2022 IEEE 38ª Conferência Internacional de Engenharia de Dados (ICDE)

CORE-SG: computação eficiente de múltiplos MSTs para métodos baseados em densidade

Antonio Cavalcante Araujo Neto
Departamento de Ciência da Computação Universidade de Alberta
antonio.cavalcante@ualberta.ca

Murilo Coelho Naldi
Departamento de Ciência da Computação Universidade Federal de Sa˜o Carlos
naldi@ufscar.br

Ricardo J. G. B. Campello
School of Math. and Phys. Science Universidade de Newcastle
ricardo.campello@newcastle.edu.au

Jo¨rg Sander
Departamento de Ciência da Computação Universidade de Alberta
jsander@ualberta.ca

Resumo - *Vários* métodos populares baseados em densidade para tarefas de aprendizado não supervisionadas e semissupervisionadas, incluindo agrupamento e classificação, podem ser formulados como instâncias de uma estrutura que se baseia no processamento de uma árvore de extensão mínima dos dados, em que os pesos das bordas correspondem a uma forma de estimativa de densidade (não normalizada) *em relação a* um parâmetro de suavização $_{mpts}$. Embora os métodos baseados em densidade sejam considerados robustos *No que diz respeito à* $_{mpts,}$ no sentido de que pequenas alterações em seu valor geralmente levam a pequenas ou nenhuma alteração na estrutura resultante, intervalos mais amplos de valores de $_{mpts}$ podem levar a resultados diferentes que um usuário gostaria de analisar antes de escolher o valor mais adequado para um determinado conjunto de dados ou aplicação. No entanto, para explorar vários resultados para um intervalo de valores de $_{mpts}$, até recentemente, era necessário executar novamente o método baseado em densidade para cada valor no intervalo de forma independente, o que é computacionalmente ineficiente. Este artigo propõe uma nova abordagem computacionalmente eficiente para calcular várias árvores de abrangência mínima baseadas em densidade com *relação a* um conjunto de valores de $_{mpts}$, aproveitando um gráfico obtido de uma única execução do algoritmo baseado em densidade, sem a necessidade de novas execuções do algoritmo original. Apresentamos resultados teóricos e experimentais que mostram que nossa abordagem supera as desvantagens do estado da arte anterior e é consideravelmente superior em termos de tempo de execução e tamanho do gráfico, além de ser mais fácil de implementar. Nossa avaliação experimental usando dados sintéticos e reais mostra que nossa estratégia pode levar a fatores de aceleração de centenas a milhares de vezes no cálculo de árvores de abrangência mínima baseadas em densidade.

***Palavras-chave* - métodos *baseados em* densidade, k-vizinhos mais próximos, mini-mum spanning tree, clustering, clustering semissupervisionado**

I. Introdução

Os métodos baseados em densidade para aprendizado não supervisionado e semissupervisionado são opções populares para análise exploratória de dados em muitos cenários de aplicação [1]-[6]. Eles podem capturar regiões contíguas em um espaço de dados em que a densidade de pontos não cai abaixo de um determinado limite, podendo, assim, por exemplo, decompor um conjunto de dados em grupos de pontos de dados que ajudam a entender a estrutura subjacente dos dados, exigindo pouco ou nenhum conhecimento prévio [7]. Na configuração não supervisionada, os métodos de agrupamento baseados em densidade, como DBSCAN [8], OPTICS [9] e HDBSCAN* [10], [11], são bem conhecidos por sua capacidade de encontrar agrupamentos baseados em densidade e discriminar agrupamentos de ruídos, além de serem estatisticamente sólidos. Esses algoritmos foram aplicados com sucesso em uma variedade de

DOI 10.1109/ICDE53745.2022.00076

campos, como robótica [12], genética [13], comportamento humano [14], química [15], astronomia e astrofísica [16], entre muitos outros. Quando uma pequena porção de objetos de dados rotulados (ou seja, informações de verdade) está disponível, os métodos de agrupamento semissupervisionado e de classificação semissupervisionada podem usar essas informações de forma eficaz e levá-las em conta ao decompor o conjunto de dados em grupos (clusters ou classes), considerando também a densidade subjacente dos dados [17].

Motivação. Gertrudes *et al.* [18] mostraram que uma variedade de algoritmos de agrupamento não supervisionados e semissupervisionados (incluindo o HDBSCAN*), bem como algoritmos de classificação semissupervisionados, podem ser descritos a partir de uma perspectiva baseada em densidade como instâncias ou derivações de uma estrutura unificada que tem como núcleo comum o processamento (direto ou indireto) de uma *Minimum Spanning Tree* (*MST*) dos dados em um espaço de dados transformado. Os pesos das bordas da *MST* nesse espaço correspondem a estimativas da densidade que conecta os pares correspondentes de objetos de dados. Normalmente, a *MST* é obtida calculando-se *dinamicamente* os pesos das bordas para cada par de pontos de dados, representando vértices adjacentes de um gráfico conceitual e completo dos dados que não precisa ser materializado. Os pesos das bordas dependem de um único parâmetro definido pelo usuário, m_{pts} , que atua como um fator de suavização clássico para estimativas de densidade não paramétricas. Embora relativamente robusta, a definição desse parâmetro pode ser desafiadora na prática porque: (a) nos cenários de aprendizado não supervisionado e semissupervisionado, no máximo uma pequena parte dos dados rotulados (se houver) está disponível, o que dificulta ou impossibilita a aplicação de abordagens de seleção de modelos supervisionados, como a validação cruzada; e (b) valores muito diferentes de m_{pts} podem levar a resultados muito diferentes, e um usuário normalmente não sabe um valor adequado com antecedência. Além disso, é bastante comum em tarefas de mineração de dados gastar um tempo considerável explorando valores de parâmetros. Os autores em [6], [19] apresentaram uma coleção de exemplos e visualizações que ilustram casos em que diferentes intervalos de valores de m_{pts} produzem resultados diferentes. Uma possível abordagem para contornar a escolha de um valor específico e potencialmente inadequado de m_{pts} consiste em realizar uma análise exploratória de vários resultados baseados em densidade. Essa

2375-026X/22/$31.00 ©2022 IEEE

2022 IEEE 38ª Conferência Internacional de Engenharia de Dados (ICDE) | 978-1-6654-0883-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDE53745.2022.00076

DOI 10.1109/ICDE53745.2022.00076

A abordagem de *MST* requer a construção do "*MST* baseado em densidade" (que pode ser processado posteriormente por cada algoritmo que se baseia nesse gráfico) várias vezes para diversos valores de m_{pts} , permitindo uma análise completa e posterior dos resultados. Para ser eficaz, seria necessário considerar um conjunto ou um intervalo de valores de m_{pts} para garantir que os valores adequados sejam incluídos e também para permitir uma exploração mais aprofundada dos dados. No entanto, o custo computacional dessa abordagem pode ser proibitivo, especialmente para grandes conjuntos de dados, porque exige o cálculo de vários *MSTs* de um *gráfico completo*, um para cada valor de m_{pts} . Embora esse gráfico completo não precise ser explicitamente materializado e armazenado na memória, suas bordas *de* $O(n^2)$ (para n objetos de dados) precisarão ser processadas dinamicamente. Essas bordas são ponderadas de acordo com o parâmetro de entrada m_{pts} , e essa dependência exige o recálculo dos pesos das bordas e, consequentemente, o recálculo do *MST* para cada valor diferente de m_{pts} . Esses recálculos podem ser acelerados com a reutilização de informações da execução anterior, ao custo de manter os dados sobre os m_{pts} vizinhos mais próximos de cada ponto de dados, bem como o gráfico completo na memória. No entanto, esses requisitos de uso de memória podem se tornar rapidamente proibitivos em aplicativos práticos com o aumento do tamanho do conjunto de dados.

Para reduzir os requisitos computacionais de reutilização do gráfico completo ao encontrar *MSTs* adicionais baseados em densidade, é possível usar um gráfico menor. Os autores em [19], [20] propuseram um método para calcular com eficiência *MSTs* baseados em densidade para toda uma gama de valores de m_{pts} seguindo essa abordagem. Especificamente, em vez de recorrer ao gráfico completo com pesos de borda diferentes para cada valor de m_{pts} , eles propuseram pré-computar um *gráfico de vizinhança relativa* (*Relative Neighborhood Graph, RNG*). Esse gráfico não contém bordas do gráfico completo que garantidamente não fazem parte de nenhuma *MST* para o intervalo determinado de valores de m_{pts} . Assim, todas as *MSTs* adicionais podem ser extraídas do único *RNG de* forma eficiente, já que o *RNG* normalmente tem muito menos bordas do que o gráfico completo. Embora a abordagem baseada *em RNG tenha* se mostrado útil em cenários práticos, ela tem deficiências que podem limitar sua eficiência ou até mesmo impedir sua aplicação, incluindo (i) o *RNG* acabará se aproximando do gráfico completo em termos de número de arestas à medida que a dimensionalidade dos dados aumenta, o que gera uma sobrecarga de pré-computação e armazenamento de um *RNG* grande para um ganho cada vez menor no desempenho geral do cálculo de *MSTs*; (ii) o *RNG* se baseia em propriedades geométricas e, para ser eficiente, uma implementação deve se basear em suposições que restringem sua aplicabilidade, ou seja, os dados devem ser vetores de características com valor real e uma métrica deve ser usada como medida de distância. Essas suposições impedem a aplicação do método em cenários em que os dados não podem ser descritos como um conjunto de pontos em um sistema de coordenadas, bem como em cenários que exigem medidas de distância que não satisfazem a desigualdade triangular. **Contribuições. Neste** artigo, descrevemos descobertas originais que superam essas deficiências e propomos uma solução nova, muito mais geral e, ao mesmo tempo, mais eficiente para o cálculo de *MSTs* baseadas em densidade múltipla para

diferentes valores de m_{pts} . Surpreendentemente, essa solução depende apenas de informações que podem ser mostradas como disponíveis ao calcular um único *MST* baseado em densidade, correspondente ao maior valor de m_{pts} em um determinado intervalo de valores candidatos. Ela evita todas as limitações e complexidades envolvidas na construção de um *RNG* e, o que é mais importante, leva a um subgráfico do *RNG*, normalmente com um número menor de bordas. Esse número reduzido de bordas se traduz em uma maneira ainda mais eficiente de computar vários resultados de algoritmos baseados em densidade não supervisionados e semissupervisionados *em relação a* um conjunto de m_{pts} valores, levando a fatores de aceleração que variam de centenas a milhares.

Em particular, mostramos que várias soluções baseadas em densidade para uma variedade de valores de m_{pts} podem ser extraídas exatamente da união de dois gráficos: (1) o *MST* do gráfico completo no espaço transformado de estimativas de densidade em que os pesos das bordas são calculados *em relação* ao maior valor candidato de m_{pts} (definido aqui como k_{max}) e (2) o *k-Nearest Neighbor Graph* (*k-NN G*), que conecta cada objeto em um conjunto de dados com seus *k* vizinhos mais próximos no espaço de dados original de acordo com uma determinada medida de dissimilaridade/distância. Também provamos formalmente que o gráfico resultante, *CORE-SG*, é um subgráfico do *RNG*. As principais implicações desses resultados são que:

1) há um limite superior linear do número de bordas no gráfico pré-computado *em relação ao* número de objetos de dados, *n* (supondo que $k_{max} << n$, como esperado em aplicações práticas). Essa é uma grande melhoria em relação ao limite superior quadrático do gráfico completo ou do *RNG*;
2) O gráfico pode ser obtido a partir de informações prontamente disponíveis após uma única execução de um algoritmo baseado em densidade que se baseia em um *MST com* m_{pts} (por exemplo, HDBSCAN*), simplesmente definindo $m_{pts} = k_{max}$;
3) nossa abordagem não faz suposições restritivas sobre a representação de dados ou sobre a medida de distância adotada, ao contrário da abordagem *baseada* em *RNG*;
4) é possível aproveitar as propriedades do *CORE-SG* para executar uma etapa de otimização opcional e obter o gráfico de substituição mínimo possível.

O restante deste documento está organizado da seguinte forma. A Seção II discute os trabalhos relacionados. A Seção III descreve nossas descobertas teóricas e comprova sua exatidão. Na Seção IV, descrevemos como essas descobertas podem ser usadas para resolver com mais eficiência o problema de interesse deste artigo. A Seção V apresenta e discute nossa avaliação experimental. Por fim, a Seção VI conclui o artigo e fornece orientações para trabalhos futuros.

II. Trabalho relacionado

Há vários trabalhos na literatura que visam acelerar/escalar os algoritmos de aprendizado baseados em densidade, especialmente para grandes conjuntos de dados [21]-[26]. Os métodos existentes recorrem, por exemplo, a informações espaciais, estruturas de dados, amostragem e/ou técnicas paralelas e distribuídas para calcular com eficiência um único modelo baseado em densidade. A contribuição do nosso trabalho tem um foco diferente, a saber: calcular com eficiência modelos *adicionais* baseados em densidade para uma variedade de valores de m_{pts} a partir de um modelo inicial baseado em densidade. O modelo inicial pode ser obtido usando qualquer algoritmo existente para calcular um modelo baseado em densidade.

MST do gráfico completo *com relação a* um único valor m_{pts} . Portanto, nosso método não é uma alternativa às abordagens de aprendizado baseadas em densidade da literatura, mas sim um complemento a elas.

O trabalho mais relacionado ao nosso é [6], [19], [20], em que os autores substituem o gráfico completo no HDBSCAN* por um gráfico tipicamente muito menor, o *RNG*, para acelerar o cálculo de várias hierarquias de agrupamento baseadas em densidade para uma gama de valores do parâmetro m_{pts} . Foi demonstrado que a estratégia *baseada em RNG* pode ser mais de 60 vezes mais rápida do que uma abordagem ingênua em que o HDBSCAN* é executado para cada valor de m_{pts} individualmente. Visualizações adequadas para análise exploratória também foram desenvolvidas em [6], [19], que também podem ser prontamente usadas em nosso trabalho atual. Na Seção IV-D, discutimos com mais detalhes as novidades e contribuições da nossa proposta atual em comparação com a abordagem *baseada* em *RNG*.

A necessidade de produzir eficientemente resultados de aprendizado não supervisionado em vários perfis de densidade motivou recentemente o trabalho em [5], que propôs uma abordagem baseada em índice para acelerar a execução do SCAN [27] (uma adaptação do DBSCAN para *gráficos*, ou seja, *agrupamento de redes*) para diferentes valores de seus parâmetros de controle de densidade, μ e ε. O índice proposto pode ser reutilizado para executar o SCAN para diferentes valores de μ e ε com economia computacional significativa. Essa abordagem é semelhante em intenção ao nosso trabalho, pois propõe uma estrutura de dados que permite executar novamente e com eficiência algoritmos para aprendizado não supervisionado (nesse caso, um único algoritmo) para vários valores de parâmetros. No entanto, o método é muito específico para encontrar componentes estruturalmente conectados em redes e não é aplicável a tarefas de agrupamento mais gerais ou a outras tarefas baseadas em densidade.

Os autores em [28] propuseram um procedimento de agrupamento com base na união da *MST* euclidiana e um gráfico de k vizinhos *mais próximos*, *k-NN G*, e estudaram a influência do parâmetro *k* para encontrar o menor valor de *k*, $\tilde{k}$, de modo que o *k-NNG* contenha a *MST*, a fim de identificar agrupamentos. Eles argumentam que o *MST* é incluído (como um subgrafo) no *k-NNG* para $\tilde{k} = O(\log n)$ quando não há estrutura de agrupamento nos dados, mas isso só ocorrerá para valores muito mais altos de *k* se houver dois ou mais agrupamentos bem separados. Da mesma forma, para um conjunto de dados com *n* objetos, a remoção de bordas do Euclidean *MST* que não estão contidas no *k-NN G*, para $k = O(\log n)$, pode levar à identificação de clusters nos dados como os componentes conectados resultantes. Com base nessa suposição, diversas variações de um algoritmo de agrupamento "*MST-kNN*" foram propostas na literatura (por *exemplo*, [29], [30]), em que os dados são representados por um gráfico ponderado completo com pesos de borda dados pelas semelhanças entre os vértices (objetos de dados). Esses métodos inicialmente particionam os dados usando a heurística acima para remover determinadas bordas do *MST* euclidiano e, em seguida, tentam refinar a solução usando diferentes estratégias para obter

uma partição final "plana" dos dados em clusters. Embora essas abordagens também usem uma forma de *MST* e um *k-NN G*, das soluções SL, principalmente o *efeito de encadeamento* e a alta sensibilidade ao ruído [32]. Por outro lado, propomos um método que permite o cálculo eficiente de uma série de *MSTs* em um gráfico diferente, definido em um espaço transformado que representa várias estruturas baseadas em densidade dos dados (não é necessário que seja euclidiano), e essas *MSTs* podem ser transformadas em soluções de agrupamento hierárquicas ou planas que não sofrem com as deficiências do SL mencionadas anteriormente.

Em [33], os autores combinam conceitos de clustering baseados em densidade, conforme definido pelo DBSCAN, com clustering espectral para encontrar clusters de alta densidade. A ideia geral consiste em minimizar um corte de gráfico para que a densidade média dos clusters obtidos seja maximizada, o que resulta em clusters semelhantes aos componentes conectados de *pontos centrais*, conforme definido pelo DBSCAN. No agrupamento espectral, a dimensionalidade dos dados é reduzida por meio do espectro da matriz de similaridade de um gráfico adequado, representando os dados. Um gráfico que desempenha essa função é o *k-NNG* [34]. Veenstra *et al.* [35] mostram que a qualidade da partição resultante pode ser afetada negativamente por determinados valores de *k*, uma vez que o *k-NNG* pode não ser um gráfico conectado para esses valores. Para garantir a conectividade, os autores propõem combinar o *k-NNG* com o Euclidean *MST* . Embora essas abordagens também usem uma forma de *MST* e um *k-NNG* para melhorar o agrupamento, elas são ortogonais às nossas. Ao representar nossos gráficos como matrizes de similaridade, também é possível combinar nosso método com o agrupamento espectral para explorar várias soluções de agrupamento espectral para uma variedade de gráficos conectados que representam a estrutura baseada em densidade dos dados.

III. Fundamentação Teórica

A estrutura unificada de métodos baseados em densidade para agrupamento e classificação semissupervisionados em [18] formaliza uma relação entre os métodos de agrupamento baseados em densidade e a abordagem baseada em gráficos para classificação transdutiva, mostrando que vários algoritmos populares podem ser interpretados (e implementados) como sendo direta ou indiretamente um *MST* calculado no espaço de *distâncias de acessibilidade mútua com um parâmetro de suavização m . Esse MST representa implicitamente uma organização hierárquica baseada em densidade dos dados, que pode ser influenciada pelo valor de m usado para estimar a densidade nos pontos de dados.*$_{pts}$Esse *MST* representa implicitamente uma organização hierárquica baseada em densidade dos dados, que pode ser influenciada pelo valor de m_{pts} usado para estimar a densidade nos pontos de dados. Por esse motivo, os usuários geralmente precisam explorar várias configurações de parâmetros para obter uma compreensão mais profunda dos dados e selecionar valores adequados de m_{pts} para uma determinada aplicação. Como a estrutura proposta em [18] separou a construção desse tipo de *MST* baseado em densidade dos algoritmos reais, o trabalho apresentado aqui pode ser aplicado a todos os algoritmos baseados nessa estrutura.

elas

Uma organ

nização hierárquica baseada em densidade (*com m*

conjunto de dados **X** com

pontos

)

de

um

simplesmente introduzem procedimentos heurísticos para remover determinadas bordas de um *MST* euclidiano a fim de obter uma partição plana dos dados. Lembrando que *os MSTs euclidianos* são equivalentes ao agrupamento de ligação única (SL) [31], essas heurísticas resultarão em partições planas que sofrem das conhecidas deficiências

n objetos é baseado nos seguintes conceitos:

- **A distância do núcleo $\mathbf{c_{m_{pts}}}$ (-).** A distância do núcleo $c_{m_{pts}}(p)$, para cada ponto $p \in \mathbf{X}$, é o menor raio ε em torno de p que cobre m_{pts} outros pontos, *ou seja*, dada uma função de distância subjacente $d(\cdot, \cdot)$, é a distância entre p e seu

mpts-nearest neighbor[1] : $c_{m\ pontos}(p) = d(p, m_{pts}\text{-}NN(p))$.

A distância do núcleo de um ponto *p* pode ser vista como o inverso de uma estimativa de densidade não normalizada no ponto *p* [11].

- **A distância de acessibilidade mútua $\mathbf{mrd_{m_{pts}}}$ (-, -).** A distância de alcançabilidade mútua, $mrd_{m_{pts}}(p, q)$, entre um par de pontos de dados *p*, *q* (w.*r.t.* m_{pts}) é definida como:

$$mrd_{m_{pts}}(p, q) = max\{c_{m_{pts}}(p), c_{m_{pts}}(q), d(p, q)\} \quad (1)$$

Observe que $mrd_{m_{pts}}(p, q)$ é a menor distância em na qual ambos os pontos, *p* e *q*, estão na vizinhança ε um do outro, e ambas as *vizinhanças* ε contêm pelo menos *m* $pontos_{pts}$. Essa distância pode ser vista como o inverso de uma densidade não normalizada (estimativa), representando a densidade máxima α (em relação a uma densidade *h*) na qual ambos os pontos *p* e *q*, e qualquer posição ao longo do caminho direto entre *p* e *q*, garantiram uma estimativa de densidade correspondente a α no conjunto de dados fornecido (com base somente nos pontos *p*, *q* e seus vizinhos mais *próximos mpts*).

Conceitualmente, o processo de obtenção de uma hierarquia baseada em densidade *com relação a* m_{pts} consiste em pegar o gráfico *G* completo e não ponderado do conjunto de dados, incorporando a acessibilidade mútua distâncias ($mrd_{m_{pontos}}$) como pesos de borda para obter um graph) Para $mrd_{m_{pts}}$ e, em seguida, calcular o *MST* desse gráfico. para encontrar as distâncias de alcançabilidade mútua, os valores de distância central para todos os $p \in \mathbf{X}$ devem ser computados, o que implica executar uma consulta *k- Nearest-Neighbor* para cada ponto *p*, com $k = m_{pts}$ para obter os k-vizinhos *mais próximos* de *p*. Assim, computar os k-vizinhos mais próximos faz parte do processo de encontrar a árvore de abrangência mínima no espaço de distâncias de alcançabilidade.

Quando uma coleção de *MSTs* (codificando várias hierarquias baseadas em densidade) para um conjunto de m_{pts} valores é necessária, re A repetição desse processo para cada valor de m_{pts} nesse conjunto exigem o recálculo dos pesos de borda $O(n^2)$ de $G_{m\ pts}$ por primeiro executando uma consulta *k-Nearest-Neighbor* (*k-NN*) para cada ponto no conjunto de dados com *k* igual ao valor atual de m_{pts} (para determinar todas as distâncias centrais) e, em seguida, calculando as distâncias de acessibilidade para todos os pares de pontos. Para acelerar o cálculo de *MSTs* baseados em densidade múltipla, pode-se, como primeira medida, evitar a reexecução da consulta *k-Nearest-Neighbor* para cada valor de m_{pts} pré-computando e reutilizando as consultas k_{max} *-NN*, uma vez que as informações *k-NN em relação a* todos os $k \leq k_{max}$ estão contidas no k_{max} *-NN* .

O uso das consultas k_{max} *-NN* para obter todas as distâncias de núcleo necessárias, no entanto, não resolve o alto custo computacional do cálculo de um número possivelmente grande de *MSTs* em gráficos completos $G_{m_{pts}}$ para os diferentes valores de m_{pts} . O custo computacional da criação de um *MST* aumenta com o número de arestas no gráfico do qual é extraído. Como $G_{m_{pts}}$ tem $\Theta(n^2)$ bordas, substituí-lo por um gráfico menor aceleraria os cálculos *do MST* e, consequentemente, reduziria o custo computacional. foi construído para conter todas as bordas necessárias para calcular o custo computacional geral da obtenção de várias hierarquias baseadas em densidade. Isso foi feito pela primeira vez em [19], substituindo o G_m no número de bordas, que é $O(n^2)$.

Neste trabalho, mostramos que há um gráfico melhor que pode substituir $G_{m_{pts}}$ para obter as hierarquias para cada valor de $m_{pts} \in [1, k_{max}]$. Esse gráfico tem $O(n)$ arestas, supondo $k_{max} << n$, o que é uma suposição realista na prática, e sua construção requer o cálculo dinâmico de um único *MST* baseado em densidade *com* $m_{pts} = k_{max}$, além de um k-gráfico de vizinhos mais próximos a partir das consultas *k-NN* para $k = k_{max}$. Do ponto de vista da geometria computacional, as consultas *k-NN* podem ser usadas para construir um gráfico *k-NNG* aumentado = (E_k, V_k) com $V_k = \{p | p \in X\}$ e $E_k = \{(p, q) | p \in \mathbf{X} \wedge q = i\text{-}NN(p)\ \forall i \in [1, k]\}$. Quando *k-NN* (*p*) não é único, o gráfico é aumentado com todos os pontos empatados como o k-ésimo vizinho. *Os* Gs *k-NN* têm o seguinte propriedades importantes, que podem ser usadas para acelerar a computação de *MSTs* baseados em densidade múltipla:

- Dadas duas *k-NNGs*, *i-NNG* e *j-NNG* com $i < j$, é verdade que $i\text{-}NNG \subseteq j\text{-}NN\ G$. De fato, a *j-NNG* pode ser obtida aumentando a *i-NNG* com bordas entre cada ponto *p* e seus vizinhos $(i + 1)$ a *j*.
- Para um determinado valor de $k = m_{pts}$, todas as bordas do *k-NNG* têm pesos determinados por distâncias de núcleo no gráfico completo $G_{m_{pts}}$ (Lema 1) e todas as bordas em $G_{m_{pts}}$ que têm pesos determinados por distâncias de núcleo também estão contidas no *k-NNG* (Lema 2).

Lema 1. $(p, q) \in k\text{-}NNG \Rightarrow mrd_k(p, q) = max\{c_k(p), c_k(q)\}$

Prova:

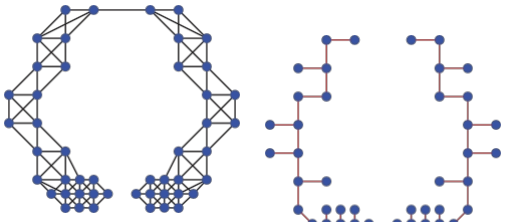
Considere uma borda (p, q) no k-NN G. Com base na definição do k-NN G, pelo menos um dos pontos p e q deve estar entre os k vizinhos mais próximos do outro ponto. Portanto, sabemos que $d(p, q) \leq c_k(p)$ ou $d(p, q) \leq c_k(q)$, ou ambos. Dada a definição de distância de acessibilidade mútua na Equação 1, como o valor $d(p, q)$ é menor ou, no máximo, igual a pelo menos uma das *distâncias centrais* de p e q, segue-se que $mrd_k(p, q) = max\{c_k(p), c_k(q)\}$.

Lema 2. $\forall m_{pts} \leq k$: $mrd_{m_{pts}}(p, q) = max\{c_{m_{pts}}(p), c_{m_{pts}}(q)\} \Rightarrow (p, q) \in k\text{-}NNG$

Prova: Vamos supor que $mrd_{m_{pts}}(p, q) = max\{c_{m_{pts}}(p), c_{m_{pts}}(q)\}$. Dada a definição da distância de acessibilidade mútua na Equação 1, segue-se que $c_{m_{pts}}(p) \geq d(p, q)$ ou $c_{m_{pts}}(q) \geq d(p, q)$, ou ambos. Portanto, em
Pelo menos um desses pontos deve estar entre os m_{pts} -nearest vizinhos do outro, *ou seja*, $q = i\text{-}NN(p)$ e/ou $p = i\text{-}NN(q)$ para algum $i \in [1, m_{pts}]$, o que implica que $(p, q) \in m_{pts}$ - NN G.

com um gráfico de vizinhança relativo especial (RNG), que

[1]O $mpts\text{-}NN(p)$ pode não ser único. Entretanto, a distância do núcleo não é afetada por empates no $mpts\text{-}NN(p)$; todos os pontos empatados têm a mesma distância até p.



(a) *k-NN G*, k=4 (b) $MST_{m_{pts}=4}$ com $m_{pontos}=4_{pts}$

Fig. 1: Exemplo de um *k-NNG* em (a) que não contém o *MST* em (b) para um determinado $G_{m_{pts}}$, em que $k = m_{pts} = 4$.

Como $m_{pts} \leq k$, segue-se que m_{pts} *-NNG* $\subseteq$ *k-NNG* e, consequentemente, $(p, q) \in$ *k-NN G*. ■

O Lema 2 implica que, para qualquer valor de $m_{pts} \in [1, k_{max}]$, todas as bordas que estão em um *MST* de $G_{m_{pts}}$ (denominado $MST_{m_{pts}}$) e cujos pesos são determinados por uma distância de núcleo podem ser obtidas de k_{max} *-NNG* em vez de serem obtidas do gráfico $G_{m_{pontos}}$. Todas as outras arestas de um MST_m (que são não determinados por uma distância de núcleo), por sua vez, não são garantidos para estar contido em k_{max} *-NN G*. Em geral, para um determinado $k = m_{pontos}$ o *k-NNG* não contém um *MST* de $G_{m_{pts}}$, mesmo que *k-NNG* seja um grafo conectado - e é por isso que todos os

Os métodos baseados em densidade, além de calcular os k-vizinhos mais *próximos*, também devem calcular explicitamente um *MST* sobre *G* .

Como exemplo, a Figura 1 ilustra que um *k-NNG* conectado por si só pode não ser suficiente para encontrar *MSTs* de $G_{m_{pts}}$.

Todas as bordas de uma de $G_{m_{pontos}}$ que não são com garantia de estar contido em k_{max} *-NNG* (ou seja, aqueles que não são determinados por uma distância de núcleo) podem ser demonstrados como necessariamente contidos no $MST_{k_{max}}$, porém, onde *o* $MST_{k_{max}}$ é uma *MST* de G_k (com $m_{pts} = k_{max}$). Juntamente com com o Lema 2, isso significa que $MST_{m_{pts}}$ deve ser contido na união de k_{max} *-NNG* e $MST_{k_{max}}$ para qualquer valor de $m_{pts} \in [1, k_{max}]$, conforme formalizado no Teorema 1.

Teorema 1. *Seja* M_i *o conjunto de todos os MSTs possíveis de* G_i *, e seja* MST_k *qualquer MST de* G_k *, então* $\forall i < k : \exists MST_i \in M_i : MST_i \subseteq MST_k \cup k\text{-}NN\ G.$

Prova: Consideremos uma aresta (p, q) em um $MST^* \in M_i$ que conecta dois conjuntos de pontos *P* e *Q*. O peso de a borda (p, q) é definida por $mrd_{m_{pts}}(p, q)$, que é o máximo da distância subjacente entre *p* e *q* e suas distâncias centrais, conforme definido na Equação 1. Devido à propriedade de corte dos *MSTs*, (p, q) deve ter o menor peso entre todas as bordas que conectam os conjuntos *P* e *Q* em G_i . Há dois (caso contrário, (p, q) não poderia ser a borda com o menor peso conectando *P* e *Q* no MST^*) e (ii) $mrd_k(p, q) \geq mrd_k(p', q')$ (caso contrário, (p', q') não poderia ser a borda com o menor peso conectando *P* e *Q* no MST_k). Agora, observe que o peso de uma borda não pode diminuir quando o valor de m_{pts} aumenta. Ao aumentar o valor de m_{pts} , as distâncias de núcleo $c_{m_{pts}}(\cdot)$ só podem aumentar e as distâncias de base entre os pontos $d(\cdot, \cdot)$ não mudam. Consequentemente, as distâncias de acessibilidade mútua (pesos das bordas), definidas na Equação 1,

só pode permanecer o mesmo ou aumentar. Portanto, quando m_{pts} aumenta de *i* para *k* ($i < k$ conforme a suposição do teorema), uma das seguintes afirmações deve ser verdadeira:

(a) $mrd_i(p, q) < mrd_k(p, q)$
(b) $mrd_i(p, q) = mrd_k(p, q)$

Se (a) for verdadeira, como a distância subjacente (base) entre dois pontos não depende de m_{pts} , a distância de alcance mútuo $mrd_k(p, q)$ deve ser determinada pela distância central de *p* ou *q*, ou *seja*, $mrd_k(p, q) = max\{c_k(p), c_k(q)\}$. Do Lema 2 com $m_{pts} = k$, segue-se que $(p, q) \in k\text{-}NNG$

e, portanto, $(p, q) \in MST_k \cup k\text{-}NN\ G$. Se (b) for verdadeira Em vez disso, a partir de (i) e (ii) acima, temos $mrd_k(p', q') \leq mrd_k(p, q) = mrd_i(p, q) \leq mrd_i(p', q')$, mas como $i < k$ e $mrd_{m_{pts}}$ não pode diminuir quando m_{pontos} aumenta, segue-se que $mrd_k(p', q') = mrd_i(p, q) = mrd_i(p', q')$. Nesse caso,

A substituição de (p, q) no MST^* por (p', q') resultará em outro $MST_i' \in M_i$, porque o peso total de MST_i' e MST_i^* são iguais. A MST^*_i *inclui a borda* (p', q') *e, por s*uposição, a borda (p, q) é a mesma. tion $(p', q') \in MST_k$, e, portanto, $(p', q') \in MST_k \cup k\text{-}NN\ G$.

Segue-se que, dada uma *MST* de G_k , MST_k , e uma MST^* de G_i , $MST_i^* \in M_i$, com $k > i$, podemos construir um grafo ponderado MST' da seguinte forma: (1) incluir todas as arestas e pesos das bordas $(p, q) \in MST_i$ que pertencem a $MST_k \cup k\text{-}NN\ G$ (de acordo com os cenários (1) ou (2)-a); (2) substituir todas as bordas $(p, q) \in MST_i$ que não pertencem ao $MST_k \cup k\text{-}NN\ G$ com arestas (p', q') que devem existir no $MST_k \cup k\text{-}NN\ G$ (conforme cenário (2)-b), conectando os mesmos dois subconjuntos de pontos que (p, q) e tendo o mesmo peso de aresta que (p, q). Esse gráfico MST' é um *MST* de G_i *, ou seja,* $MST' \in M_i$ e $MST' \subseteq MST_k \cup k\text{-}NN\ G$. Portanto, existe um $MST_i \in M_i$ de modo que $MST_i \subseteq MST_k \cup k\text{-}NN\ G$. ■

O Teorema 1 sustenta que o gráfico completo $G_{m_{pts}}$ pode ser substituído por $MST_k \cup k\text{-}NNG$ ao computar

cenários:

(1) $(p, q) \in MST_k$

(2) $(p, q) \notin MST_k$

No cenário (1), uma vez que $(p, q) \in MST_k$, segue-se trivialmente que $(p, q) \in MST_k \cup k\text{-}NN\ G$. No cenário (2), MST_k deve ter uma aresta diferente (p', q') que conecte os conjuntos P e Q e tenha o menor peso entre as arestas que conectam esses conjuntos (observe que $MSTs$ são grafos conectados). Consequentemente, as duas afirmações a seguir são verdadeiras (i) $mrd_i (p, q) \leq mrd_i (p', q')$

$MST_{m_{pts}}$ resulta *em* $m_{pts} < k$, com potencial para grande economia de tempo de execução e memória. No Teorema 2, mostramos que *o* $MST_k \cup k\text{-}NNG$ também é um substituto válido para o Relative Neighborhood Graph (*RNG*), atualmente usado por um programa de última geração.
método artístico (consulte a Seção II).

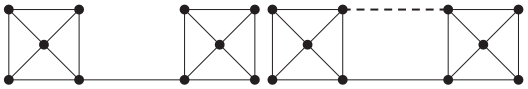
Teorema 2. $MST_k \cup k\text{-}NNG \subseteq RNG_k$.

Prova: Para qualquer aresta $(p, q) \in MST_k \cup k\text{-}NN\ G$, devemos considerar dois cenários (não mutuamente exclusivos):

(1) $(p, q) \in MST_k$

(2) $(p, q) \in k\text{-}NNG$

Para o cenário (1), Neto *et al* [19] comprovaram que o MST_k é um subgráfico do RNG_k e, consequentemente, cada



(a) $MST_k \cup k\text{-}NNG$, $k = 3$ (b) RNG_k, $k = 3$

Fig. 2: Exemplo do Teorema 2

A borda no MST_k também deve estar no RNG_k.

Para o cenário (2), precisamos provar que $k\text{-}NNG \subseteq RNG_k$.

De acordo com o Lema 1, se dois pontos p e q forem adjacentes no k-NN G, então a distância de acessibilidade mútua entre eles *com relação* a $m_{pts} = k$ é determinada pela maior de suas distâncias centrais. Vamos supor, sem perda de generalidade, que $c_k(p) \geq c_k(q)$, o que resulta em $mrd_k(p, q) = c_k(p)$. Devido às propriedades da função máxima, é trivial que $c_k(p) \leq max\{c_k(p), c_k(t), d(p, t)\}$ $\forall t$, que pode ser reescrito como:

$$c_k(p) \leq mrd_k(p, t) \quad \forall t \tag{2}$$

Novamente, devido às propriedades da função max, podemos substituir o lado direito da desigualdade 2 por um max

função que recebe como argumento o termo $mrd_k(p, t)$ e qualquer
outro termo, incluindo $mrd_k(q, t)$, *ou seja*:

$$c_k(p) \leq max\{mrd_k(p, t), mrd_k(q, t)\} \quad \forall t \tag{3}$$

De acordo com nossa suposição de que $mrd_k(p, q) = c_k(p)$, a desigualdade 3 pode ser reescrita somente em termos de *distâncias de alcance mútuo*:

$$mrd_k(p, q) \leq max\{mrd_k(p, t), mrd_k(q, t)\} \quad \forall t \tag{4}$$

Observe que a desigualdade 4 corresponde à formulação do RNG_k e determina que os pontos p e q são *vizinhos relativos* no espaço de distâncias de alcance mútuo [19], [20]. Isso também é verdadeiro assumindo que $mrd_k(p, q) = c_k(q)$ em vez disso (apenas trocando p e q). Esse resultado significa que a borda (p, q) considerada como estando no k-NNG deve necessariamente estar no RNG_k, *ou seja*, $k\text{-}NNG \subseteq RNG_k$.

Combinados, os resultados dos casos (1) e (2) implicam que a união de k-NNG e MST_k é um subgrafo de RNG_k. ■

Para mostrar que, em geral, o oposto não é verdadeiro, ou seja, $RNG_k \not\subset MST_k \cup k\text{-}NN\ G$, apresentamos um contraexemplo na Figura 2, em que o RNG_k (Figura 2b) tem mais uma aresta
do que o k-$NNG \cup MST_k$ (indicado por uma linha tracejada).

IV. Abordagem proposta

A. Algoritmo

Com base no Teorema 1, apresentamos no Algoritmo 1 uma abordagem para apoiar de forma eficiente a tarefa de calcular árvores de abrangência mínima baseadas em densidade múltipla $MST_{m_{pts}}$ para diferentes valores de m_{pts}, até um valor máximo k_{max}. Na parte (A), tanto *a* $MST_{k_{max}}$ quanto, sem muito esforço adicional, a $CORE\text{-}SG_{k_{max}} = MST_{k_{max}} \cup k_{max}\text{-}NNG$ são construídas. Qualquer algoritmo MST pode ser aplicado para construir um $MST_{k_{máximo}}$ sobre o gráfico completo e "virtual" [2] $G_{k_{máximo}}$, cujos pesos de borda são distâncias de acessibilidade mútua *em relação a* $m_{pts} = k_{max}$. Para calcular essas distâncias de acessibilidade mútua, é necessário determinar as distâncias centrais de cada ponto e, para determinar essas distâncias centrais, é necessário encontrar os k_{max} vizinhos mais próximos de cada ponto. $MST_{k_{max}}$ representa uma parte de
$CORE\text{-}SG_{k_{max}}$. O cálculo de $k_{max}\text{-}NNG$ pode ser feito "on-
durante o cálculo de acessibilidade mútua, começando com um gráfico vazio e, em seguida, adicionando a ele todas as bordas de
G_{kmax} cujos pesos são determinados pelas distâncias centrais *em relação a*
$m_{pts} = k_{max}$. Outra opção é executar primeiro k_{max}-nearest neighbor queries para encontrar, para cada ponto, o conjunto de pontos dentro de
sua distância do vizinho mais próximo k_{max} e armazenar os resultados de modo que
que podem ser usados tanto para o $MST_{k_{máximo}}$ computação (para

Vamos nos referir ao gráfico $MST_k \cup k\text{-}NNG$ como o *Gráfico de abrangência baseado na distância CORE*, $CORE\text{-}SG_k$. O

determinar as distâncias de acessibilidade mútua e de núcleo), bem como para a construção do k_{max} -*NN G*. Observe que só precisamos do *CORE-SG* como um gráfico não direcionado, pois o peso da borda de qualquer borda entre dois pontos é determinado pela distância de acessibilidade mútua, que não depende da "direção" de uma borda. Portanto, já construímos o *k-NNG* como um gráfico não direcionado, o que leva a uma pequena redução no número total de bordas em comparação com um gráfico de k-vizinhos *mais próximos* direcionado, já que há apenas uma borda (em vez de duas) para cada par de k-vizinhos *mais próximos*.

Algoritmo 1 Computação de *MSTs* baseadas em densidade múltipla

Exigir: X: conjunto de dados; < $k_1, \ldots, k_{max}$ >: sequência crescente de valores de m_{pts};

(A) Construir simultaneamente o $MST_{K_{max}}$ e o $CORE\text{-}SG_{K_{max}}$

1) Encontre e armazene os vizinhos mais próximos k_{max} para todos os objetos de dados em X e armazene as distâncias centrais de cada objeto para $m_{pts} \in [1, \ldots, k_{max}]$.
2) Calcule um $MST_{k_{max}}$ para o gráfico de acessibilidade mútua, $G_{k_{max}}$.
3) Calcule o gráfico k_{max}-*NNG* não direcionado, reutilizando as consultas de vizinhos mais próximos da Etapa 1.
4) Crie *o* $CORE\text{-}SG_{k_{max}}$ como k_{max} -$NNG \cup MST_{k_{max}}$.

(B) Obter as MSTs restantes

1) Para m_{pts} em $\{k_1, \ldots, k_{max}\}$:
   a) Obtenha *o* $CORE\text{-}SG_{m_{pts}}$ atribuindo distâncias de acessibilidade mútua *em relação a* m_{pts} como pesos de borda no $CORE\text{-}SG_{k_{max}}$.
   b) Encontre o $MST_{m_{pts}}$ da $CORE\text{-}SG_{m_{pts}}$.

$CORE\text{-}SG_{m_{pontos}}$ é um importante gráfico de substituição para o Gráfico completo $G_{m_{pts}}$ ao calcular *MSTs* baseadas em densidade e suas hierarquias de dados codificadas baseadas em densidade para vários valores de m_{pts}, porque seu número de arestas é $\Theta(nm_{pts})$, em em contraste com $\Theta(n^2)$ para o $G_{m_{pontos}}$ ou $O(n^2)$ para o *RNG*, e sua construção requer apenas informações prontamente disponíveis de uma pré-computação dinâmica única de um único *MST* baseado em densidade *e m relação* ao maior valor de m_{pts} em um intervalo desejado.

Para a parte (B) do Algoritmo 1, ou seja, o cálculo das *MSTs* adicionais para valores de m_{pts} menores que k_{max} todos os *MSTs* solicitados são computados independentemente e

[2]Cada par de pontos de dados (vértices) é adjacente nesse gráfico, que não precisa ser materializado, pois os pesos das bordas podem ser calculados dinamicamente.

diretamente de uma versão ponderada do $CORE\text{-}SG_{k_{max}}$. Para cada valor solicitado de $m_{pts} = i < k_{max}$, que pode ser solicitado em qualquer ordem, as bordas no $CORE\text{-}SG_{k_{max}}$ são atribuídas
pesos correspondentes à distância de acessibilidade mútua *com i*, e o MST_i é então extraído do gráfico ponderado resultante. Esse cálculo independente e não ordenado é fácil de implementar e pode ser vantajoso para a paralelização.

B. Otimização

Embora *o* $CORE\text{-}SG_{k_{max}}$ seja um substituto muito compacto para o $G_{(\cdot)}$, ele não é ideal, pois contém bordas que não fazem parte de nenhum dos *MSTs com* $m_{pts} < k_{max}$. Para encontrar o menor gráfico de substituição (ou seja, ideal), um procedimento opcional de otimização baseado no Teorema 1 pode ser usado para selecionar e remover essas bordas desnecessárias do $CORE\text{-}SG_{k_{max}}$. A ideia consiste em construir versões menores (em *relação ao* número de bordas) de um $CORE\text{-}SG_{m_{pts}}$ em cada iteração, para valores decrescentes de m_{pts} , usando *apenas* a quantidade necessária de vizinhos mais próximos para o valor atual de m_{pts} , descartando as bordas que não fazem parte do respectivo *MST* . O resultado, doravante denominado *CORE-SG** , é um gráfico que é ótimo (mínimo) como a união de todos os *MSTs* da sequência desejada de m_{pts} . Esse procedimento é apresentado no Algoritmo 2.

Algoritmo 2 Procedimento de otimização opcional

Exigir: $CORE\text{-}SG_{kmax}$ *;* MST_{kmax} *;* $< k_1, \ldots, k_{max} >$: sequência crescente de valores de $_{mpts}$;

1) *CORE-SG anterior* = $CORE\text{-}SG_{k_{max}}$.
2) $CORE\text{-}SG^* = {}_{MSTk}{}_{\substack{máx \\ imo}}$.
3) Para cada $_{mpts}$ em $< \mathbf{kmax\text{-}1}, \ldots, \mathbf{k1} >$:
   a) Atribuir distâncias de acessibilidade mútua *com relação aos* $_{mpts}$ como pesos de borda no *CORE-SG anterior*.
   b) Encontre *o* MST_{mpts} a partir do *CORE-SG anterior* ponderado, escolhendo as bordas que estão no $CORE\text{-}SG^*$ em caso de empate.
   c) Calcular o gráfico $mpts\text{-}NNG$ não direcionado, reutilizando parte das consultas de vizinhos mais próximos pré-computadas.
   d) $prev\_CORE\text{-}SG = m_{pts}\text{-}NNG \cup MST_{mpts}$.
   e) $CORE\text{-}SG^* = CORE\text{-}SG^* \cup {}_{MSTm}{}_{\substack{po \\ nto \\ s}}$.

A otimização é opcional e pode ser aplicada após a parte (A) do Algoritmo 1, antes ou como um substituto para a parte (B). Se todas as *MSTs* calculadas internamente no Algoritmo 2 forem armazenadas, elas poderão ser prontamente recuperadas para qualquer valor desejado de m_{pts} , de modo que a parte (B) do Algoritmo 1 se torna desnecessária. Como alternativa, podemos manter apenas o *CORE-SG* final* do Algoritmo 2, o que minimiza os requisitos de armazenamento e, em seguida, extrair os *MSTs* na parte (B) do Algoritmo 1 a partir desse gráfico. A extração das *MSTs* do *CORE-SG** é mais rápida do que a extração do $CORE\text{-}SG_{m_{pts}}$, especialmente se as bordas forem rotuladas com o(s) valor(es) m_{pts} para o(s) qual(is) elas fazem parte de uma $MST_{m_{pts}}$. Isso pode ser vantajoso para aplicativos que exigem recuperação rápida e sob demanda de um pequeno gráfico pré-processado, mas tem o preço da pré-computação adicional do *CORE-SG** no Algoritmo 2 e a desvantagem de que essa abordagem não é tão facilmente paralelizável.

C. Análise de complexidade

Nossométodo começapor computação $MST_{k_{\substack{máx \\ imo}}}$ e k_{max} *-NNG* para criar *o* $CORE\text{-}SG_{k_{\substack{máx \\ imo}}}$ com $\Theta(nk_{max})$ bordas. Esses cálculos são equivalentes a uma única execução do HDBSCAN* para $m_{pts} = k_{max}$, um algoritmo que é analisado em detalhes em [11]. Sua complexidade computacional no pior caso, supondo que $m_{pts} << n$ (como esperado na prática), é $O(n^2)$ quando as distâncias entre pares são fornecidas, ou $O(an^2)$ quando a função de distância subjacente $d(\text{-}, \text{-})$ pode ser calculada em tempo $O(a)$ para cada par de objetos. Observe que métodos como [23] e [25], que calculam um resultado aproximado do HDBSCAN*, possivelmente de uma forma computacionalmente mais eficiente, também podem ser usados para criar um *CORE-SG*. Entretanto, o uso de uma *MST* inicial aproximada levará a *MSTs* adicionais que também são apenas aproximadas. Após a obtenção do $CORE\text{-}SG_{k_{max}}$, as atualizações do peso da borda são necessário para cada valor de m_{pts} , com um custo computacional de $O(nm_{pts})$. Em seguida, o cálculo das *MSTs* adicionais (ou a verificação e recuperação de quais subconjuntos de arestas do *CORE-SG** fazem parte de cada uma delas) pode ser feito em $O(|E| \log |V|)$ usando qualquer algoritmo eficiente conhecido, por exemplo, o algoritmo de Kruskal ou Prim. Como $|E|$ é $O(nm_{pts})$ e $|V| = n$, a complexidade computacional para extrair um $MST_{m_{pts}}$ é $O(nm_{pts} \log n)$. Considerar um intervalo $[k_1, \ldots, k_{max}]$ de m valores de_{pts} com no máximo k valores de$_{max}$ no total resultará em um complexidade de $_{\substack{máx \\ imo}}$ log n) para calcular todas as *MSTs* $O(nk^2$ dentro de esse intervalo. Com $_{\substack{máx \\ imo}}$ $<< n$, como esperado em cenários reais, nossa abordagem tem como limite superior o cálculo inicial de $CORE\text{-}SG_{k_{\substack{máx \\ imo}}}$ que é $O(n^2)$ (veja acima).

D. Comparação com a abordagem RNG

Um gráfico usado anteriormente para substituir o $G_{m_{pts}}$ no cálculo de várias hierarquias do HDBSCAN* foi o *RNG* [19], [20], que mostrou uma economia computacional significativa em muitos cenários. No entanto, o *RNG* tem as seguintes deficiências em comparação com a abordagem proposta no presente documento:

- *Esforço computacional adicional*: o *RNG* não é uma parte intrínseca dos métodos baseados em densidade e deve ser com- posto de forma independente, o que adiciona um esforço computacional assintótico proibitivo de $O(n^3)$. Em uma tentativa de reduzir esse esforço, a heurística de pares bem separados proposta em
  [36] pode ser aplicado sob a suposição de que os pontos estão em posição geral, o que pode resultar em economia computacional, mas ainda é limitado por $O(n^3)$ no pior caso.
- *Restrições geométricas*: embora a adoção da heurística de pares bem separados muitas vezes possa resultar em economia computacional quando aplicável, sua suposição impõe restrições adicionais aos cenários de aplicação: a dissimilaridade/distância subjacente deve ser uma métrica e requer acesso direto às coordenadas espaciais de um objeto, ou seja, não pode ser aplicada a dados mais gerais.
- *Tamanho do gráfico*: o número de bordas no *RNG* geralmente é menor do que no gráfico completo $G_{m_{pts}}$, mas ainda é limitado por $O(n^2)$.
- *Ineficiência para dados de alta dimensão*: à medida que o número de dimensões aumenta, o número de vizinhos relativos aumenta.

Variáveis	Valores
kmax	20, 30, 50
#pontos1k	, 10k, 50k
#dimensões4	, 8, 16, 32, 64, 128
#clusters10	, 30, 50

TABELA I: Configuração experimental - Análise do gráfico base

(e bordas) aumenta, o que torna o *RNG* menos eficaz [19]. Além disso, o esforço para construir o *RNG* se aproxima do pior caso com o aumento da dimensionalidade.

Em contraste, o *CORE-SG* é um gráfico muito mais simples que pode ser obtido diretamente de uma única execução de um algoritmo baseado em densidade, como o HDBSCAN*. Esse gráfico não está restrito a métricas de distância adequadas e tem apenas $\Theta(kn)$ bordas, *ou seja*, $\Theta(n)$ se $k << n$, independentemente da dimensionalidade dos dados. Além disso, as propriedades do *CORE-SG* permitem sua otimização iterativa no gráfico de substituição mínima *CORE-SG** . As características acima tornam *o CORE-SG* uma alternativa muito melhor ao gráfico completo do que *o RNG*.

V. Avaliação Experimental

Implementamos nossa abordagem *CORE-SG* sem otimização (Algoritmo 1) e com otimização (*CORE-SG** , Algoritmo 2), e comparamos ambas com um *RNG* parcialmente filtrado que obteve o melhor desempenho geral entre as abordagens *RNG* [19], [20]. Para uma comparação justa de desempenho de tempo de execução, todas as implementações usadas neste trabalho computam o *MST* exato$_m$ de G_m . O *CORE-SG*,

Foram implementadas abordagens baseadas em *CORE-SG* e *RNG** em Python/Cython e os experimentos foram realizados em uma máquina virtual com 16 GB de RAM executando o Ubuntu 20.04.

Nossa avaliação experimental está organizada em três partes. N a primeira parte, discutimos o comportamento do *CORE-SG*, *CORE-SG** e *RNG** como gráficos de base e como seu tamanho muda *em relação ao* número de bordas de acordo com diferentes parâmetros. Na segunda parte, nossa análise se concentra no tempo de execução e na escalabilidade, variando cada parâmetro individualmente. Por fim, na terceira parte, são considerados conjuntos de dados reais para avaliar o comportamento de nossas abordagens em cenários reais.

A. Análise do gráfico base

A Tabela I exibe os parâmetros que variamos ao analisar o número de arestas nos gráficos de base considerados em nossa avaliação experimental. Avaliamos o comportamento de cada gráfico
em relação a k_{max} , bem como *em relação* ao número de pontos, dimensões e clusters nos dados. Para obter conjuntos de dados com propriedades variadas, conforme exibido na Tabela I, usamos o gerador de [37]. Calculamos *o* $MST_{m_{pts}}$ para todos os $m_{pts} = 1, 2, \text{---}, k_{max}$, de modo que k_{max} corresponde ao tamanho do intervalo m_{pts} .

A Figura 3 mostra o número de bordas para todas as k_{max} *-NN* graph, que depende apenas do valor de k_{max} e do tamanho do conjunto de dados. O *CORE-SG** se destaca por seu tamanho menor e estável em todos os cenários, pois o método de otimização reduz sistematicamente o número de bordas usando os resultados de iterações anteriores. Nosso concorrente de linha de base, o *RNG** , não é construído a partir do resultado de um cálculo anterior e, portanto, não pode ser otimizado de forma semelhante.

As bordas do *RNG** são parcialmente filtradas com base em uma heurística que se baseia exclusivamente em informações sobre os vizinhos k_{max} *-NN* de cada ponto para determinar se dois pontos são vizinhos relativos ou não. Valores menores de k_{max} fornecem menos informações sobre a vizinhança de cada ponto, e a heurística de filtragem perde eficácia, principalmente à medida que a dimensionalidade aumenta. Como resultado, o número de bordas em um *RNG** *com* valores menores de k_{max} torna-se significativamente maior do que o número de bordas de um *CORE-SG* correspondente.

O efeito que o aumento da dimensionalidade tem em um *RNG** tem a ver com a "esparsidade dos dados", que favorece a presença de vizinhos relativos no *RNG** . Ao adicionar um número maior de clusters (bem definidos e separados) nos dados, a quantidade de "espaço vazio" entre pares de pontos em diferentes clusters tende a aumentar, aumentando o número de vizinhos relativos. Como esperado, esse efeito se torna mais perceptível em espaços de alta dimensão, o que acentua a esparsidade dos dados em geral. Uma maior esparsidade reduz a eficácia da heurística de filtragem, resultando em um número maior de bordas no *RNG** para um número maior de clusters e maior dimensionalidade, principalmente, como mencionado, para valores menores de m_{pts} .

Ao aumentar o número de pontos, o número de gerados na construção do *RNG** aumenta,

combinações de dimensionalidade, número de pontos, número de clusters e k_{max} . De modo geral, observamos que o número de bordas do *CORE-SG* e do *CORE-SG** é muito estável, e o comportamento é semelhante nos diferentes cenários. Isso é explicado pela formulação do *CORE-SG*, amplamente baseada em um

especialmente em espaços de alta dimensão. Nesse caso, a ineficácia mencionada anteriormente da heurística de filtragem do *RNG** para valores mais baixos de k_{max} torna-se ainda mais perceptível. Esse comportamento é particularmente claro no subplot correspondente a 50k pontos de dados e 50 clusters na Figura 3, em que o número de bordas no *RNG** com k_{max} = 10 torna-se maior do que o número de bordas com k_{max} = 30, ou mesmo com k_{max} = 50 na extremidade alta do intervalo de dimensionalidade.

A Figura 4 informa o tempo de execução para construir o *RNG** *, o CORE-SG* e o *CORE-SG** , cujos tamanhos são informados na Figura 3. É possível observar que a construção do *RNG** é muito mais intensiva em termos de computação do que a construção do *CORE-SG*, exceto em conjuntos de dados de dimensões muito baixas. A diferença nos tempos de processamento aumenta drasticamente com o aumento da dimensionalidade, o aumento do tamanho dos dados e o aumento do valor de k_{max} . Esses resultados corroboram nossa alegação de que a construção de um *CORE-SG* exige um esforço computacional consideravelmente menor do que a construção de um *RNG* correspondente* e, ao mesmo tempo, é mais comportado em termos do número de bordas. Também está claro que o processo de otimização opcional para construir o *CORE-SG** acrescentou pouco esforço computacional adicional, apresentando um tempo de execução total próximo ao *CORE-SG* original.

B. Análise de tempo de execução e aumento de velocidade

Avaliamos o desempenho e a escalabilidade de nossa estratégia *com* os mesmos parâmetros da seção anterior, mas agora

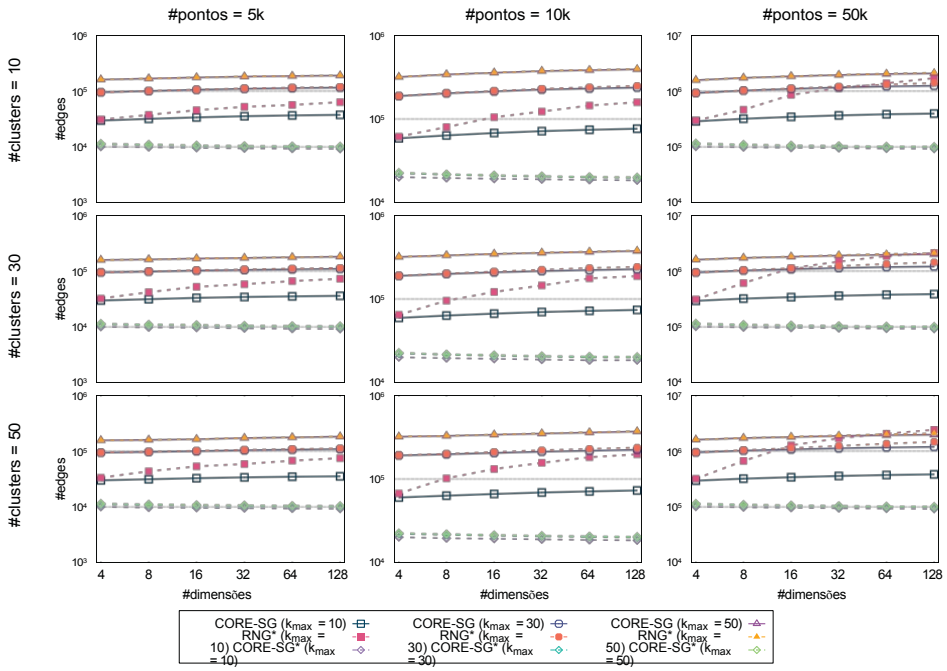


Fig. 3: Número de bordas no *RNG* , CORE-SG* e no *CORE-SG** .

Variáveis	Valores
kmax	20, 40, **60**, 80, 100
#pontos1k	, 5k, **10k**, 50k, 100k
#dimensões4	, 8, 16, **32**, 64, 128
#clusters10	, **30**, 50

TABELA II: Configuração experimental - tempo de execução e aumento de velocidade

dentro de faixas mais amplas dos valores dos parâmetros, conforme exibido na Tabela II. As entradas em negrito indicam valores padrão para cada parâmetro, que são mantidos fixos ao avaliar os efeitos individuais da variação dos valores de outros parâmetros.

1) *Análise do tempo de execução:* A Figura 5 mostra o tempo total (em minutos) para construir os gráficos de base e k_{max} *MSTs* para cada um dos métodos comparados, variando diferentes parâmetros. *Variação do número de clusters.* A Figura V-A mostra o tempo de execução.

tempo das diferentes abordagens ao aumentar o número de clusters bem separados em um conjunto de dados de tamanho fixo. Observe que *o CORE-SG* e *o CORE-SG** têm tempos de execução semelhantes, que são muito menores do que o tempo de execução da abordagem baseada no *RNG** . Por exemplo, para o conjunto de dados com 50 clusters, nossas abordagens são cerca de 20 vezes mais rápidas. Além disso, o tempo de execução da abordagem baseada no *RNG** aumenta com o número de clusters, o que provavelmente se deve a um número crescente de bordas entre clusters em um *RNG*. Nossas abordagens, por outro lado, não são realmente afetadas pelo número de clusters.

Variação de k_{max} . A Figura 5b indica que o tempo de execução, ao aumentar k_{max} , cresce lentamente para cada um dos métodos comparados. No entanto, o tempo de execução geral usando nosso *CORE-SG*

são muito mais baixos do que com o uso do *RNG** .

Variação do tamanho do conjunto de dados. A Figura 5c mostra os tempos de execução para conjuntos de dados de diferentes tamanhos. Como esperado, o tempo de execução de todos os três algoritmos aumenta *com* o número de pontos. Embora não haja diferença perceptível entre *o CORE-SG* e *o CORE-SG** , eles são substancialmente mais rápidos do que o *RNG** em todos os casos. Por exemplo, usar o *CORE-SG* para calcular $k_{max} = 60$ *MSTs* para o conjunto de dados com 100.000 pontos é mais de 20 vezes mais rápido do que usar a abordagem *RNG** .

Variação da dimensionalidade do conjunto de dados. A Figura 5d mostra os tempos de execução para diferentes dimensionalidades de dados. Semelhante ao que aconteceu ao aumentar o tamanho do conjunto de dados, não há diferença perceptível no tempo de execução entre *o CORE-SG* e *o CORE-SG** , e os tempos de execução de todos os três métodos aumentam com a dimensionalidade dos dados. Mais uma vez, enquanto o tempo de execução dos métodos propostos aumenta apenas ligeiramente, o do *RNG** aumenta muito mais substancialmente e em um ritmo mais rápido. Por exemplo, usar o *CORE-SG* para calcular $k_{max} = 60$ *MSTs* para o conjunto de dados de 128 dimensões (com 10.000 pontos) é 15 vezes mais rápido do que usar o *RNG** .

2) Construção do gráfico de base versus construção de MSTs adicionais: O comportamento do tempo total de execução dos diferentes métodos apresentados na Figura 5 é semelhante ao tempo de execução observado para a construção dos gráficos de base discutidos na Seção V-A. Isso sugere que o tempo total de execução para computar um grande número de *MSTs* usando essas abordagens é dominado pelo tempo de execução para construir os gráficos de base subjacentes. Aqui, mostramos que esse é de fato o caso, usando os conjuntos de dados *10k*, *50k* e *100k*. A Tabela III mostra a divisão do tempo de execução na construção do gráfico de base e na extração de *MSTs* adicionais.

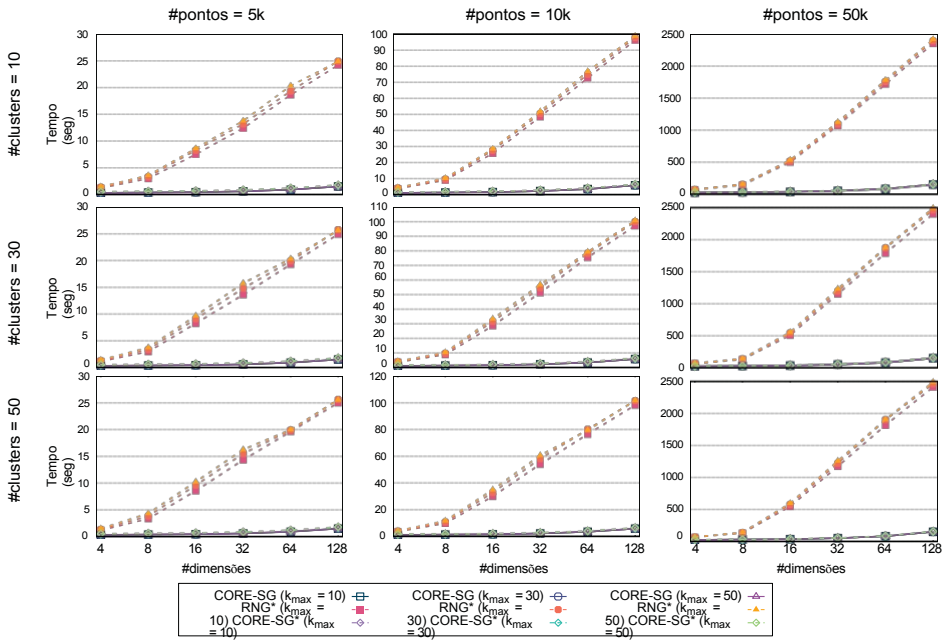
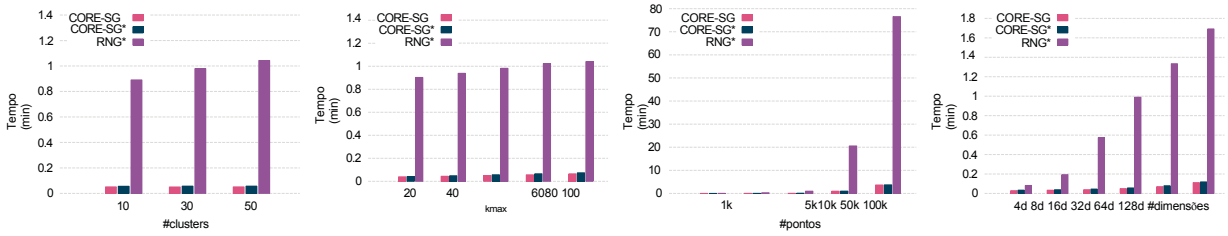


Fig. 4: Tempo de execução para criar o *RNG** , *CORE-SG* e no *CORE-SG** .



(a) #clusters vs. tempo (b) k_{max} vs. Tempo (c) #pontos vs. tempo (d) #dimensões vs. tempo

Fig. 5: Tempo de execução para criar o *CORE-SG*, *o CORE-SG** e o *RNG** e calcular k_{max} *MSTs*.

Embora os tempos de execução para extrair *MSTs* adicionais dos gráficos de base sejam semelhantes para o *CORE-SG* e o *RNG** , eles são reduzidos quase pela metade com o uso do *CORE-SG** . Observe que os tempos de execução para computar *MSTs* adicionais são ordens de grandeza menores do que os tempos de execução para construir os gráficos de base iniciais, tendo, portanto, menos impacto no tempo total.

3) Acelerando a construção de MSTs baseadas em densidade: Nesta seção, investigamos os aumentos de velocidade que podem ser obtidos ao calcular várias *MSTs* usando as abordagens *CORE-SG*, *CORE-SG** e *RNG** em comparação com a abordagem padrão que usa o gráfico completo de acessibilidade mútua. Medimos a relação entre o tempo total de execução para computar k_{max} *MSTs a* partir do gráfico completo e o tempo total de execução para computar os gráficos de substituição e extrair k_{max} *MSTs* desses gráficos. A Figura 6 mostra os aumentos de velocidade para conjuntos de dados de diferentes tamanhos ao usar diferentes valores de k_{max} , que também é o número de *MSTs* computados. Os números de dimensões e clusters foram definidos com seus valores padrão em negrito na Tabela II.

Para o *CORE-SG* e *o CORE-SG** , há uma interação perceptível entre o tamanho do conjunto de dados e o valor de k_{max} . Quando a proporção de k_{max} em relação ao número de pontos de dados é maior, o que significa que o tamanho máximo da vizinhança é relativamente grande em comparação com o tamanho do conjunto de dados, a redução alcançada no tamanho do gráfico (em relação ao gráfico completo) não é tão proeminente e, como resultado, os aumentos de velocidade são menores. Por outro lado, se k_{max} for pequeno em comparação com o tamanho do conjunto de dados (geralmente o caso em aplicativos práticos), os gráficos de base serão consideravelmente menores do que o gráfico completo, o que se traduz em acelerações maiores.

Observe que a otimização do *CORE-SG* para o *CORE-SG** exige um esforço extra e, embora a redução do número de bordas desempenhe um papel importante no cálculo de *MSTs* adicionais, a sobrecarga da etapa de otimização opcional ainda é um fator dominante no tempo total de execução, de modo que *o CORE-SG** produz taxas de aceleração menores do que o *CORE-SG*, mas ainda assim notavelmente superiores ao *RNG** . As taxas de aumento de velocidade observadas para o *RNG** quase não são afetadas à medida que k_{max} e o tamanho dos dados aumentam,

Conjunto de dados	Construção do gráfico de base			Construção de MSTs k_{max}			Tempo total		
	CORE-SG	CORE-SG*	RNG*	CORE-SG	CORE-SG*	RNG*	CORE-SG	CORE-SG*	RNG*
10k, 32d, 30c, kmax=60	0.0369	0.0506	0.9639	0.0130	0.0068	0.0134	0.0500	0.0574	0.9773
50k, 32d, 30c, kmax=60	0.8640	1.2215	20.4489	0.1037	0.0513	0.1053	0.9677	1.2728	20.5542
100k, 32d, 30c, kmax=60	3.4044	4.7882	76.3006	0.2406	0.1367	0.2509	3.6450	4.9249	76.5515

TABELA III: Tempo de execução (em minutos) para o gráfico de base e construção de *MSTs* adicionais para as abordagens proposta e de linha de base.

mas a sobrecarga de construção desse gráfico é muito maior, impedindo-o de atingir índices de aceleração semelhantes aos *do CORE-SG* e *do CORE-SG** .

C. Conjuntos de dados do mundo real

Aplicamos nossas técnicas a conjuntos de dados do mundo real em diferentes domínios de aplicativos, ou seja, áudio, imagem e texto. Todos os conjuntos de dados pré-processados são disponibilizados on-line e podem ser baixados para reprodução. O conjunto de dados FMA [38] foi criado para análise de música e consiste em recursos extraídos de uma coleção de músicas de vários gêneros e artistas. Usamos duas versões do conjunto de dados do FMA correspondentes a diferentes recursos de áudio: *croma* e *MFCC (Mel Frequency Cepstral Coefficient)*. O conjunto de dados ImageNet [39] tem sido amplamente usado pelas comunidades de aprendizado de máquina e visão computacional. Selecionamos aleatoriamente 80 clusters, correspondentes a aproximadamente 100 mil pontos desse conjunto de dados, em que cada ponto corresponde a uma imagem colorida de 16 × 16. Por fim, aplicamos nosso método ao conjunto de dados *20 Newsgroups* [40]. Esse conjunto de dados contém os valores TF-IDF dos termos extraídos dos artigos do Newsgroups. Para nossa finalidade, selecionamos os 1000 principais termos com os maiores valores de TF-IDF e aplicamos a PCA para reduzir o número de dimensões aos 500 componentes principais que explicam 75% da variação nos dados. O tamanho e a dimensionalidade de cada conjunto de dados são mostrados na Tabela IV.

Conjunto de dados	#pontos	# dimensões
FMA (Chroma Cens)	106,571	84
FMA (MFCC)	106,574	140
20 grupos de notícias	11,314	1000
20 Grupos de notícias (PCA)	11,314	500
ImageNet	100,610	768

TABELA IV: Conjuntos de dados reais.

A Figura 7 mostra o tempo total de execução (em minutos) necessário para construir os gráficos de base e $k_{max} = 60$ *MSTs*, para cada um dos conjuntos de dados considerados. Como esperado, não há diferença notável no desempenho entre as abordagens *CORE-SG* e *CORE-SG** , e ambas superam consideravelmente o *RNG** . Isso é consistente com os resultados de nossos experimentos anteriores envolvendo conjuntos de dados artificiais.

A Tabela V mostra a divisão detalhada do tempo de execução em tempo para a construção do gráfico de base e tempo para a construção de *MSTs* adicionais. Mais uma vez, é possível notar que a construção dos gráficos de base domina o tempo total de execução, que tem o *CORE-SG* como a abordagem mais rápida, seguido de perto pelo *CORE-SG** . Assim como nos experimentos com dados sintéticos, *o CORE-SG** apresenta os melhores tempos de execução para extrair *MSTs* adicionais dos gráficos de base, e esses tempos de execução correspondem a uma pequena fração do tempo para construir o gráfico de base inicial.

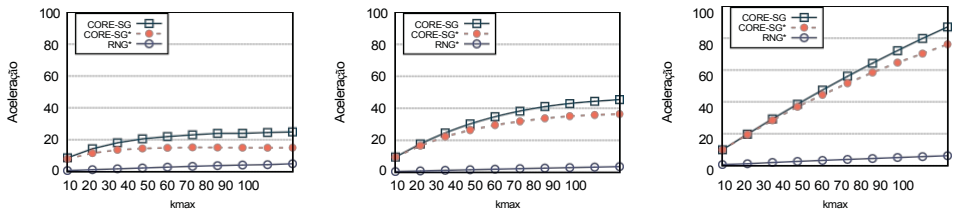
Até agora, demonstramos a eficiência da abordagem baseada no *CORE-SG* em comparação com o atual estado da arte, o *RNG** . Em seguida, mostraremos que (i) o esforço computacional necessário para criar um *CORE-SG* representa apenas um pequeno acréscimo em relação ao tempo de execução de um método típico baseado em densidade, como o HDBSCAN*, para um único valor de m_{pts} , e (ii) a produção do resultado do mesmo algoritmo baseado em densidade novamente para um valor diferente de m_{pts} pode ser obtida com uma pequena fração do custo ao usar uma abordagem baseada em *CORE-SG*.

A Tabela VI mostra os tempos de execução para calcular uma única hierarquia de agrupamento usando o HDBSCAN* "simples" com $m_{pts} = 60$, em comparação com a construção dos gráficos *CORE-SG* ou *CORE-SG** para $k_{max} = m_{pts} = 60$ e, em seguida, produzir a mesma hierarquia HDBSCAN* com base nos *MSTs* correspondentes. Observe que a adoção do *CORE-SG* acrescenta um mínimo de sobrecarga associada à sua construção, que varia de 0,05% a 0,66%. Conforme discutido anteriormente, a construção do *CORE-SG* não exige muito mais informações do que as já computadas pelos algoritmos baseados em densidade para aprendizado não supervisionado/semi- supervisionado. A construção *do CORE-SG** exige um esforço extra associado ao processo de otimização no Algoritmo 2, o que aumentou as taxas de sobrecarga na Tabela VI, mas sem afetar a ordem de grandeza do tempo total de execução.

A Tabela VII mostra os tempos de execução necessários para calcular uma segunda hierarquia de agrupamento, *com* um valor diferente do parâmetro m_{pts} ($m_{pts} = 30$), para cada um dos conjuntos de dados reais. Ao usar o HDBSCAN* "simples", o custo é essencialmente o mesmo que o custo inicial necessário para calcular a primeira hierarquia ($m_{pts} = 60$). Em contrapartida, com ambas as abordagens baseadas no *CORE-SG*, podemos calcular qualquer outra hierarquia em uma pequena fração do tempo necessário para calcular a primeira, levando, por exemplo, a um fator de aceleração de cerca de 7.700 vezes para os dados do ImageNet.

VI. Conclusão

Propusemos uma estratégia para substituir o gráfico completo que representa as estimativas de densidade entre pares de pontos em um conjunto de dados por um gráfico de abrangência simples e muito menor, o *CORE-SG*, com o objetivo de obter uma coleção de *MSTs* baseadas em densidade *em relação a* vários valores do parâmetro m_{pts} , que controla a suavidade das estimativas de densidade. Provamos que o *CORE-SG* correspondente a um determinado valor de m_{pts} contém todas as informações necessárias para encontrar *MSTs* para quaisquer valores menores de m_{pts} e que o *CORE-SG* nunca pode ser maior do que o gráfico de vizinhança relativa (RNG), a solução atual de última geração para esse problema. Também criamos um processo de otimização opcional para reduzir iterativamente o tamanho do *CORE-SG* para o *CORE-SG** mínimo, o que pode ser uma alternativa melhor para aplicativos com



(a) #pontos = 1k (b) #pontos = 5k (c) #pontos = 50k

Fig. 6: Aceleração em relação ao cálculo de *MSTs* adicionais para abordagens baseadas em *CORE-SG*, *CORE-SG** e *RNG** .

Conjunto de dados	Base Gı Construção dı ión aph			Construir ion de k_{max} MSTs			Tempo total		
	CORE-SG	CORE-SG*	RNG*	CORE-SG	CORE-SG*	RNG*	CORE-SG	CORE-SG*	RNG*
FMA(C)	7.51	7.96	78.00	0.27	0.15	0.29	7.77	8.11	78.29
FMA(M)	12.07	12.60	83.51	0.25	0.14	0.26	12.32	12.75	83.77
20NG(P)	0.43	0.63	4.08	0.09	0.08	0.09	0.52	0.72	4.17
20NG	0.86	0.96	7.17	0.07	0.11	0.07	0.92	1.07	7.24
IMGN	56.14	56.60	384.25	0.37	0.26	0.38	56.51	56.86	384.63

TABELA V: Detalhamento do tempo de execução para conjuntos de dados reais (em minutos).

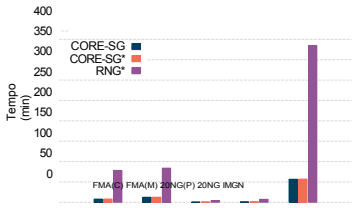


Fig. 7: Tempo de execução para criar *o CORE-SG*, *CORE-SG** e
*RNG** , e calcular $k_{max} = 60$ *MSTs* para conjuntos de dados reais.

Conjunto de dados	HDBSCAN*	CORE-SG HDBSCAN*	Sobrecarga	CORE-SG* HDBSCAN*	Sobrecarga
FMA(C)	7.36	7.39	0.33%	7.62	3.44%
FMA(M)	12.13	12.16	0.21%	12.39	2.13%
20NG(P)	0.43	0.44	0.66%	0.64	48.26%
20NG	0.85	0.86	0.41%	0.97	13.26%
IMGN	56.48	56.50	0.05%	56.79	0.56%

TABELA VI: Tempo de execução em minutos para calcular uma primeira hierarquia de agrupamento baseada em densidade com o HDBSCAN*, com e sem *o CORE-SG/CORE-SG** , e as despesas gerais correspondentes.

restrições de armazenamento e que exigem recuperação rápida e sob demanda em tempo real de vários MSTs baseados em densidade. Nossa proposta pode ser aplicada a uma coleção de algoritmos que processam uma *MST* dos dados no espaço de estimativas de densidade dadas por distâncias de alcançabilidade mútua. Demonstramos experimentalmente que tanto as versões otimizadas quanto as não otimizadas de nosso novo

Conjunto de dados	HDBSCAN*	CORE-SG HDBSCAN*	Aceleração	CORE-SG* HDBSCAN*	Aceleração
FMA(C)	7.37	0.008	874.57	0.006	1292.03
FMA(M)	12.27	0.009	1358.63	0.006	1972.46
20NG(P)	0.43	0.002	176.89	0.001	293.83
20NG	0.85	0.001	667.38	0.001	852.35
IMGN	56.62	0.011	5002.86	0.007	7786.86

TABELA VII: Tempo de execução em minutos para calcular uma segunda hierarquia de agrupamento baseada em densidade

superam significativamente a abordagem de última geração. De modo geral, o *CORE-SG* pode ser facilmente adotado por profissionais de ciência de dados, pois é mais eficiente, mais amplamente aplicável e muito mais fácil de implementar do que a abordagem anterior baseada em *RNG*. Em nossa avaliação experimental, mostramos que o uso do *CORE-SG* (i) é muito estável em termos do número de bordas em relação a diferentes fatores, (ii) é muito mais rápido para
com o HDBSCAN*, com e sem *o CORE-SG/CORE-SG** , e os aumentos de velocidade correspondentes.

(iii) ele pode acelerar o cálculo de, por exemplo, hierarquias de agrupamento adicionais baseadas em densidade por fatores de centenas a milhares.

No futuro, planejamos adaptar a abordagem *CORE-SG* a ambientes distribuídos e dimensionáveis, para lidar com dados distribuídos e dados que não podem ser processados em uma única máquina. Outra direção interessante para trabalhos futuros é investigar como um *CORE-SG* pode ser atualizado dinamicamente quando o conjunto de dados não é estático e pode mudar com o tempo. De forma ortogonal às linhas de investigação mencionadas anteriormente, conjecturamos que nosso método também poderia ser combinado com o agrupamento espectral para explorar potencialmente várias soluções para uma variedade de gráficos conectados, conforme discutido preliminarmente na Seção II. Também relacionado a essa ideia, *o CORE-SG* poderia ser fornecido como entrada para métodos de agrupamento espectral para buscar agrupamentos baseados em densidade incorporados no gráfico. Além disso, em aplicações de grande escala em que o usuário está disposto a trocar alguma precisão por ganhos computacionais adicionais, pode-se considerar o uso da busca por k-vizinhos *mais próximos* aproximada combinada com o *CORE-SG* proposto. Por fim, a capacidade de obter resultados para uma gama de valores de parâmetros de um método baseado em densidade em aproximadamente o mesmo tempo necessário para computar um único resultado com o método "simples" abre a porta para investigar novas estratégias de seleção de parâmetros baseadas em resultados.

AGRADECIMENTO

Este estudo foi parcialmente financiado pelo NSERC, FAPESP - Subvenção 2019/09817-6 e Serasa Experian.

Referências

[1] R. Sarkhel e A. Nandi, "Improving information extraction from visually rich documents using visual span representations", *Proc. VLDB Endow.*, vol. 14, no. 5, p. 822-834, Jan. 2021. [Online]. Available: https://doi.org/10.14778/3446095.3446104

[2] J. Ding, V. Nathan, M. Alizadeh e T. Kraska, "Tsunami: A learned multi-dimensional index for correlated data and skewed workloads", *Proc. VLDB Endow.*, vol. 14, no. 2, p. 74-86, outubro de 2020. [Online].
Available: https://doi.org/10.14778/3425879.3425880

[3] G. Li, X. Zhou, S. Li e B. Gao, "Qtune: A query-aware database tuning system with deep reinforcement learning", *Proc. VLDB Endow.*, vol. 12, no. 12, p. 2118-2130, ago. 2019. [Online]. Available: https://doi.org/10.14778/3352063.3352129

[4] Y. Zhang e A. Kumar, "Panorama: A data system for unbounded vocabulary querying over video", *Proc. VLDB Endow.*, vol. 13, no. 4, p. 477-491, Dec. 2019. [Online]. Available: https://doi.org/10.14778/3372716.3372721

[5] D. Wen, L. Qin, Y. Zhang, L. Chang e X. Lin, "Efficient structural graph clustering: an index-based approach", *The VLDB Journal*, vol. 28, no. 3, pp. 377-399, Jun 2019. [Online]. Available: https://doi.org/10.1007/s00778-019-00541-4

[6] A. C. A. Neto, M. A. Nascimento, J. Sander e R. J. G. B. Campello, "Mustache: Um explorador de hierarquias de agrupamento múltiplo", *Proc. VLDB Endow.*, vol. 11, no. 12, p. 2058-2061, ago. 2018. [Online]. Available: https://doi.org/10.14778/3229863.3236259

[7] R. J. G. B. Campello, P. Kro¨ger, J. Sander e A. Zimek, "Density-based clustering", *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 2, p. e1343, 2020. [Online]. Disponível: https://doi.org/10.1002/widm.1343

[8] M. Ester, H.-P. Kriegel, J. Sander e X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", em *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, p. 226-231.

[9] M. Ankerst, M. M. Breunig, H.-P. Kriegel e J. Sander, "Optics: Ordering points to identify the clustering structure", em *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '99. Nova York, NY, EUA: ACM, 1999, pp. 49-60. [Online]. Available: http://doi.acm.org/10.1145/304182.304187

[10] R. J. G. B. Campello, D. Moulavi e J. Sander, "Density-based clustering based on hierarchical density estimates", em *Advances in Knowledge Discovery and Data Mining*, ser., Lecture Notes in Computer Science, J. Pei, V. S. Tseng, L. Cao, H. Motoda e G. Xu, Eds. Lecture Notes in Computer Science, J. Pei, V. S. Tseng, L. Cao, H. Motoda e G. Xu, Eds., vol. 7819. Berlim, Heidelberg: Springer, 2013, pp. 160-172. [Online].
Disponível: https://doi.org/10.1007/978-3-642-37456-2_14

[11] R. J. G. B. Campello, D. Moulavi, A. Zimek e J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection", *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 1, pp. 5:1-5:51, jul. 2015. [Online]. Available: http://doi.acm.org/10.1145/2733381

[12] W. Savoie, T. A. Berrueta, Z. Jackson, A. Pervan, R. Warkentin,
S. Li, T. D. Murphey, K. Wiesenfeld e D. I. Goldman, "A robot made of robots: Emergent transport and control of a smarticle ensemble", *Science Robotics*, vol. 4, no. 34, 2019. [Online]. Disponível: https://robotics.sciencemag.org/content/4/34/eaax4316

[13] T. M. Norman, M. A. Horlbeck, J. M. Replogle, A. Y. Ge, A. Xu,
M. Jost, L. A. Gilbert e J. S. Weissman, "Exploring genetic interaction manifolds constructed from rich single-cell phenotypes", *Science*, vol. 365, no. 6455, pp. 786-793, 2019. [Online]. Disponível: https://science.sciencemag.org/content/365/6455/786

[14] I. Djonlagic, S. Mariani, A. L. Fitzpatrick, V. M. G. T. H. Van Der Klei, D. A. Johnson, A. C. Wood, T. Seeman, H. T. Nguyen, M. J. Prerau, J. A. Luchsinger, J. M. Dzierzewski, S. R. Rapp, G. J. Tranah,
K. Yaffe, K. E. Burdick, K. L. Stone, S. Redline e S. M. Purcell, "Macro and micro sleep architecture and cognitive performance in older adults", *Nature Human Behaviour*, vol. 5, no. 1, pp. 123-145, jan. 2021. [Online]. Available: https://doi.org/10.1038/s41562-020-00964-y

[15] L. A. Miccio e G. A. Schwartz, "Mapping chemical structure- glass transition temperature relationship through artificial intelligence", *Macromolecules*, vol. 54, no. 4, pp. 1811-1817, fev. 2021. [Online]. Disponível: https://doi.org/10.1021/acs.macromol.0c02594

[16] Logan, C. H. A. e Fotopoulou, S., "Unsupervised star, galaxy, qso classification - application of hdbscan", *A&A*, vol. 633, p. A154, 2020. [Online]. Available: https://doi.org/10.1051/0004-6361/201936648

[17] J. C. Gertrudes, A. Zimek, J. Sander e R. J. G. B. Campello, "A unified framework of density-based clustering for semi-supervised classification" (Uma estrutura unificada de agrupamento baseado em densidade para classificação semissupervisionada), em *Anais da 30ª Conferência Internacional sobre Gerenciamento de Banco de Dados Científicos e Estatísticos, SSDBM 2018, Bozen-Bolzano, Itália, 09 a 11 de julho de 2018*, D. Sacharidis, J. Gamper e
M. H. Bo¨hlen, Eds. ACM, 2018, pp. 11:1-11:12. [Online]. Available: https://doi.org/10.1145/3221269.3223037

[18] --, "Uma visão unificada dos métodos baseados em densidade para agrupamento e classificação semissupervisionados", *Data Min. Knowl. Discov.*, vol. 33, no. 6, pp. 1894-1952, 2019. [Online]. Available: https://link.springer.com/article/10.1007/s10618-019-00651-1

[19] A. C. A. Neto, J. Sander, R. J. G. B. Campello e M. A. Nascimento, "Efficient computation and visualization of multiple density-based clustering hierarchies", *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 8, pp. 3075-3089, 2021. [Online]. Available: https://doi.org/10.1109/TKDE.2019.2962412

[20] --, "Efficient computation of multiple density-based clustering hierarchies", em *2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, EUA, 18-21 de novembro de 2017*,
V. Raghavan, S. Aluru, G. Karypis, L. Miele e X. Wu, Eds. IEEE Computer Society, 2017, pp. 991-996. [Online]. Disponível: https://doi.org/10.1109/ICDM.2017.127

[21] H. Shiokawa, Y. Fujiwara e M. Onizuka, "Scan++: Efficient algorithm for finding clusters, hubs and outliers on large-scale graphs", *Proc. VLDB Endow.*, vol. 8, no. 11, p. 1178-1189, Jul. 2015. [Online]. Available: https://doi-org.ez31.periodicos.capes.gov.br/10.14778/2809974.2809980

[22] A. Lulli, M. Dell'Amico, P. Michiardi e L. Ricci, "Ng-dbscan: Scalable density-based clustering for arbitrary data", *Proc. VLDB Endow.*, vol. 10, no. 3, p. 157-168, Nov. 2016. [Online]. Disponível: http://www.vldb.org/pvldb/vol10/p157-lulli.pdf

[23] L. McInnes e J. Healy, "Accelerated hierarchical density based clustering", em *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 33-42.

[24] M. de Berg, A. Gunawan e M. Roeloffzen, "Faster dbscan and hdbscan in low-dimensional euclidean spaces", *International Journal of Computational Geometry & Applications*, vol. 29, no. 01, pp. 21-47, 2019. [Online]. Available: https://doi.org/10.1142/S0218195919400028

[25] Y. Wang, S. Yu, Y. Gu e J. Shun, "Fast parallel algorithms for euclidean minimum spanning tree and hierarchical spatial clustering", em *SIGMOD '21: Conferência Internacional sobre Gerenciamento de Dados, Evento Virtual, China, 20 a 25 de junho de 2021*, G. Li, Z. Li, S. Idreos e
D. Srivastava, Eds. ACM, 2021, pp. 1982-1995. [Online]. Available: https://doi.org/10.1145/3448016.3457296

[26] J. A. d. Santos, T. I. Syed, M. C. Naldi, R. J. G. B. Campello e
J. Sander, "Hierarchical density-based clustering using mapreduce", *IEEE Transactions on Big Data*, vol. 7, no. 1, pp. 102-114, 2021. [Online]. Available: https://doi.org/10.1109/TBDATA.2019.2907624

[27] X. Xu, N. Yuruk, Z. Feng e T. A. J. Schweiger, "Scan: A structural clustering algorithm for networks", em *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '07. Nova York, NY, EUA: ACM, 2007, p. 824-833. [Online]. Available: https://doi.org/10.1145/1281192.1281280

[28] J. M. Gonza'lez-Barrios e A. J. Quiroz, "A clustering procedure based on the comparison between the k nearest neighbors graph and the minimal spanning tree", *Statistics and Probability Letters*, vol. 62, no. 1,
pp. 23-34, 2003. [Online]. Disponível: https://doi.org/10.1016/S0167-7152(02)00421-2

[29] A. S. Arefin, C. Riveros, R. Berretta e P. Moscato, "knn-mst-agglomerative: A fast and scalable graph-based data clustering approach on gpu", em *2012 7th International Conference on Computer Science Education (ICCSE)*, 2012, pp. 585-590.

[30] A. S. Arefin, C. Riveros, R. Berretta e P. Moscato, "The mst-knn with paracliques", em *Artificial Life and Computational Intelligence*, S. K. Chalup, A. D. Blair e M. Randall, Eds. Cham: Springer International Publishing, 2015, pp. 373-386.

[31] A. K. Jain e R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, EUA: Prentice-Hall, Inc., 1988.

[32] H.-P. Kriegel, P. Kro¨ger, J. Sander e A. Zimek, "Density-based clustering", *WIREs Data Mining and Knowledge Discovery*, vol. 1, no. 3,
pp. 231-240, 2011. [Online]. Disponível: https://doi.org/10.1002/widm.30

[33] S. Hess, W. Duivesteijn, P. Honysz e K. Morik, "The spectacl of nonconvex clustering: A spectral approach to density- based clustering", em *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 3788-3795. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33013788
[34] U. von Luxburg, "A tutorial on spectral clustering", *Stat. Comput.*, vol. 17, no. 4, pp. 395-416, 2007. [Online]. Available: https://doi.org/10.1007/s11222-007-9033-z
[35] P. Veenstra, C. Cooper e S. Phelps, "Spectral clustering using the knn-mst similarity graph", em *2016 8th Computer Science and Electronic Engineering Conference, CEEC 2016, Colchester, Reino Unido, 28 a 30 de setembro de 2016*. IEEE, 2016, pp. 222-227. [Online]. Available: https://doi.org/10.1109/CEEC.2016.7835917
[36] P. B. Callahan e S. R. Kosaraju, "A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields", *Journal of the ACM*, vol. 42, no. 1, pp. 67-90, 1995.
[37] J. Handl e J. Knowles, "Cluster generators for large high-dimensional data sets with large numbers of clusters", 2005.
[38] M. Defferrard, K. Benzi, P. Vandergheynst e X. Bresson, "FMA: um conjunto de dados para análise musical", na *18ª Conferência da Sociedade Internacional para Recuperação de Informações Musicais (ISMIR)*, 2017. [Online]. Disponível: https://arxiv.org/abs/1612.01840
[39] J. Deng, W. Dong, R. Socher, L. Li, K. Li e L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", em *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 de junho de 2009, Miami, Flórida, EUA*. IEEE Computer Society, 2009, pp. 248-
255. [Online]. Available: https://doi.org/10.1109/CVPR.2009.5206848
[40] K. Lang, "Newsweeder: Learning to filter netnews", em *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, A. Prieditis and S. J. Russell, Eds. Morgan Kaufmann, 1995, pp. 331-339. [Online]. Disponível: http://people.csail.mit.edu/jrennie/20Newsgroups/