segunda etapa associa cada elemento não selecionado ao elemento mais próximo presente na amostra. Na terceira etapa, o custo de amostragem é computado. Na quarta e última etapa, é verificado se o custo de amostragem calculado é maior que o parâmetro de custo.

Em ROS FRÉDÉRIC; GUILLAUME (2018) o autor do método afirma, baseado em testes experimentais, que o método é robusto a ruídos. Uma quantidade crescente de ruído aleatório uniforme (de 1% a 19%) foi adicionado aos dados e relatado que o método ainda é capaz de representar a distribuição dos dados mesmo com um nível significativo de ruídos.

Coresets são representações aproximadas do conjunto de dados. Algoritmos baseados em FFT produzem *coresets* mesmo que não sejam desenvolvidos para este propósito, visto que a amostra final encontrada é um (k, γ)-coreset do conjunto de dados completo. O ProTraS, ao contrário do DENDIS ou do DIDES, visa produzir um *coreset*, portanto, o custo de amostragem tem papel fundamental. Ele é utilizado em conjunto com o valor estipulado para o parâmetro γ no critério de parada, fazendo o método se encerrar quando atingido o custo desejado. O uso de cálculos de probabilidade permite que o algoritmo obtenha uma compensação entre representar a densidade e realizar a cobertura espacial do conjunto de dados, dado que quanto maior a distância ou maior a cardinalidade maior é a probabilidade de redução de custo, permitindo que ambos possam se sobressair no decorrer das iterações.

Uma vantagem do método corresponde a compensação entre representar a densidade e realizar a cobertura espacial, dado que permite que a amostra seja construída com elementos de regiões densas e obtenha o elemento mais distante dos representantes, replicando os formatos das partições dos conjuntos de dados. Além disso, pode ser considerado como vantagem a robustez a ruídos, sendo que não há um custo computacional adicional. No entanto, assim como seus antecessores DENDIS e DIDES, a estimativa do parâmetro pode ser uma dificuldade, dado que feito de forma equivocada compromete a qualidade da amostra.

2.2.4 Rough*-DBSCAN

O Rough-DBSCAN (VISWANATH P.; BABU, 2009), assim como os outros métodos já apresentados, possui duas etapas. Em sua primeira etapa aplica-se o algoritmo de agrupamento Leader (HARTIGAN, 1975) com a finalidade de obter os líderes como candidatos a compor a amostra. Na segunda etapa são filtrados alguns líderes encontrados na primeira etapa para compor a amostra, sendo que a decisão do líder pertencer ou não a amostra é baseada em sua densidade. Contudo, dado que é computacionalmente custoso calcular a densidade de cada líder, os autores propõem uma forma de estimá-la, denominada *Rough-Cardinality*, na qual se considera que a quantidade de líderes a uma distância inferior a ε é igual ao somatório dos seguidores de todos os líderes que são inferior a essa distância. Após a construção da amostra o DBSCAN é executado sobre a mesma, o grupo atribuído aos líderes é expandido a todos o seus seguidores e os elementos que ficaram sem grupo são definidos como ruídos.

O método ROUGH*-DBSCAN(R*) (LUCHI; RODRIGUES; VAREJÃO, 2019) é semelhante ao Rough-DBSCAN, mas a versão original do Leader foi modificada para fornecer uma maneira trivial de estimar a densidade de cada líder. Esta variação do algoritmo necessita de uma segunda varredura do conjunto de dados que, ao contrário do Leader original, associa um elemento não líder a vários líderes dentro da distância ε, tornando possível estimar a densidade de cada um dos líderes com base no número de elementos associados a ele. Essa alteração é motivada porque a estimativa de densidade por *Rough-Cardinality* apresenta dois tipos de erros que não estão presentes na estimativa de densidade proposta no método R*, o qual apresenta resultados superiores conforme relatado em LUCHI, RODRIGUES e VAREJÃO (2019).

O R* usa os parâmetros ε e $minPTS$ do DBSCAN e o parâmetro τ do Leader para executar o Leader*. Somente são selecionados para compor a amostra os líderes que contém mais de $minPTS$ seguidores ao redor de uma esfera de tamanho ε, sendo descartados os que não atendam à premissa. Sabendo que todos os líderes estão associados a mais que $minPTS$ seguidores em um a esfera de raio ε, todos possuem uma densidade acima da requerida pelo algoritmo DBSCAN. Deste modo, o parâmetro $minPTS$ do DBSCAN responsável por essa checagem passa a não ter utilidade e o DBSCAN é executado com $minPTS = 1$ na amostra de líderes. A última etapa do algoritmo expande os grupos obtidos pelos líderes para todos os respectivos seguidores obtidos na execução do Leader*. Nos casos de associação a mais de um líder, os duplicados recebem arbitrariamente o grupo de apenas um líder, considerado apenas o último grupo atribuído para cada elemento na iteração de expansão de grupos, sobrescrevendo o anterior se já atribuído. Os demais elementos sem associação são marcados como ruído.

Entre as vantagens a serem destacadas do algoritmo, encontra-se a qualidade da

estimativa de densidade dos elementos líderes. Apesar de ser necessário mais uma varredura no conjunto de dados, a estimativa de densidade é realizada a um custo computacional relativamente baixo. No entanto, o método possui a desvantagem de ser baseado no algoritmo Leader que é sensível a ordem que os dados são apresentados ao mesmo, isto implica em poder gerar amostras diferentes dependendo da ordem de iteração do Leader. Além disso, o único tratamento de ruídos do método é aquele realizado pelo próprio DBSCAN, ou seja, caso o ruído não esteja presente na amostra e seu líder seja rotulado como elemento denso, o ruído também será considerado um elemento denso.

3

A abordagem BIRCHSCAN

Este capítulo inicialmente introduz o algoritmo BIRCH, que serve como base para o método proposto, e em seguida, apresenta o BIRCHSCAN, um método para aplicação do DBSCAN em grandes conjuntos de dados.

3.1 BIRCH

O método de agrupamento BIRCH (*Balanced Interative Reducing and Clustering using Hierarchies*) (ZHANG; RAMAKRISHNAN; LIVNY, 1996) foi desenvolvido para trabalhar com conjuntos de dados muito grandes. O algoritmo funciona de forma hierárquica e dinâmica, agrupando entradas multidimensionais para produzir o agrupamento de melhor qualidade, considerando a memória disponível.

Os conceitos fundamentais usados pelo algoritmo BIRCH são os de *Clustering Feature* (CF) e CF-Tree. Uma CF corresponde a um resumo de um grupo de elementos e contém uma tripla de informações. Dado um grupo de N elementos com d dimensões, a tripla CF é formada por N, o vetor $\overrightarrow{LS}$ com d dimensões, que contém a soma linear de cada dimensão dos elementos do grupo e o vetor $\overrightarrow{SS}$ que é o somatório de cada dimensão ao quadrado de todos os elementos do grupo. A partir desses valores é possível encontrar o centroide $\frac{\overrightarrow{LS}}{N}$ e raio $\sqrt{\frac{\overrightarrow{SS}}{N} - (\frac{\overrightarrow{LS}}{N})^2}$ do grupo. Duas CF são unidas através da soma de seus componentes.

A CF-Tree é uma estrutura de dados árvore B+, que é a representação compacta de todo o conjunto de dados, sendo organizada pelos parâmetros B e $Threshold$. A estrutura da CF-Tree pode ser vista na Figura 6. Os nós não-folha possuem uma lista de CFs de tamanho máximo B no formato $[CF_i, filho_i]$ onde i = 1,2, ..., B. Esta lista contém um resumo do nível inferior tal que "$filho_i$" é um ponteiro para o nó filho e CF_i é a soma dos CFs do subgrupo representado por este filho. Um nó folha é definido por uma lista de CFs de tamanho máximo L no formato $[CF_i]$ onde i = 1,2, ..., L, de modo que cada CF

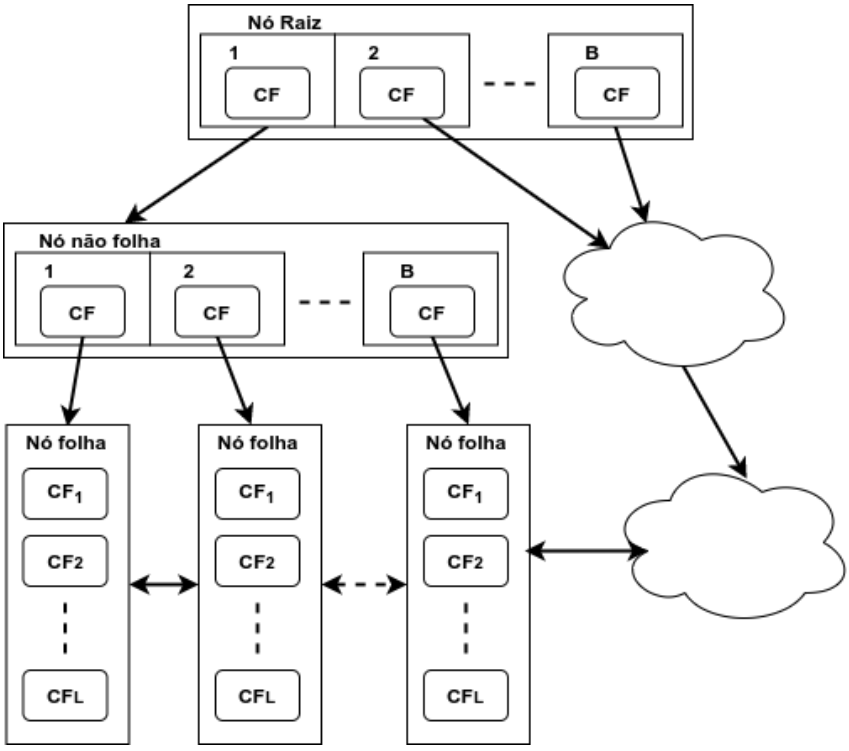

Figura 6 – Estrutura da CF-Tree do algoritmo BIRCH. Figura baseada na imagem apresentada em Wan e Yang (2013).

representa um subgrupo. Para que uma CF pertença a um nó folha, o raio do subgrupo deve ser inferior ao parâmetro *Threshold*.

A Figura 7 apresenta as quatro fases do algoritmo BIRCH.

A fase 1 carrega os dados na memória, varrendo todo o conjunto de dados e construindo uma CF-Tree inicial na memória. O BIRCH tenta representar as informações do conjunto de dados agrupando-as da melhor forma possível de acordo com a memória disponível. Ele começa com um valor arbitrário de *Threshold* fornecido como parâmetro inicial do algoritmo, caso fique sem memória antes de terminar a varredura de dados, o valor de *Threshold* é incrementado dinamicamente considerando a quantidade de elementos já inspecionados e o número de CFs em cada nó folha. Dado que quanto maior o Threshold menor é o tamanho da CF-Tree, uma nova e menor CF-Tree é reconstruída, reinserindo as folhas da antiga CF-Tree e continuando a verificação de onde parou.

A fase 1 também realiza o descarte de ruídos, de modo que, se um CF de algum nó folha tiver menos de 25% do número médio de todos elementos dos CFs de nó folha, este CF é descartado. Esta fase cria um resumo do conjunto de dados em memória, trazendo os benefícios de uma execução mais rápida, já que não há mais necessidade de operações de I/O e também reduzindo o problema de agrupamento dos dados originais, com maior precisão devido ao descarte de ruídos.

A fase 2 é uma fase opcional e é similar a Fase 1, porém, em vez de varrer o

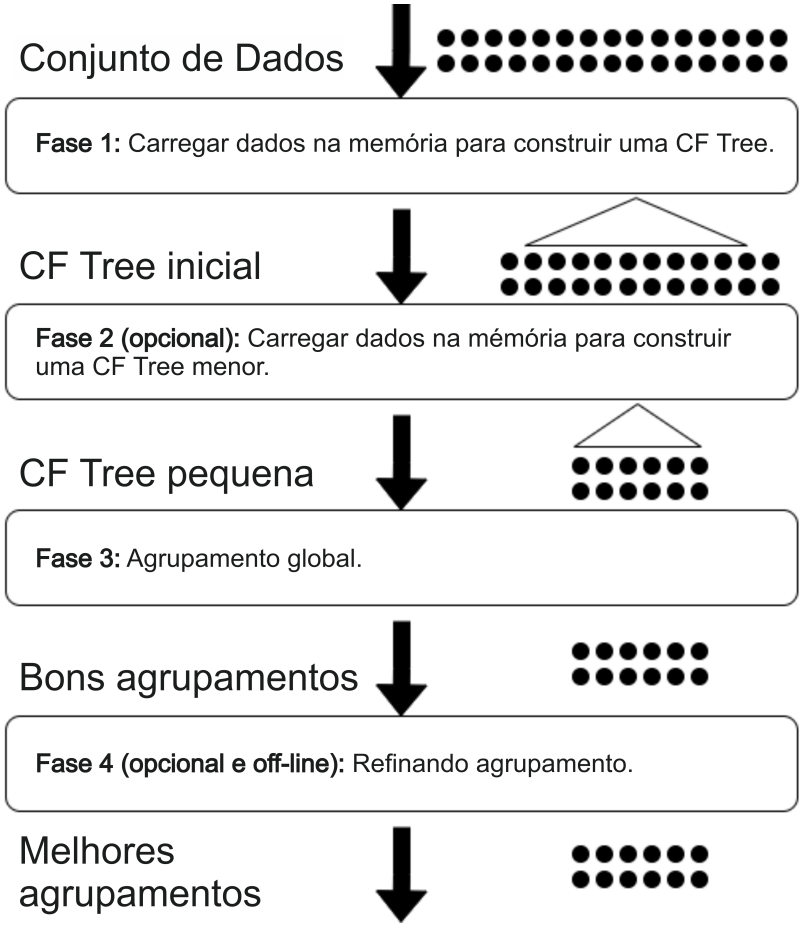

Figura 7 – Visão geral do BIRCH. Baseada na figura apresentada em ZHANG, RAMAKRISHNAN e LIVNY (1996).

conjunto de dados, a entrada são os nós folhas da CF-Tree gerada na Fase 1. O objetivo é reconstruir uma CF-Tree mais compacta. Para alcançar isso são juntados os subgrupos que possuem uma distância próxima. Ao inserir um CF verifica-se se a CF mais próxima "absorve" a CF a ser inserida, ou seja, se o raio do novo subgrupo gerado é inferior ao *Threshold*. Em caso positivo os subgrupos se tornam apenas um e novamente se remove os ruídos utilizando o mesmo processo de descarte de ruídos da fase 1, este processo resulta em uma CF-Tree menor.

A fase 3 realiza o agrupamento do conjunto de dados a partir dos subgrupos representados pelos nós folhas da CF-Tree. De modo ingênuo, é possível calcular o centroide de cada subgrupo, dado que a partir da identificação do centroide, o subgrupo passar a ser tratado como um único elemento e após isso aplicar um algoritmo de agrupamento qualquer. Outra opção é considerar o número N de elementos do subgrupo como se houvesse replicado o centroide N vezes, fazendo uma leve alteração em algum algoritmo de agrupamento para considerar os centroides e suas informações de contagem na realização do agrupamento. Isto possibilita resultados mais precisos, como por exemplo a validação de elementos densos no DBSCAN ficaria mais precisa com essa modificação, facilitando que

sejam realizados agrupamentos mais semelhantes do algoritmo de agrupamento aplicado no conjunto de dados inteiro. Como última opção, existe a possibilidade de adaptar um algoritmo existente diretamente na tripla CF do subgrupo, dado que a informação de cada CF é suficiente para calcular a maioria das métricas de distância e qualidade utilizadas por algoritmos de agrupamento.

A apresentação do BIRCH em ZHANG, RAMAKRISHNAN e LIVNY (1996) aplica um algoritmo de agrupamento hierárquico aglomerativo adaptado que utiliza a tripla CF de cada subgrupo para realizar o agrupamento de todo o conjunto de dados.

Após a fase 3, tem-se um conjunto de agrupamentos capaz de reproduzir a distribuição principal do conjunto de dados. No entanto, o conjunto pode conter imprecisões e ambiguidades, por exemplo, um elemento duplicado pertencendo a nós folhas distintos. A fase 4, que também é opcional, implica em um custo adicional para corrigir falhas e refinar os grupos. Os centroides dos subgrupos identificados na fase 3 são usados na fase 4 como sementes. Esta fase redistribui os elementos para as sementes mais próximas, gerando novos conjuntos de grupos, e também oferece a opção de descartar ruídos dos elementos muito distantes (distância maior que duas vezes o raio do subgrupo) de sua semente mais próxima, evitando que sejam incluídos no resultado.

As principais vantagens do BIRCH são apresentadas a seguir:

1. Possui computação local, onde cada decisão de agrupamento é feita sem a varredura de todo o conjunto de dados ou de todos os grupos existentes.

2. Explora a ideia de que o espaço dos dados não é ocupado uniformemente, ou seja, os elementos não são igualmente importantes para a proposta de agrupamento. Uma região de elementos densos é tratada coletivamente como um único elemento. Os elementos em regiões esparsas são tratados como ruídos e removidos opcionalmente.

3. Faz uso total da memória disponível para fornecer os melhores subgrupos possíveis, enquanto minimiza os custos de I/O (garantindo precisão e eficiência).

4. Se forem omitidas as fases 2 e 4 do algoritmo, o BIRCH é um algoritmo incremental que não exige tratar todo o conjunto de dados de uma única vez na memória, e requer uma única varredura no conjunto de dados.

3.2 BIRCHSCAN

Este trabalho combina os algoritmos de agrupamento BIRCH e DBSCAN para propor um método para agrupar grandes conjuntos de dados de forma computacionalmente eficiente e com resultados semelhantes ao que se obteria na aplicação isolada do DBSCAN em

todo o conjunto. O método proposto possui três parâmetros, os parâmetros ε e $minPTS$ do DBSCAN e o parâmetro δ. Os resultados sugerem que a proposta, denominada BIRCHSCAN, é bastante competitiva se comparada a outros métodos com a mesma finalidade.

O principal objetivo dos métodos de amostragem tendenciosa é gerar uma amostra que represente todo o conjunto de dados sem perder pequenas partições. O BIRCHSCAN usa o BIRCH para atingir esse objetivo, encontrando elementos representativos com base na estrutura de dados CF-Tree construída em seu processo de execução. Na verdade, a CF-Tree armazena um resumo do conjunto de dados de forma que cada entrada inserida em um nó folha não seja apenas um elemento, mas um subgrupo que engloba muitos elementos dentro de uma vizinhança esférica definida por um limite específico (parâmetro $Threshold$ do BIRCH). Este limite é obtido através do parâmetro ε do DBSCAN e do parâmetro δ do BIRCHSCAN, sendo $Threshold = \varepsilon \times \delta$. O BIRCHSCAN usa os centroides desses subgrupos como uma amostra para realizar o agrupamento. É necessário esclarecer que essa amostra enviesada, corresponde a um novo conjunto de dados de menor cardinalidade com elementos representativos que não fazem parte do conjunto de dados original.

Para fornecer uma melhor compreensão de como o BIRCH pode ser usado para criar uma amostra tendenciosa, um aplicativo exemplo simples de dois grupos é simulado e apresentado na Figura 8. A figura contém 100100 elementos (pontos pretos) com 2 grupos, um contendo 100000 elementos e o outro 100. O algoritmo BIRCH executado com $Threshold = 200000$ e $B = 1000$ gerou 314 centroides (pontos vermelhos) com base em seus CFs. Como pode ser visto, os centroides (pontos vermelhos) estão espalhados e presentes em ambos os grupos, não permitindo que informações do grupo com menor cardinalidade sejam perdidas.

O algoritmo 2 apresenta o BIRCHSCAN, com os três parâmetros ajustáveis. $minPTS$ e ε também são usados pelo DBSCAN. O parâmetro δ define o $Threshold$ usado pelo BIRCH. Todo o processo consiste em quatro etapas para o agrupamento dos dados.

Inicialmente, o BIRCH é aplicado a todo o conjunto de dados usando $minPTS$ como o número limite de subgrupos em cada nó não folha da CF-Tree (parâmetro B do BIRCH) e δ como um fator multiplicador aplicado a ε. Este produto define o $Threshold$ utilizado pelo algoritmo BIRCH para compor a CF-Tree. O BIRCHSCAN utiliza apenas as duas primeiras fases do BIRCH, uma vez que a fase de agrupamento global e seu refinamento são etapas desnecessárias no método proposto. As fases desconsideradas do algoritmo BIRCH servem para fazer o agrupamento e melhorar a qualidade do agrupamento, no entanto, como nosso propósito é realizar este agrupamento com o DBSCAN, utilizar essas fases seria redundante e consumiria recursos desnecessários. No final desta etapa, são obtidos os subgrupos gerados na CF-Tree.

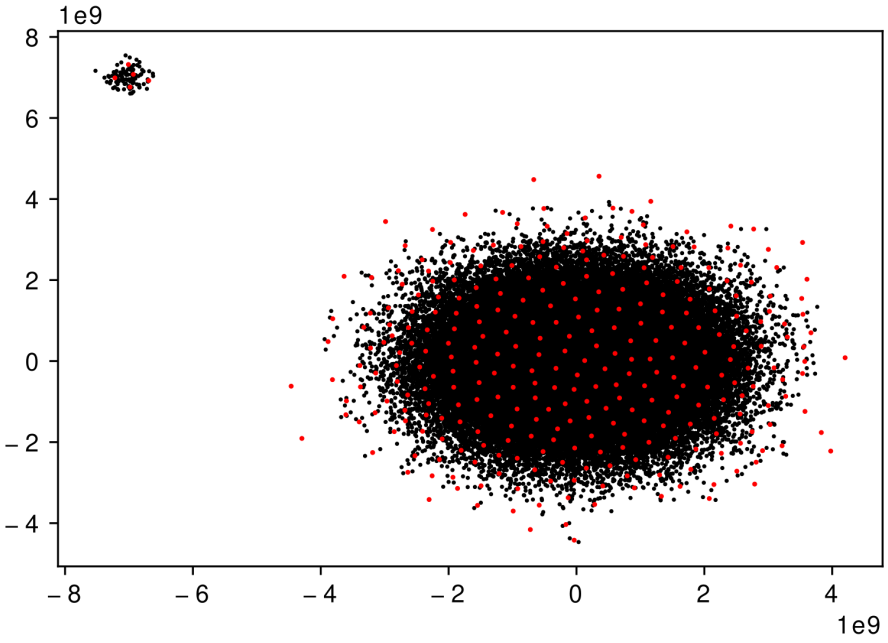

Figura 8 – Visualização da distribuição dos centroides BIRCH

Algorithm 2: BIRCHSCAN

Input : Conjunto D; ε; $minPTS$; δ
Output : $Clusters$

1 $Threshold \leftarrow \varepsilon \times \delta$

2 $S \leftarrow$ Executar o algoritmo `BIRCH` em D utilizando somente as duas primeiras fases com o parâmetro $Threshold$ e $B = minPTS$ e obter os subgrupos.

3 C, $W \leftarrow$ Obter centroides e pesos no subaglomerados.

4 $SampLb \leftarrow$ Executar o `DBSCAN` aplicado em C com os parâmetros ε,$minPTS$ e W para receber os centroides rotulados.

5 $Clusters \leftarrow$ Executar um método `EXPAND_CLUSTERS` para agrupar os subaglomerados S com os grupos obtidos pelos centroides em $SampLb$.

return : $Clusters_i (i = 1, ..., |D|)$

A segunda etapa obtém os centroides dos subgrupos para criar a amostra enviesada para aplicação do DBSCAN. Junto com os centroides, a segunda etapa também obtém o tamanho (número de elementos) dos subgrupos a serem usados como pesos para o DBSCAN. Dado que o DBSCAN é um algoritmo de agrupamento baseado em densidade, os pesos permitem que o agrupamento seja mais consistente ao identificar elementos densos, pois apenas um único elemento representa a quantidade de elementos de seu peso, isto permite que apenas um elemento possa ser considerado como denso.

Na terceira etapa, o DBSCAN é aplicado à amostra usando os parâmetros $minPTS$, ε e os pesos de cada elemento da amostra. Qualquer elemento da amostra com peso maior ou igual a $minPTS$ não é considerado um ruído. Contudo, elementos da amostra que não possuem peso maior ou igual a $mintPTS$ não necessariamente são considerados como ruído, dado que para esses elementos o fato de ser ou não ruído dependerá de sua vizinhança.

Na quarta e última etapa, a rotulagem de todo o conjunto de dados é realizada com base nos subgrupos retornados pelo BIRCH. Os elementos dos subgrupos obtêm o mesmo grupo de seus respectivos centroides. Por exemplo, se um centroide está agrupado em um grupo a, todos os elementos de seus subgrupo também serão agrupados no grupo a. Esse procedimento é válido para elementos densos como ruídos. Observe que todos os elementos do conjunto de dados são agrupados, pois a união dos subgrupos é o conjunto de dados inteiro. Além disso, o processo de rotulagem é uma tarefa rápida e econômica, uma vez que os subgrupos já foram criados.

O tamanho da CF-Tree gerada pelo algoritmo BIRCH é diretamente influenciado pelo parâmetro $Threshold$. Visto que ε é um parâmetro do DBSCAN escolhido de acordo com a aplicação, só é possível ajustar $Thresold = \varepsilon \times \delta$ ajustando o valor de δ. Antes de executar os experimentos, um estudo empírico foi conduzido para definir um valor razoável para δ.

A princípio o estudo empírico de δ foi realizado em conjuntos de dados sintéticos semelhante ao da Figura 8, com o objetivo de compreender melhor o comportamento do parâmetro. Um valor fixo foi definido para ε e variou-se o valor de δ. Pode-se identificar que para valores maiores que 1 todo elemento é considerado como denso ou ruído. Deste modo, não existe nenhuma conexão dentro de um raio ε entre nenhum centroide no momento da aplicação do conjunto de dados gerado pela saída do BIRCH e aplicado no DBSCAN. E para valores muito próximos a 0, obtém-se praticamente o conjunto de dados inteiro, não sendo uma opção interessante. Por isso, os estudos foram observados para o intervalo aberto entre 0 e 1.

Nos conjuntos de dados iniciais foi possível identificar que quanto mais próximo de 0, maior é o tamanho da amostra e o inverso também é verdadeiro, sendo consequência da variação do parâmetro $Threshold$ do BIRCH. Portanto, quanto menor o valor $Threshold = \varepsilon(fixo) * delta$, maior será a quantidade de subgrupos retornados, e maior

será o conjunto de dados de elementos representantes gerados pelo BIRCH aplicados no DBSCAN. Contudo, não necessariamente uma amostra maior apresentou resultados superiores. Amostras pequenas demais também não apresentaram qualidade satisfatória. Após análises, afim de se obter equilíbrio, foi selecionado o valor de $\delta = 0,5$, devido este apresentar resultados satisfatórios independente do conjunto de dados considerado. Depois desse estudo preliminar, também foram utilizados para esses testes os conjuntos de dados sintéticos dos experimentos de ROS FRÉDÉRIC; GUILLAUME (2018), os quais confirmaram que o velor de $\delta = 0,5$ obtém resultados satisfatórios. Portanto, ele foi usado em todos os experimentos deste trabalho.

4

Experimentos computacionais

Este capítulo descreve os experimentos computacionais realizados neste trabalho. Primeiramente, apresenta detalhadamente a metodologia de experimentação utilizada e depois descreve os resultados obtidos.

4.1 Metodologia Experimental

Esta seção apresenta a metodologia experimental utilizada para avaliar o desempenho dos métodos. Ela começa descrevendo os conjuntos de dados considerados neste trabalho. A seguir, é apresentado como os experimentos foram conduzidos. Por fim, descreve a métrica de avaliação usada para comparar o desempenho dos métodos.

4.1.1 Bases de dados

Os conjuntos de dados usados para realizar a comparação dos métodos estão descritos na Tabela 1. Eles são todos públicos e foram selecionados a partir das seguintes fontes:

- Os conjuntos de dados de 1 a 7 estão disponíveis no repositório LIBSVM [1]. Eles também foram usados no trabalho de LUCHI, RODRIGUES e VAREJÃO (2019),
- Os conjuntos de dados de 8 a 11 estão disponíveis no repositório de dados da escola de informática da Eastern University of Finland [2]. Eles também foram usados no trabalho de ROS FRÉDÉRIC; GUILLAUME (2018),
- Os conjuntos de dados de 12 a 25 estão disponíveis no repositório de domínio público UCI [3].

[1] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/
[2] https://cs.joensuu.fi/sipu/datasets/
[3] https://archive.ics.uci.edu/

Os conjuntos de dados foram escolhidos para verificar o comportamento dos métodos em diferentes cenários, variando dimensões de 2 a 617, o tamanho de 150 a quase 3 milhões de elementos, a forma e densidade dos grupos. A Tabela 1 descreve o número de elementos e a quantidade de características de cada conjunto de dados. Para os experimentos, apenas conjuntos de dados não sintéticos foram considerados para evitar qualquer viés que possa ocorrer entre um formato de conjunto de dados específico e um algoritmo.

Tabela 1 – Conjuntos de dados usados nos experimentos

#	Conjunto	Elementos	Características
1	Abalone	4177	8
2	Cadata	20640	8
3	Letter	20000	16
4	Mushrooms	8124	112
5	Pendigits	10992	10
6	Sensorless	58509	48
7	Shuttle	58000	7
8	r3	13467	2
9	r4	950	10
10	r7	34112	3
11	r8	45781	3
12	breast	699	9
13	leaves	1600	100
14	wdbc	569	32
15	iris	150	4
16	wine	178	13
17	yeast	1484	8
18	thyroid	215	5
19	glass	214	9
20	HAR	10.299	561
21	Epileptic Seizure Recognition	11.500	178
22	Isolet	7.797	617
23	Bitcoin	2.916.697	8
24	Household Power Consuption	2.075.260	7
25	US Census 1990	2.458.286	68

4.1.2 Experimentos

Os conjuntos de dados foram usados separadamente em três experimentos diferentes. Os conjuntos de dados 1 a 19 foram utilizados no Experimento 1 com o objetivo de comparar a qualidade dos métodos (BIRCHSCAN, DENDIS, DIDES, ProTraS e R*). A qualidade dos métodos indica o quão próximo os grupos formados pelos métodos estão daqueles gerados pela execução do DBSCAN em todo o conjunto de dados. Os conjuntos de dados 1 a 19 têm tamanhos limitados, que permitem a aplicação do DBSCAN em todo o conjunto

de dados. Apesar do tamanho dos conjuntos de dados, este experimento é relevante por permitir investigar e comparar de forma mais ampla os métodos de amostragem em várias configurações distintas.

Para investigar o comportamento dos métodos em grandes conjuntos de dados, dois outros experimentos foram conduzidos. O Experimento 2 avalia a escalabilidade do BIRCHSCAN em comparação com o DBSCAN e os outros métodos de amostragem. Os conjuntos de dados 20 a 22, que são grandes em número de características ou número de exemplos, foram usados. Esses conjuntos de dados são custosos devido o seu tamanho, porém ainda é possível executar o DBSCAN para realizar a avaliação. Embora ainda seja preferível executar o DBSCAN nesses tipos de conjuntos de dados, este experimento vai um pouco além na comparação dos métodos de amostragem em conjuntos de dados maiores. Mesmo que não seja possível investigar uma variedade mais ampla de configurações dos valores dos parâmetros do DBSCAN, como no experimento 1, os resultados deste experimento devem ter maior probabilidade de serem reproduzidos ao executar esses métodos em conjuntos de dados nos quais não é viável a execução do DBSCAN.

O experimento 3 concentra-se apenas em realizar uma comparação de tempo entre os métodos de amostragem avaliados. Os conjuntos de dados 23 a 25 são muito grandes para a execução do DBSCAN, devido ao alto custo computacional, não sendo possível verificar se os métodos de amostragem apresentam resultados de boa qualidade. No entanto, este também é um experimento importante para verificar se os métodos realmente atingem seu propósito de permitir um agrupamento supostamente semelhante ao DBSCAN sempre que a execução do próprio DBSCAN não for viável.

Os métodos de amostragem usam o DBSCAN para agrupar a amostra. Consequentemente, é necessário definir os valores dos parâmetros ε e $minPTS$ do DBSCAN. O primeiro define uma vizinhança esférica, enquanto o último define o número mínimo de elementos necessários para considerar um elemento como de núcleo. Além disso, cada método possui parâmetros específicos que influenciam diretamente na amostragem. Este trabalho usa os mesmos valores de parâmetros definidos pelos autores em seus artigos originais. Portanto, para os métodos DIDES e DENDIS o parâmetro de granularidade é igual a 0,1. Para o algoritmo ProTraS, o parâmetro de custo γ também é igual a 0,1. Para o algoritmo R*, o parâmetro τ é igual a $\frac{\varepsilon}{2}$. Finalmente, para o BIRCHSCAN, o parâmetro δ é definido como 0,5.

Os três experimentos realizam procedimentos diferentes. O experimento 1 avalia os métodos de amostragem para diferentes valores dos parâmetros do DBSCAN. O processo de avaliação foi inspirado no trabalho de ROS FRÉDÉRIC; GUILLAUME (2018). O procedimento considera que os parâmetros são específicos para cada conjunto de dados e sua modificação permite gerar grupos de diferentes tamanhos e formas.

Da mesma forma em ROS FRÉDÉRIC; GUILLAUME (2018), o parâmetro $minPTS$

é fixado como $max(1; 0,01 \times$ número de exemplos de todo o conjunto de dados). Um conjunto de valores de ε é usado para avaliar grupos de diferentes tamanhos e formas. Este conjunto de valores é determinado para cada conjunto de dados executando o DBSCAN sucessivamente com diferentes valores de ε até que ele retorne dois grupos. Deixe D_{init} representar este valor, então o intervalo $[\frac{D_{init}}{1000}, D_{init}]$ de 1000 valores equidistantes é considerado como o parâmetro ε para executar o DBSCAN.

As amostras geradas por ProTraS, DENDIS e DIDES são independentes dos parâmetros do DBSCAN, ou seja, para esses métodos apenas uma única amostra é obtida e, posteriormente, o DBSCAN é executado nesta amostra selecionada para os 1000 valores de ε. Por outro lado, para os métodos BIRCHSCAN e R*, os valores de ε também são usados para gerar a amostra. Portanto, 1000 amostras diferentes são criadas, uma para cada valor de ε.

Este trabalho realiza uma pequena alteração no procedimento usado em ROS FRÉDÉRIC; GUILLAUME (2018) descartando valores muito pequenos para ε. O DBSCAN não retorna grupos se o valor de ε for muito pequeno porque nenhum grupo denso será encontrado. Portanto, este trabalho considera apenas valores de ε para o qual alguns grupos são detectados, e não apenas ruídos. Observe que o valor de ε influencia diretamente o número de grupos relatado pelo DBSCAN. De fato, considerando $minPTS$ fixo, quanto maior o valor de ε, menor o número de grupos. Portanto, o procedimento do Experimento 1 testa os métodos sobre diferentes números de grupos, permitindo uma comparação mais diversa e completa.

O experimento 2 avalia a qualidade e os tempos de execução do agrupamento ao aplicar métodos de amostragem a conjuntos de dados de alta dimensão. Apenas quatro valores diferentes dos parâmetros DBSCAN para cada conjunto de dados foram considerados devido o maior custo computacional. Os parâmetros para essas execuções foram selecionados com base no critério de que o DBSCAN não poderia retornar todos os elementos como ruídos e que o número de grupos difere entre as execuções.

Os experimentos 1 e 2 realizam as seguintes etapas para cada conjunto de dados e conjunto de valores de parâmetros do DBSCAN considerados.

1. Executa o algoritmo DBSCAN em todo o conjunto de dados;
2. Executa cada método de amostragem usando a configuração específica de valores de parâmetros;
3. Expande os grupos para todo o conjunto de dados;
4. Calcula a métrica para comparar as saídas dos métodos de amostragem com as saídas do DBSCAN em todo o conjunto de dados;

Para comparar o resultado dos algoritmos com o DBSCAN, é necessário expandir os grupos para todo o conjunto de dados (etapa 3 acima). Os métodos BIRCHSCAN e R* usam características do próprio método para expandir grupos para todo o conjunto de dados. Para o BIRCHSCAN, todos os elementos dos subgrupos obtidos pelo BIRCH estão agrupados nos mesmos grupos de seus respectivos cetroides. Para R*, todos os seguidores obtidos pelo Leader* são agrupados com no mesmo grupo obtido pelo líder presente na amostra.

Os algoritmos apresentados em ROS FRÉDÉRIC; GUILLAUME (2017), ROS FRÉDÉRIC; GUILLAUME (2016), ROS FRÉDÉRIC; GUILLAUME (2018) não explicam como expandir os grupos para todo o conjunto de dados. A comparação de desempenho nesses trabalhos é baseada na comparação dos resultados do DBSCAN na amostra e em todo o conjunto de dados. A comparação avalia se os elementos da amostra pertencem ao mesmo grupo em ambos os agrupamentos. O problema com essa abordagem é que os exemplos que não estão presentes na amostra não são considerados na comparação. Inspirado em Chen et al. (2019) e na ideia de que elementos vizinhos são altamente possíveis de serem do mesmo tipo (elemento de núcleo, elemento de fronteira ou ruído), uma abordagem de vizinho mais próximo foi usada para expandir os grupos para todo o conjunto de dados.

Assim, o processo de rotulagem de DENDIS, DIDES e ProTraS consiste em aplicar o DBSCAN na amostra e, em seguida, começar a rotular os elementos do conjunto de dados não presentes na amostra usando o vizinho mais próximo. Deve-se notar que quando um elemento é considerado um ruído na amostra, seus vizinhos mais próximos não presentes na amostra também são considerados ruídos.

Da mesma forma que no Experimento 2, apenas quatro execuções de cada método de amostragem foram realizadas no Experimento 3 para cada conjunto de dados. Cada execução usa uma configuração diferente de valores de parâmetros do DBSCAN. Como o foco deste experimento é a comparação do tempo de execução entre todos os métodos de amostragem, a faixa de variação dos parâmetros do DBSCAN não tem grande importância uma vez que os mesmos valores são usados para todos os métodos. No entanto, para evitar rotular todos os elementos como ruídos ou agrupar todos os elementos em um único grupo, os parâmetros foram escolhidos por meio de testes empíricos. Os valores foram escolhidos para atingir um número diferente de grupos entre as execuções ao aplicar R* aos dados. O método R* foi a escolha para encontrar os parâmetros por entregar, entre os concorrentes, o melhor desempenho no Experimento 2, evitando qualquer favorecimento ao BIRCHSCAN. O terceiro conjunto de experimentos realiza apenas a segunda e a terceira etapas listadas acima e o tempo de execução desse processo é relatado.

4.1.3 Métrica de avaliação

O objetivo dos dois primeiros conjuntos de experimentos é avaliar o quão próximos os agrupamentos dos métodos de amostragem estão dos resultados do DBSCAN aplicado a todo o conjunto de dados. Duas métricas são comumente usadas para avaliar a similaridade entre agrupamentos, o *Rand Index* (RI) (RAND, 1971) e o *Adjusted Rand Index* (ARI) (HUBERT LAWRENCE; ARABIE, 1985). Suas respectivas formulas são apresentadas nas equações (4.1) e (4.2).

$$RI = \frac{a+b}{\binom{n}{2}} \tag{4.1}$$

Assumindo que C é o conjunto verdadeiro de grupos e K o conjunto de grupos atribuídos, na equação (4.1) definimos a e b como:

- a: Número de pares de elementos que estão no mesmo conjunto em C e no mesmo conjunto em K.
- b: Número de pares de elementos que estão em conjuntos diferentes em C e em conjuntos diferentes em K.

$$ARI = \frac{\sum_{i,j}\binom{n_{i,j}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{4.2}$$

Ambas as métricas tem valor máximo 1, que é alcançado quando os agrupamentos são exatamente iguais. Contudo, é possível que dois conjuntos aleatórios tenham um RI próximo a 1 no caso de muitos grupos. Isso ocorre porque há uma grande chance de que um par de itens seja agrupado em grupos diferentes em ambos os agrupamentos, e isso é contado como um acerto no cálculo da métrica RI. Por esta razão, a métrica selecionada para medir o desempenho dos métodos é o *Adjusted Rand Index* (ARI). O ARI é uma correção de RI que estabelece uma linha de base usando a similaridade esperada de um modelo aleatório.

A expressão matemática do ARI é dada na equação (4.2). Nessa equação, $n_{i,j}, a_i, b_j$ são valores da Tabela de contingência 2. Considere dois agrupamentos X e Y com r e s grupos respectivamente, o elemento $n_{i,j}$ desta tabela representa o número de objetos que foram agrupados no $i-$*ésimo* grupo no agrupamento X e no $j-$*ésimo* grupo no agrupamento Y. Os valores a_i, b_j representam as somas marginais.

O ARI pode gerar valores negativos, que significa que o resultado obtido é inferior ao esperado por um agrupamento aleatório. No contexto deste artigo, o ARI é calculado

Tabela 2 – Tabela de contingência usada para computar o ARI

X/Y	Y_1	Y_2	...	Y_s	somas
X_1	$n_{1,1}$	$n_{1,2}$	...	$n_{1,s}$	a_1
X_2	$n_{2,1}$	$n_{2,2}$	...	$n_{2,s}$	a_2
...	...	...	...	...	...
X_r	$n_{r,1}$	$n_{r,2}$	...	$n_{r,s}$	a_r
somas	b_1	b_2	...	b_s	n

comparando o agrupamento obtido por cada método com o agrupamento obtido pelo DBSCAN aplicado em todo o conjunto de dados.

Com o objetivo de verificar se existe diferença estatística no desempenho entre os algoritmos avaliados, foi realizado o teste estatístico de Friedman (FRIEDMAN, 1937; FRIEDMAN, 1940). Em caso de diferenças significativas entre os métodos, o teste post-hoc de Nemenyi (NEMENYI, 1963) foi empregado para verificar a diferença crítica entre os pares de algoritmos.

4.2 Resultados

Os experimentos foram realizados em uma máquina com a seguinte configuração: sistema operacional Linux Ubuntu 18.04.5 LTS, processador Intel (R) Core (TM) i5-4200U CPU a 1,60 GHz e memória RAM de 16 Gb. Os métodos R*, ProTraS, DIDES e DENDIS são implementados em linguagem C / C ++. O código-fonte desses algoritmos foi gentilmente cedido por seus respectivos autores. O BIRCHSCAN faz uso das bibliotecas sklearn, numpy e pandas com implementações escritas em C/C++ e encapsuladas em python. Como todos os métodos são implementados em C/C ++, as comparações de tempo são consideradas válidas. A expansão do grupo para os métodos DIDES, DENDIS e ProTraS foi implementada em bibliotecas python, que também contêm código escrito em C/C++.

Foi utilizada nos experimentos a implementação do DBSCAN da biblioteca python sklearn. Esta foi configurada para selecionar uma das buscas kd-tree, ball-tree e força bruta para computação dos vizinhos mais próximos. A implementação do DBSCAN escolhe qual busca aplicar com base nas propriedades do conjunto de dados.

Os resultados são apresentados nas três subseções a seguir.

4.2.1 Experimento 1: Avaliação dos métodos de amostragem para diferentes valores dos parâmetros do DBSCAN

A Tabela 3 mostra, para cada conjunto de dados, o número de vezes que o DBSCAN foi executado sem considerar todos os exemplos como ruídos e os resultados da métrica ARI para cada um dos métodos de amostragem. Para cada método, μ, σ e $\circ$ são respectivamente a média, o desvio padrão e a classificação (do melhor ao pior) com base no desempenho médio do ARI. Além disso, quando um método específico não foi executado em um determinado conjunto de dados, ele é classificado como último nos resultados, pois um algoritmo que não executa é pior do que aquele que executa.

O algoritmo de agrupamento DENDIS não foi executado em alguns conjuntos de dados por exceder o número de dimensões aceitas pela implementação fornecida. Neste caso, a célula da tabela foi preenchida com $-$. O algoritmo BIRCHSCAN falhou ao executar com êxito todas as execuções no conjunto de dados SENSORLESS. O intervalo a partir de D_{init} tem valores muito pequenos para ε, portanto requer muita memória para ser executado. Nesse caso, o resultado médio do ARI está associado ao número de execuções bem-sucedidas (mostrado entre parênteses). Como o número de execuções ainda é grande, consideramos válida a comparação de desempenho. Os valores em negrito indicam o melhor desempenho médio obtido em cada conjunto de dados.

As duas últimas linhas da Tabela 3 resumem os resultados. A linha *Médias* calcula a média de μ, σ e $\circ$ entre os conjuntos de dados. A linha *Vitórias* calcula o número de vitórias (número de classificações igual a 1) para cada método. Os resultados sugerem que o BIRCHSCAN alcançou um desempenho ARI superior. Este apresenta o desempenho médio geral mais alto com a menor variabilidade. Além disso, também alcançou o ranking médio mais baixo e relata o maior número de vitórias.

O teste estatístico de Friedman (FRIEDMAN, 1937; FRIEDMAN, 1940) visa verificar se há diferença estatística entre os algoritmos avaliados. O teste de Friedman foi aplicado considerando as médias dos 19 conjuntos de dados do Experimento 1 (Tabela 3). A estatística de Friedman resultou em um valor de p igual à 0,0002, o que faz com que a hipótese nula seja rejeitada a um nível de significância de 0,99%, ou seja, os algoritmos são significativamente diferentes. O teste post-hoc de Nemenyi foi empregado para detectar a diferença crítica entre os algoritmos. A Tabela 4 mostra os valores-p do teste de Nemenyi para cada par de métodos.

Todos os métodos que apresentam uma diferença significativa ($p < 0,05$) são marcados em negrito. Pode-se dizer que o BIRCHSCAN possui uma diferença significativa em relação ao DENDIS, DIDES e R*. No entanto, não é estatisticamente possível fazer

Tabela 3 – Resultados do Experimento 1. Número de execuções do DBSCAN por conjunto de dados, média e desvio padrão dos valores ARI em porcentagem e classificação dos algoritmos. Os maiores valores médios por conjunto de dados estão em negrito.

Bases	**Qtd**	**ProTraS**			**BIRCHSCAN**			**DENDIS**			**DIDES**			**R***		
		μ	σ	○	μ	σ	○	μ	σ	○	μ	σ	○	μ	σ	○
breast	1000	0,06	2,00	4	**97,08**	4,79	1	0,06	2,00	4	0,06	2,02	4	94,52	0,03	2
leaves	553	**77,02**	28,68	1	71,4	10,06	2	-	-	5	0	0,05	4	29,76	13,55	3
wdbc	1000	**99,86**	2,37	1	95,95	11,81	3	-	-	5	53,96	29,86	4	99,43	4,45	2
iris	1000	54,55	43,11	4	**74,83**	23,40	1	32,05	31,20	5	62,94	44,52	2	60,25	42,66	3
wine	553	35,29	39,80	4	**83**	15,54	1	9,87	11,77	5	38,7	39,74	3	38,71	39,74	2
yeast	741	83,26	24,33	2	74,2	9,87	3	57,94	16,46	4	**96,86**	3,06	1	52,50	26,62	5
thyroid	827	63,82	36,66	4	70,48	18,07	3	42,58	28,62	5	**86,83**	22,21	1	76,44	19,21	2
glass	917	73,61	33,84	4	79,52	8,63	3	59,67	33,30	5	**92,11**	9,92	1	83,03	11,62	2
r3	1000	**80,57**	16,44	1	68,74	19,69	2	67,45	24,89	3	60,86	27,68	4	47,80	29,46	5
r4	844	87,68	17,87	2	34,79	17,12	5	72,85	22,92	3	**98,61**	11,34	1	65,20	33,59	4
r7	728	72,40	30,02	2	62,06	20,32	3	**81,86**	19,97	5	7,82	19,25	2	0,54	1,79	1
r8	788	73,44	28,30	4	78,07	29,81	3	46,29	22,58	5	85,16	33,18	2	**86,49**	33,43	1
Abalone	952	94,25	17,89	2	88,95	9,25	4	**96,42**	11,46	1	88,55	28,55	5	92,13	19,36	3
Cadata	580	65,51	16,22	3	**80,42**	10,64	1	75,72	13,16	2	7,46	7,52	5	7,47	10,97	4
Letter	341	5,63	4,87	3	**59,67**	13,00	1	12,66	9,58	2	0.00	0,00	4	-1,91	9,89	5
Mushrooms	429	-0,29	1,39	4	60,47	34,46	2	-	-	5	0,78	1,49	3	**73,73**	37,91	1
Pendigits	611	-1,21	3,95	5	**67,02**	19,73	1	4,51	7,71	3	-1,19	2,86	4	5,79	5,42	2
Sensorless	701	24,07	25,04	3	**78,85** (625)	16,85	1	-	-	5	17,04	25,90	4	67,97	39,01	2
Shuttle	274	35,75	37,41	3	78,64	13,14	2	**89,98**	11,90	1	-0,45	0,52	4	-1,11	0,75	5
Médias		53,96	21,59	2.95	**73,63**	**16,11**	**2.21**	49,99	17,83	3.63	41,90	16,30	3.16	51,51	19,97	3.05
Vitórias				3			**7**			3			4			2

a mesma afirmação ao comparar o BIRCHSCAN e o ProTraS. Apesar de não encontrar diferença significativa, o BIRCHSCAN obteve média superior, classificação inferior, maior número de vitórias e menor variância do que o ProTraS. Estas são fortes indicações da superioridade do BIRCHSCAN sobre o ProTraS.

Tabela 4 – Teste Posthoc de Nemenyi

	ProTraS	**BIRCHSCAN**	**DENDIS**	**DIDES**
BIRCHSCAN	0,590	-	-	-
DENDIS	0,648	**0,044**	-	-
DIDES	0,648	**0,044**	0,900	-
R*	**0,033**	**0,001**	0,532	0,532

Embora o Experimento 1 não tenha como foco o tempo de execução, uma vez que os conjuntos de dados usados para comparar a qualidade dos métodos são muito pequenos, a Tabela 5 relata o tempo de execução médio entre conjuntos de dados para todos os métodos. Os tempos de execução do DBSCAN também são apresentados para uma melhor comparação. Os melhores resultados são apresentados em negrito. Observe que DBSCAN e BIRCHSCAN sempre obtiveram tempos de execução menores do que os outros métodos. Nos maiores conjuntos de dados (como Letter, Mushrooms, Pendigits e Sensorless), o BIRCHSCAN foi consideravelmente mais rápido do que o DBSCAN.

Tabela 5 – Tempos de execução em segundos do Experimento 1

Bases	**ProTraS**	**BIRCHSCAN**	**Dendis**	**Dides**	**R***	**DBSCAN**
breast	0.8	0.2	0.82	0.72	0.99	**0.01**
leaves	0.62	0.36	-	2.1	1.93	**0.23**
wdbc	1.02	0.03	-	0.89	0.62	**0.02**
iris	0.32	0.07	0.25	0.21	0.25	**0.003**
wine	0.54	0.22	0.39	0.4	0.27	**0.004**
yeast	1.36	0.23	1.73	1.59	1.48	**0.08**
thyroid	0.33	0.07	0.41	0.28	0.32	**0.004**
glass	0.33	0.05	0.35	0.29	0.31	**0.004**
r3	15.72	**0.73**	14.92	13.14	25.66	1.52
r4	1.04	0.12	1.10	0.95	1.55	**0.02**
r7	43.21	**2.08**	39.89	32.60	34.89	3.13
r8	37.99	**1.38**	45.09	54.02	45.23	5.8
Abalone	6.05	**0.2**	11.41	4.17	3.69	0.34
Cadata	24.93	2.68	37.11	20.59	23.04	**1.22**
Letter	24.15	**3.25**	38.15	19.91	25.23	17.08
Mushrooms	22.41	**0.92**	-	20.14	14.62	23.28
Pendigits	23.17	**1.43**	30.46	17.09	20.54	3.04
Sensorless	113.62	**14.16**	-	75.85	68.73	123.8
Shuttle	52.84	**3.58**	59.93	44.54	45.30	6.06

4.2.2 Experimento 2: Avaliação de qualidade e tempo em conjuntos de dados de grandes dimensões

Os resultados do Experimento 2 mostram o tempo gasto pelo DBSCAN, o tempo gasto por cada método de amostragem e os respectivos ARIs, apresentados na Tabela 6. A célula da tabela é preenchida com − em casos em que o método não pôde ser executado devido a alguma limitação específica. O método DENDIS não pôde ser executado em nenhum conjunto de dados utilizado no Experimento 2. Observe que ARI é igual a zero em ProTraS e DIDES devido ao fato de que os resultados desses métodos são equivalentes a um método aleatório , isto provavelmente ocorreu devido a implementação dos métodos fornecida pelos autores não conseguir trabalhar com conjuntos de dados grandes. Em contraste, o BIRCHSCAN apresentou altos valores de ARI em todos os conjuntos de dados, enquanto ainda é o método mais rápido de execução. Os resultados do experimento sugerem que o método proposto supera as alternativas em conjuntos de dados de alta dimensão.

Tabela 6 – Resultados do Experimento 2. Valores dos parâmetros, tempos de execução em segundos e ARI

Conjunto	**Parameters**		**DBSCAN**	**BIRCHSCAN**		**ProTraS**		**DENDIS**		**DIDES**		**R***	
	ϵ	$minPTS$	*Tempo*	*Tempo*	*ARI*	*Tempo*	*ARI*	*Tempo*	*ARI*	*Tempo*	*ARI*	*Tempo*	*ARI*
HAR	3.5	5	114.61	**64.68**	**79.57**	93.26	0.0	-	-	79.18	0.0	-	-
	4		95.30	**47.47**	**87.51**	89.63	0.0	-	-	81.03	0.0	-	-
	4.5		115.41	**44.70**	**74.20**	91.97	0.0	-	-	88.35	0.0	-	-
	5		111.09	**27.05**	**85.90**	90.65	0.0	-	-	80.63	0.0	-	-
Epileptic Seizure Recognition	1000	5	63.12	**16.51**	**96.92**	37.24	0.0	-	-	41.17	0.0	42.22	8.78
	1500		67.20	**6.68**	**93.85**	37.12	0.0	-	-	44.70	0.0	33.75	32.34
	2000		66.46	**6.98**	**91.24**	37.00	0.0	-	-	44.24	0.0	27.24	24.91
	2500		67.73	**4.19**	**88.13**	37.21	0.0	-	-	44.58	0.0	28,28	49.38
Isolet	9	5	116.08	**45.72**	**57.86**	65.84	0.0	-	-	68,06	0.0	-	-
	10		116.72	**35.85**	**89.58**	66.02	0.0	-	-	67.14	0.0	-	-
	11		117.65	**29.31**	**64.25**	66.23	0.0	-	-	67.75	0.0	-	-
	12		117.43	**19.62**	**96.53**	66.53	0.0	-	-	65.09	0.0	-	-

4.2.3 Experimento 3: Avaliação de tempo em grandes conjuntos de dados

A Tabela 7 mostra os tempos de execução dos métodos em conjuntos de dados muito grandes. O DBSCAN não pôde ser executado devido as limitações de recursos de hardware. O BIRCHSCAN teve tempos de execução significativamente maiores do que o ProTraS, DENDIS e DIDES em dois processos de agrupamento do conjunto de dados BITCOIN e um do conjunto de dados *HouseHold*. Por outro lado, o BIRCHSCAN relatou os menores tempos de execução em 9 das 12 execuções (75%). A execução de ProTraS, DIDES e DENDIS no conjunto de dados US CENSUS 1990 levou a falha de segmentação que pode ser causada pela alta dimensionalidade e número de exemplos deste conjunto de dados.

Tabela 7 – Resultados do Experimento 3. Métodos aplicados em conjuntos de dados muito grandes. Valores dos parâmetros especificados e tempos de execução em segundos.

Dataset	**Parameters**		**BIRCHSCAN**	**ProTraS**	**DENDIS**	**DIDES**	**R***
	ϵ	$minPTS$	$Tempo$	$Tempo$	$Tempo$	$Tempo$	$Tempo$
BITCOIN	10000	500	15506.32	3090.41	**2203.31**	2205.37	35320.91
	100000		6311.59	2932.69	**2193.83**	2209.59	10490.57
	5000000		**1078.00**	2235.99	2204.16	2211.45	2823,70
	10000000		**837.20**	2630.59	2334.78	2203.44	2635,36
Household Power Consuption	1	4	3492.87	1723.47	1565.75	**1549.63**	2571.31
	1.5		2255.28	2204.91	**1538.05**	1570.70	1858.18
	2		**1218.45**	1941.72	1542.86	1562.42	1894.59
	3		**669.38**	2096.28	1536.86	1559.74	1806.60
US CENSUS 1990	10	10	**2818.32**	-	-	-	5462.16
	15		**481.70**	-	-	-	2551.33
	25		**138.00**	-	-	-	2018.05
	50		**114.43**	-	-	-	1772.27

5

Conclusão

Este trabalho propôs um método para encontrar elementos representativos de grandes conjuntos de dados com a finalidade de obter uma aproximação do resultado da execução do algoritmo DBSCAN conjunto de dados completo. O método proposto foi comparado experimentalmente com outros métodos do estado da arte da literatura científica dessa área de pesquisa.

O BIRCHSCAN é baseado no método de análise de agrupamento BIRCH capaz de trabalhar de modo eficiente com grandes conjuntos de dados. O algoritmo realiza uma única varredura no conjunto de dados construindo uma estrutura de dados que o resume. O BIRCH possibilita realizar operações de agrupamento baseadas em elementos próximos que representam um subconjunto de elementos em vez de apenas um ponto.

Baseados na ideia de reprodutibilidade, Rauber et al. (2020) afirma que uma pesquisa de boa qualidade deve fornecer uma descrição detalhada dos experimentos, permitindo que a comunidade científica reproduza os resultados. Atendendo a este conceito, exceto para os códigos-fonte de DIDES, DENDIS e ProTraS (que devem ser solicitados aos seus autores), os conjuntos de dados, códigos-fonte, amostras e agrupamentos gerados por todos os métodos apresentados nos experimentos, juntamente com suas respectivas documentações, estão disponíveis em http://bit.ly/3q2iFw3.

5.1 Contribuições

Existem poucos métodos amostrais na literatura que foram desenvolvidos especificamente para funcionar em conjunto com o DBSCAN, uma das principais razões corresponde a dificuldade de se gerar uma amostra no qual os parâmetros ε e $minPTS$ do DBSCAN sejam bons tanto para o algoritmo amostral quanto para o método em si. A principal contribuição deste trabalho é o desenvolvimento de um método com capacidade de gerar um conjunto de elementos representativos que sejam bons para o DBSCAN utilizando

os mesmos parâmetros ε e $minPTS$ no algoritmo e que permita encontrar um resultado aproximado do DBSCAN aplicado no conjunto de dados inteiro com um tempo de execução melhor.

Este trabalho propôs um método que coleta amostras(elementos representativos) de grandes conjuntos de dados para obter uma solução de agrupamento por aproximação para o algoritmo DBSCAN. O método proposto foi comparado experimentalmente com métodos de amostragem no estado da arte. Por meio da experimentação, foi possível identificar que nenhum dos métodos superou os demais em todos os conjuntos de dados investigados. Porém, o algoritmo proposto (BIRCHSCAN) apresenta os melhores resultados em geral, tendo menos oscilação e uma diferença estatisticamente significativa em relação a quase todos os concorrentes. No caso do algoritmo para o qual não houve diferença estatisticamente significativa no desempenho, o método proposto obteve melhor desempenho e menor variabilidade em todos os experimentos, sendo um bom indicativo de sua superioridade.

Por ser um algoritmo mais complexo, esperava-se que a execução do algoritmo BIRCH (3.1) em pequenos conjuntos de dados fosse mais lenta do que o Leader* usado no algoritmo R*. A partir dos experimentos, observou-se que o BIRCH ainda é rápido ao trabalhar com grandes conjuntos de dados e fornece um conjunto de elementos representativos com resultados mais confiáveis, diferenciando-se também do Leader* em ser pouco sensível a forma que os dados são apresentados para o algoritmo.

É necessário destacar que os métodos DIDES, DENDIS e ProTraS não são propostos para trabalhar exclusivamente com o DBSCAN, dado que suas amostras são geradas independente dos parâmetros do DBSCAN. Portanto, apesar de seus autores terem realizado testes com o DBSCAN, as amostras geradas por estes métodos podem trabalhar em conjunto com outros métodos de análise de agrupamento, diferentemente do BIRCHSCAN que foi desenvolvido exclusivamente para trabalhar com o DBSCAN, dado que sua amostra utiliza os parâmetros ε e $minPTS$ do DBSCAN. Além disso, é preciso esclarecer que caso exista algum método de expansão dos grupos para o DIDES, DENDIS e ProTraS diferentemente do vizinho mais próximo, é possível que as comparações de tempo realizadas possam ser alteradas.

Uma fraqueza visível no método proposto está relacionado ao hiperparâmetro δ, sendo que este é um parâmetro de difícil estimativa e que pode comprometer a qualidade do resultado a ser obtido. Sabe-se que o valor utilizado nos experimentos mesmo que tenha apresentado resultados satisfatórios não é o ideal para qualquer conjunto de dados, pois a partir dos testes empíricos de definição do parâmetro, foi possível obter resultados melhores com valores específicos para determinados conjuntos de dados.

5.2 Trabalhos Futuros

A principal desvantagem da proposta deste trabalho diz respeito à definição do parâmetro δ, que, em conjunto com o parâmetro ε do DBSCAN, gera o valor de *Threshold* usado no BIRCH. Este parâmetro influencia diretamente o conjunto de elementos representantes encontrados pelo BIRCH, e não é definido intuitivamente ou com base no conjunto de dados. Neste trabalho, o valor 0,5 foi escolhido empiricamente. No entanto, pode ser possível encontrar uma transformação que defina a melhor escolha para *Threshold* dado um valor específico do parâmetro ε do DBSCAN. Outra possibilidade é realizar uma análise de sensibilidade para descobrir o melhor valor para este parâmetro em uma determinada aplicação. Um estudo inicial de sensibilidade poderá ser realizado testando diversos valores do parâmetro δ para conjuntos de dados específicos, possibilitando visualizar qual o impacto no tamanho de elementos representantes encontrado pelo BIRCH e na aproximação do resultado do DBSCAN. Este problema será estudado em trabalhos futuros.

Um estudo a ser investigado futuramente, consiste na realização de um novo experimento. O experimento será realizado gradualmente a partir de diversas execuções do DBSCAN e os métodos de agrupamento baseados em amostragem. As execuções iniciam-se sendo aplicadas em um pequeno subconjunto do conjunto de dados e posteriormente aumentando o tamannho do subconjunto até a execução do conjunto de dados completo. Cada execução validará tempo e qualidade de aproximação(se possível) do conjunto de dados. Um experimento desses permitirá explorar informações de até quando é interessante utilizar somente o DBSCAN.

Outro ponto a ser investigado em trabalhos futuros é a variação dos resultados no experimentos, o método proposto, assim como outros presentes no experimento, apresentou resultados superiores em alguns conjuntos de dados, porém, em outros obteve um desempenho mediano. Portanto, é interessante a realização de um estudo verificando se existe algum padrão nos conjuntos de dados que favoreça a aplicação de um ou outro método.

Por fim, estender a etapa experimental com novos conjuntos de dados, métodos e utilizando diferentes estratégias de avaliação fornecerá uma confiança ainda maior em relação aos resultados apresentados. A comparação do método proposto com outros métodos que fazem a otimização do DBSCAN para trabalhar com grandes conjuntos de dados, mas não utilizam técnicas de amostragem será um trabalho futuro a ser realizado.

Referências

AAKER, D.; KUMAR, V.; DAY, G. Pesquisa de marketing. são paulo: Atlas. eating in brazil. *Rev. Saude Publica*, v. 44, n. 2, p. 221–229, 2001.

AHMAD, A. U.; STARKEY, A. Application of feature selection methods for automated clustering analysis: a review on synthetic datasets. *Neural Computing and Applications*, Springer, v. 29, n. 7, p. 317–328, 2018.

AL-MAMORY, S.; KAMIL, I. Enhancing of dbscan based on sampling and densitybased separation. *Iraqi Journal for Computers and Informatics*, v. 42, n. 1, p. 38–47, 2016.

ALBUQUERQUE, M. A. d. Estabilidade em análise de agrupamento (cluster analysis). *Biometry Masters Thesis. Recife: Universidade Federal Rural de Pernambuco*, 2005.

APPEL, A. P. et al. Biased box sampling-a density-biased sampling for clustering. In: *Proceedings of the 2007 ACM symposium on Applied computing*. [S.l.: s.n.], 2007. p. 445–446.

APPEL, A. P. et al. A density-biased sampling technique to improve cluster representativeness. In: SPRINGER. *European Conference on Principles of Data Mining and Knowledge Discovery*. [S.l.], 2007. p. 366–373.

APPEL, A. P. e. a. A density-biased sampling technique to improve cluster representativeness. *European Conference on Principles of Data Mining and Knowledge Discovery. Springer, Berlin, Heidelberg*, p. 366–373, 2007.

ARLIA, D.; COPPOLA, M. Experiments in parallel clustering with dbscan. In: SPRINGER. *European Conference on Parallel Processing*. [S.l.], 2001. p. 326–331.

ARTHUR, D.; VASSILVITSKII, S. How slow is the k-means method? In: *Proceedings of the twenty-second annual symposium on Computational geometry*. [S.l.: s.n.], 2006. p. 144–153.

BEJARANO, J. et al. Sampling within k-means algorithm to cluster large datasets. *UMBC Student Collection*, 2011.

CARDOSO, V. G.; BARROS, E.; BARBOSA, J. A. Porosity features extraction based on image segmentation technique applying k-means clustering algorithm. *Rio Oil & Gas*, 2020.

CHAUDHURI, S.; MOTWANI, R.; NARASAYYA, V. On random sampling over joins. *ACM SIGMOD Record*, ACM New York, NY, USA, v. 28, n. 2, p. 263–274, 1999.

CHÁVEZ, E. et al. Searching in metric spaces. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 33, n. 3, p. 273–321, 2001.

CHEN, Y. et al. Block-dbscan: Fast clustering for large scale data. *Pattern Recognition*, Elsevier, v. 109, p. 107624, 2020.

CHEN, Y. et al. Knn-block dbscan: Fast clustering for large-scale data. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, IEEE, 2019.

CHEN YEWANG, e. a. A fast clustering algorithm based on pruning unnecessary distance computations in dbscan for high-dimensional data. *Pattern Recognition 83*, p. 375–387, 2018.

CURISKIS, S. A. et al. An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit. *Information Processing & Management*, Elsevier, v. 57, n. 2, p. 102034, 2020.

DESGRAUPES, B. Clustering indices. *University of Paris Ouest-Lab Modal'X*, v. 1, p. 34, 2013.

DOLNICAR, S.; LEISCH, F. Segmenting markets by bagged clustering. *Australasian Marketing Journal (AMJ)*, Elsevier, v. 12, n. 1, p. 51–65, 2004.

EL-SONBATY, Y.; ISMAIL, M. A.; FAROUK, M. An efficient density based clustering algorithm for large databases. *16th IEEE international conference on tools with artificial intelligence. IEEE*, 2004.

EL-SONBATY, Y.; ISMAIL, M. A.; FAROUK, M. An efficient density based clustering algorithm for large databases. In: IEEE. *16th IEEE international conference on tools with artificial intelligence*. [S.l.], 2004. p. 673–677.

ESTER, M. e. a. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*, p. 226–231, 1996.

FAHAD, A. et al. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, IEEE, v. 2, n. 3, p. 267–279, 2014.

FELDMAN, D.; LANGBERG, M. A unified framework for approximating and clustering data. In: *Proceedings of the forty-third annual ACM symposium on Theory of computing*. [S.l.: s.n.], 2011. p. 569–578.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937.

FRIEDMAN, M. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, JSTOR, v. 11, n. 1, p. 86–92, 1940.

FURQON, M. T.; MUFLIKHAH, L. Clustering the potential risk of tsunami using density-based spatial clustering of application with noise (dbscan). *Journal of Environmental Engineering and Sustainable Technology*, v. 3, n. 1, p. 1–8, 2016.

GOLDSCHMIDT, R.; PASSOS, E.; BEZERRA, E. *Data Mining*. [S.l.]: Elsevier Brasil, 2015.

GOVENDER, P.; SIVAKUMAR, V. Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). *Atmospheric Pollution Research*, Elsevier, v. 11, n. 1, p. 40–56, 2020.

GUHA, S.; RASTOGI, R.; SHIM, K. Cure: An efficient clustering algorithm for large databases. *ACM Sigmod record*, ACM New York, NY, USA, v. 27, n. 2, p. 73–84, 1998.

HAIR, J. F. et al. *Análise multivariada de dados*. [S.l.]: Bookman editora, 2009.

HAN, J.; KAMBER, M.; PEI, J. Data mining: concepts and technologies. *Data Mining Concepts Models Methods & Algorithms,*, v. 5, n. 4, p. 1–18, 2001.

HARTIGAN, J. A. Clustering algorithms. *John Wiley & Sons, Inc.*, 1975.

HRUSCHKA, E. R.; EBECKEN, N. F. A genetic algorithm for cluster analysis. *Intelligent Data Analysis*, IOS Press, v. 7, n. 1, p. 15–25, 2003.

HUANG, Z. Clustering large data sets with mixed numeric and categorical values. In: CITESEER. *Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining,(PAKDD)*. [S.l.], 1997. p. 21–34.

HUBERT LAWRENCE; ARABIE, P. Comparing partitions. *Journal of classification*, v. 2, n. 1, p. 193–218, 1985.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. *ACM computing surveys (CSUR)*, Acm New York, NY, USA, v. 31, n. 3, p. 264–323, 1999.

JANG, J.; JIANG, H. Dbscan++: Towards fast and scalable density clustering. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2019. p. 3019–3029.

KERDPRASOP, K.; KERDPRASOP, N.; SATTAYATHAM, P. Density-biased clustering based on reservoir sampling. In: IEEE. *16th International Workshop on Database and Expert Systems Applications (DEXA'05)*. [S.l.], 2005. p. 1122–1126.

KOLLIOS, G. et al. Efficient biased sampling for approximate clustering and outlier detection in large data sets. *IEEE Transactions on knowledge and data engineering*, IEEE, v. 15, n. 5, p. 1170–1187, 2003.

KOLLIOS, G. et al. An efficient approximation scheme for data mining tasks. In: IEEE. *Proceedings 17th International Conference on Data Engineering*. [S.l.], 2001. p. 453–462.

KORN, F.; JOHNSON, T.; JAGADISH, H. Range selectivity estimation for continuous attributes. In: IEEE. *Proceedings. Eleventh International Conference on Scientific and Statistical Database Management*. [S.l.], 1999. p. 244–253.

KUMAR, K. M.; REDDY, A. R. M. A fast dbscan clustering algorithm by accelerating neighbor searching using groups method. *Pattern Recognition*, Elsevier, v. 58, p. 39–48, 2016.

KUO, F. Y.; SLOAN, I. H. Lifting the curse of dimensionality. *Notices of the AMS*, v. 52, n. 11, p. 1320–1328, 2005.

LI, H. et al. A novel density-based clustering algorithm using nearest neighbor graph. *Pattern Recognition*, Elsevier, v. 102, p. 107206, 2020.

LI, S.-S. An improved dbscan algorithm based on the neighbor similarity and fast nearest neighbor query. *IEEE Access*, IEEE, v. 8, p. 47468–47476, 2020.

LINDEN, R. Técnicas de agrupamento. *Revista de Sistemas de Informação da FSMA*, n, v. 4, n. 4, p. 18–36, 2009.

LUCHI, D.; RODRIGUES, A. L.; VAREJÃO, F. M. Sampling approaches for applying dbscan to large datasets. *Pattern Recognition Letters*, v. 117, p. 90–96, 2019.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. *In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, v. 1, n. 14, p. 281–297, 1967.

MATIAS, Y.; VITTER, J. S.; WANG, M. Wavelet-based histograms for selectivity estimation. In: *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*. [S.l.: s.n.], 1998. p. 448–459.

NANOPOULOS, A.; MANOLOPOULOS, Y.; THEODORIDIS, Y. An efficient and effective algorithm for density biased sampling. In: *Proceedings of the eleventh international conference on Information and knowledge management*. [S.l.: s.n.], 2002. p. 398–404.

NANOPOULOS, A.; THEODORIDIS, Y.; MANOLOPOULOS, Y. C2p: Clustering based on closest pairs. In: *VLDB*. [S.l.: s.n.], 2001. p. 331–340.

NANOPOULOS, A.; THEODORIDIS, Y.; MANOLOPOULOS, Y. Indexed-based density biased sampling for clustering applications. *Data & Knowledge Engineering*, Elsevier, v. 57, n. 1, p. 37–63, 2006.

NEMENYI, P. Distribution-free multiple comparisons (doctoral dissertation, princeton university, 1963). *Dissertation Abstracts International*, v. 25, n. 2, p. 1233, 1963.

NG, R. T.; HAN, J. Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, IEEE, v. 14, n. 5, p. 1003–1016, 2002.

NGOAN, P. V.; DUC, N. V. et al. A sample-based algorithm for visual assessment of cluster tendency (vat) with large datasets. In: SPRINGER. *International Conference on Future Data and Security Engineering*. [S.l.], 2018. p. 145–157.

NISA, K. K.; ANDRIANTO, H. A.; MARDHIYYAH, R. Hotspot clustering using dbscan algorithm and shiny web framework. In: IEEE. *2014 International Conference on Advanced Computer Science and Information System*. [S.l.], 2014. p. 129–132.

OHADI, N. et al. Sw-dbscan: A grid-based dbscan algorithm for large datasets. In: IEEE. *2020 6th International Conference on Web Research (ICWR)*. [S.l.], 2020. p. 139–145.

OTAIR, D. et al. Approximate k-nearest neighbour based spatial clustering using kd tree. *arXiv preprint arXiv:1303.1951*, 2013.

PACIFICO, L. D. S.; LUDERMIR, T. B. A group search optimization method for data clustering. In: IEEE. *2014 Brazilian Conference on Intelligent Systems*. [S.l.], 2014. p. 342–347.

PALMER, C. R.; FALOUTSOS, C. Density biased sampling: An improved method for data mining and clustering. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. [S.l.: s.n.], 2000. p. 82–92.

PATWARY, M. M. A. et al. A new scalable parallel dbscan algorithm using the disjoint-set data structure. In: IEEE. *SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. [S.l.], 2012. p. 1–11.

POOSALA, V.; IOANNIDIS, Y. E. Selectivity estimation without the attribute value independence assumption. In: *VLDB*. [S.l.: s.n.], 1997. v. 97, p. 486–495.

RAHMAH, N.; SITANGGANG, I. S. Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra. In: IOP PUBLISHING. *IOP Conference Series: Earth and Environmental Science*. [S.l.], 2016. v. 31, n. 1, p. 012012.

RAND, W. M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association 66.336*, p. 846–850, 1971.

RAUBER, T. W. et al. An experimental methodology to evaluate machine learning methods for fault diagnosis based on vibration signals. *Expert Systems with Applications*, Elsevier, p. 114022, 2020.

ROS FRÉDÉRIC; GUILLAUME, S. Dendis: A new density-based sampling for clustering algorithm. *Expert Systems with Applications*, v. 56, p. 349–359, 2016.

ROS FRÉDÉRIC; GUILLAUME, S. Dides: a fast and effective sampling for clustering algorithm. *Knowledge and information systems*, v. 50, n. 2, p. 543–568, 2017.

ROS FRÉDÉRIC; GUILLAUME, S. Protras: a probabilistic traversing sampling algorithm. *Expert Systems with Applications*, v. 105, p. 65–76, 2018.

ROSENKRANTZ, D. J.; STEARNS, R. E.; LEWIS II, P. M. An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing*, SIAM, v. 6, n. 3, p. 563–581, 1977.

SAINT-PAUL, R.; RASCHIA, G.; MOUADDIB, N. General purpose database summarization. In: CITESEER. *Proceedings of the 31st international conference on Very large data bases*. [S.l.], 2005. p. 733–744.

SARMA, T. H.; VISWANATH, P. Speeding-up the k-means clustering method: A prototype based approach. In: SPRINGER. *International Conference on Pattern Recognition and Machine Intelligence*. [S.l.], 2009. p. 56–61.

SCHUBERT, E. et al. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, ACM New York, NY, USA, v. 42, n. 3, p. 1–21, 2017.

SCOTT, D. W. *Multivariate density estimation: theory, practice, and visualization*. [S.l.]: John Wiley & Sons, 2015.

SHEN, J. et al. Real-time superpixel segmentation by dbscan clustering algorithm. *IEEE transactions on image processing*, IEEE, v. 25, n. 12, p. 5933–5942, 2016.

SHERIDAN, K. et al. An application of dbscan clustering for flight anomaly detection during the approach phase. In: *AIAA Scitech 2020 Forum*. [S.l.: s.n.], 2020. p. 1851.

SIBSON, R. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, v. 16, n. 1, p. 30–34, 1973.

SILVERMAN, B. W. *Density estimation for statistics and data analysis*. [S.l.]: CRC press, 1986. v. 26.

TANG, W.; PI, D.; HE, Y. A density-based clustering algorithm with sampling for travel behavior analysis. In: SPRINGER. *International Conference on Intelligent Data Engineering and Automated Learning*. [S.l.], 2016. p. 231–239.

VELMURUGAN, T. Performance based analysis between k-means and fuzzy c-means clustering algorithms for connection oriented telecommunication data. *Applied Soft Computing*, Elsevier, v. 19, p. 134–146, 2014.

VISWANATH P.; BABU, V. S. Rough-dbscan: A fast hybrid density based clustering method for large data sets. *Pattern Recognition Letters*, v. 30, n. 16, p. 1477–1488, 2009.

VITTER, J. S.; WANG, M.; IYER, B. Data cube approximation and histograms via wavelets. In: *Proceedings of the seventh international conference on Information and knowledge management*. [S.l.: s.n.], 1998. p. 96–104.

WAN, L.; YANG, J. Advanced split birch algorithm in reconfigurable network. *J. Networks*, v. 8, p. 2050–2056, 2013.

WANG, L. et al. Approximate pairwise clustering for large data sets via sampling plus extension. *Pattern Recognition*, Elsevier, v. 44, n. 2, p. 222–235, 2011.

WANG, X.; HAMILTON, H. J. Dbrs: a density-based spatial clustering method with random sampling. *Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg*, 2003.

WANG, X.; HAMILTON, H. J. Dbrs: A density-based spatial clustering method with random sampling. In: SPRINGER. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. [S.l.], 2003. p. 563–575.

XIAO, Y. et al. A k-farthest-neighbor-based approach for support vector data description. *Applied intelligence*, Springer, v. 41, n. 1, p. 196–211, 2014.

XU, R.; WUNSCH, D. Survey of clustering algorithms. *IEEE Transactions on neural networks*, Ieee, v. 16, n. 3, p. 645–678, 2005.

YANG, M.-S.; WU, K.-L. A modified mountain clustering algorithm. *Pattern analysis and applications*, Springer, v. 8, n. 1-2, p. 125–138, 2005.

YUNOH, M. et al. K-means clustering analysis and artificial neural network classification of fatigue strain signals. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Springer, v. 39, n. 3, p. 757–764, 2017.

ZHANG et al. Machine learning approach for sequence clustering with applications to ground-motion selection. *Journal of Engineering Mechanics*, American Society of Civil Engineers, v. 146, n. 6, p. 04020040, 2020.

ZHANG, T.; RAMAKRISHNAN, R.; LIVNY, M. Birch: an efficient data clustering method for very large databases. *ACM Sigmod Record*, v. 25, n. 2, p. 103–114, 1996.

ZHAO, X.; LIANG, J.; DANG, C. A stratified sampling based clustering algorithm for large-scale data. *Knowledge-Based Systems*, Elsevier, v. 163, p. 416–428, 2019.

ZHOU, H.; WANG, P.; LI, H. Research on adaptive parameters determination in dbscan algorithm. *Journal of Xi'an University of Technology*, v. 28, n. 3, p. 289–292, 2012.

ZHOU, S. et al. Combining sampling technique with dbscan algorithm for clustering large spatial databases. In: SPRINGER. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. [S.l.], 2000. p. 169–172.