UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

ROBSON NOVATO LOBÃO
Orientador: Jadson Castro Gertrudes

UM COMPARATIVO ENTRE MÉTODOS DE CONSTRUÇÃO DE GRAFOS PARA CLASSIFICAÇÃO SEMISSUPERVISIONADA

Ouro Preto, MG
2024

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

ROBSON NOVATO LOBÃO

UM COMPARATIVO ENTRE MÉTODOS DE CONSTRUÇÃO DE GRAFOS PARA CLASSIFICAÇÃO SEMISSUPERVISIONADA

Monografia II apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Jadson Castro Gertrudes

Ouro Preto, MG
2024

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO

FOLHA DE APROVAÇÃO

Robson Novato Lobão

Um comparativo entre métodos de construção de grafos para classificação semissupervisionada

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 9 de Fevereiro de 2024.

Membros da banca:

Jadson Castro Gertrudes (Orientador) - Doutor - Universidade Federal de Ouro Preto
Lais Soares Caldeira (Examinadora) - Mestre - PPGCC UFOP
Hugo Eduardo Ziviani (Examinador) - Mestre - PPGCC UFOP

Jadson Castro Gertrudes, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 9/02/2024.

Documento assinado eletronicamente por **Jadson Castro Gertrudes**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 13/02/2024, às 17:24, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0665736** e o código CRC **6979D5D7**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591692 - www.ufop.br

Dedico este trabalho a todos meu familiares, amigos e professores que sempre me ajudaram e incentivaram nesse processo de formação como bacharel em Ciência Da Computação pela UFOP.

Agradecimentos

Agradeço aos meus familiares, em especial meus pais, Ana Bela e Edevaldo e à minha irmã, Iasmin, pelo apoio, conselhos e paciência no meu processo de graduação. Aos meus professores que me auxiliaram na formação como pessoa e profissional, em especial ao Jadson, meu orientador de monografia. Aos meus amigos e minha namorada, Marina, pelo suporte e incentivo.

Resumo

O aprendizado semissupervisionado se posiciona no ponto intermediário entre o aprendizado supervisionado e não supervisionado. Ele busca encontrar informações de pontos em um conjunto de dados com base em outras instâncias que já foram previamente classificadas. Ele é relevante em situações em que podemos facilmente coletar informações que geram um determinado conjunto de dados, entretanto, classificá-las manualmente é desvantajoso. Há 4 formas descritas na literatura para aprendizado semissupervisionado, são elas a de Modelo Generativo (*Generative Models*), Separação De Baixa Densidade (*Low-Density Separation*), Mudança de Representação (*Change of Representation*) e Métodos Baseados em Grafos (*Graph-Based Methods*). O presente trabalho apresenta uma comparação entre métodos de construção de grafos para a propagação de rótulos no processo de aprendizado semissupervisionado. Especificamente, foi avaliado o desempenho do algoritmo semissupervisionado de campo aleatório Gaussiano (*Gaussian Random Field*) ao ser apresentado com um grafo que representa uma árvore geradora mínima, produzida pelo *framework* HDBSCAN*, em comparação com métodos tradicionais de construção de grafos. Resultados superiores foram observados em dois conjuntos de dados, quando comparados a experimentos realizados anteriormente.

Palavras-chave: Aprendizado Semissupervisionado. Aprendizado não supervisionado. Classificação semissupervisionada. Grafos. Propagação de Rótulos.

Abstract

Semi-supervised learning sits at the midpoint between supervised and unsupervised learning. It seeks to find information about points in a dataset based on other instances that have already been previously classified. It is relevant in situations where we can easily collect information that generates a certain set of data, however, classifying it manually is disadvantageous. There are 4 forms described in the literature for semi-supervised learning, they are Generative Models, Low-Density Separation, Change of Representation and Graph-Based Methods. The present work presents a comparison between graph construction methods for label propagation in the semi-supervised learning process. Specifically, the performance of the semi-supervised Gaussian Random Field algorithm was evaluated when presented with a graph that represents a minimum spanning tree, produced by *framework* HDBSCAN*, in comparison with traditional graph construction methods . Superior results were observed in two data sets when compared to previously performed experiments.

Keywords: Semi-supervised Learning. Unsupervised learning. Semi-supervised classification. Graphs. Label Propagation.

Lista de Ilustrações

Lista de Tabelas

Lista de Algoritmos

Lista de Abreviaturas e Siglas

ϵN	ϵ-neighborhood
KNN	κ-nearest neighbors
HM	Similarity function of Hein & Maier
LLE	Local Linear Embedding
mutKNN	mutual KNN
symKNN	symmetric KNN
symFKNN	symmetry-favored KNN
GRF	Gaussian Random Fields
LGC	Local and Global Consistency
LapRLS	Laplacian Regularized Least Squares
LapSVM	Laplacian Support Vector Machines
RMGT	Robust Multi-class Graph Transduction
AGM	Árvore Geradora Mínima
HDBSCAN*	Hierarchical Density-Based Spatial Clustering of Applications with Noise *

Sumário

1 Introdução

O uso generalizado do aprendizado de máquina teve início por volta de 1970, quando começou a ser aplicado extensivamente para resolver problemas práticos. Esse conceito envolve a programação de dispositivos computacionais para aprender com experiências passadas, utilizando princípios de inferência para extrair conclusões genéricas a partir de um conjunto específico de exemplos (FACELI et al., 2021).

Dentro do aprendizado de máquina podemos ter o aprendizado supervisionado, que é um paradigma de treinamento de modelos no qual o algoritmo é alimentado com um conjunto de dados que inclui exemplos rotulados (KOTSIANTIS; ZAHARAKIS; PINTELAS, 2006), enquanto o aprendizado não supervisionado o algoritmo é alimentado com um conjunto de dados que não possui rótulos ou categorias previamente definidas. Ao contrário do aprendizado supervisionado, onde o modelo recebe exemplos rotulados para aprender uma relação entre entradas e saídas, no aprendizado não supervisionado, o objetivo é descobrir padrões, estruturas ou relações intrínsecas nos dados sem a orientação de rótulos preexistentes (HINTON; SEJNOWSKI, 1999).

O aprendizado semissupervisionado (SSL) é uma abordagem no campo de aprendizado de máquina em que o conjunto de dados utilizado para o treinamento contém tanto exemplos rotulados (com classes conhecidas) quanto exemplos não rotulados (sem classes atribuídas). Este método tem ganhado crescente atenção e aplicação, principalmente devido à realidade de que nem todas as bases de dados disponíveis para treinamento apresentam todas as suas instâncias devidamente etiquetadas (CHAPELLE; SCHOLKOPF; ZIEN, 2006). Ao treinar um modelo, geralmente é necessário utilizar conjuntos de dados extensos para orientar as decisões do modelo treinado.

Em muitos casos, é comum que bancos de dados volumosos possuam a maioria de suas instâncias sem rótulos, pois a aquisição de dados é mais fácil do que a sua categorização. Por exemplo, no reconhecimento de fala, gravar grandes quantidades de áudio tem baixo custo computacional, mas rotular esses dados requer esforço humano para transcrição. Outro exemplo é no sequenciamento de proteínas, onde os dados são obtidos em larga escala industrial, mas a determinação de suas funções demanda um extenso trabalho científico (CHAPELLE; SCHOLKOPF; ZIEN, 2006).

Dado que os dados não rotulados fornecem menos informações do que os rotulados, é essencial tê-los em grande quantidade para melhorar significativamente a precisão das previsões. Isso destaca a importância de algoritmos semissupervisionados rápidos e eficientes (CHAPELLE; SCHOLKOPF; ZIEN, 2006).

Neste estudo, investigamos abordagens semissupervisionadas para a propagação de rótulos, que é o processo de atribuir ou espalhar rótulos conhecidos para instâncias não rotuladas,

com base no trabalho de Sousa, Rezende e Batista (2013), que realiza uma comparação entre o desempenho de métodos de construção de grafos e propagação de rótulos. Tipicamente, esse processo é composto por três fases: a primeira etapa abrange a criação do grafo de adjacências, seguida pela segunda fase que inclui a geração de matrizes ponderadas com base nesse grafo. Por fim, a terceira fase engloba a propagação de rótulos para os dados não rotulados, utilizando algoritmos de propagação/difusão de rótulos. Geralmente, a eficácia desses métodos pode ser impactada pela forma como o grafo é construído.

Em geral, o algoritmo *k-nearest neighbor* (KNN) é o método mais utilizado para construção de grafos. Neste caso, uma aresta entre dois objetos $\mathbf{x}_i$ e $\mathbf{x}_j$ existe desde que $\mathbf{x}_j$ seja um dos k objetos próximos a $\mathbf{x}_i$. Algumas estratégias de simetria utilizando o KNN são aplicadas, tais como o KNN mútuo, que constrói uma aresta desde que $\mathbf{x}_j$ seja um dos k vizinhos de $\mathbf{x}_i$ e $\mathbf{x}_i$ seja um dos k vizinhos de $\mathbf{x}_j$ e o KNN simétrico, que determina o peso da aresta em ambas as direções desde que um objeto esteja dentro da vizinhança do outro ($\mathbf{x}_i \in N(\mathbf{x}_j)$ or $\mathbf{x}_j \in N(\mathbf{x}_i)$).

Estudos têm sido conduzidos em relação aos métodos de construção de grafos, com foco na determinação dos parâmetros mais apropriados para sua execução. Por exemplo, valores elevados para o parâmetro k podem resultar em grafos densos, inviabilizando sua aplicação em problemas que envolvem grandes volumes de dados. Por outro lado, valores reduzidos para esses parâmetros podem resultar em grafos esparsos e fragmentados, o que, por sua vez, pode exigir uma quantidade substancial de dados rotulados para otimizar o desempenho de algoritmos semissupervisionados (ARAÚJO; ZHAO, 2016).

No escopo deste trabalho, propomos a abordagem na construção de grafos, que consiste na criação de uma árvore geradora mínima por meio do grafo de alcançabilidade mútua. Essa abordagem é obtida por intermédio do algoritmo de agrupamento de dados baseado em densidade HDBSCAN* (*Hierarchical Density-Based Spatial Clustering of Applications with Noise*), conforme proposto por Campello et al. (2015).

O agrupamento baseado em densidade consiste na identificação de agrupamentos de pontos de dados em um espaço, levando em consideração a densidade local. Ao contrário dos métodos tradicionais de agrupamento, que presumem que os *clusters* possuem formas e tamanhos predefinidos, o agrupamento baseado em densidade possibilita a detecção de grupos com diversas formas e tamanhos, fundamentando-se na densidade de pontos. O HDBSCAN* é um algoritmo de agrupamento baseado em densidade que incorpora os conceitos de densidade local e uma abordagem hierárquica para identificar *clusters* de maneira robusta, automaticamente determinando o número de *clusters* (CAMPELLO et al., 2015).

A geração de um grafo que evidencie uma estrutura potencial de grupos pode fornecer suporte ao algoritmo de propagação de rótulos durante a sua implementação. No contexto dos experimentos realizados nesta pesquisa, empregamos o algoritmo de *Gaussian Random Fields – GRF* (ZHU; GHAHRAMANI; LAFFERTY, 2003), reconhecido como um dos métodos mais emblemáticos para a propagação de rótulos. Cabe ressaltar que esse algoritmo também foi

empregado em estudos anteriores, como o estudo realizado por Sousa, Rezende e Batista (2013), permitindo, assim, uma comparação direta com os resultados do artigo original.

1.1 Justificativa

Raramente nos deparamos com conjuntos de dados integralmente rotulados em cenários práticos, uma vez que a obtenção de dados não rotulados de maneira manual é facilmente acessível e prática, enquanto a tarefa de rotulagem precisa e adequada é dispendiosa e complexa (LIU; CHANG, 2009). Em tais contextos, os algoritmos de aprendizado semissupervisionado são frequentemente empregados, uma vez que podem aprimorar a eficácia do processo de aprendizado ao considerar informações implícitas presentes nos dados não rotulados, evitando assim a necessidade de rotulagem manual extensiva.

Além disso, a incorporação da construção de grafos como uma fase preliminar é respaldada pela assertiva de que os grafos constituem uma ferramenta substancial para a visualização de informações, permitindo a detecção de padrões e a disseminação de rótulos com base na estrutura, conforme discutido por Zhang et al. (2019). Existe a possibilidade que a abordagem de construção de grafos, particularmente aquela fundamentada no *framework* HDBSCAN*, tenha um impacto positivo no desempenho de classificação do algoritmo de propagação de rótulos GRF. Isso ocorre, pois tal método efetua a construção de um grafo capaz de revelar potenciais regiões densas dentro de um conjunto de dados.

1.2 Objetivos

O propósito deste estudo é empregar um método alternativo de construção de grafos para avaliar empiricamente o impacto no desempenho do algoritmo semissupervisionado de propagação de rótulos GRF. O grafo em questão será o grafo de alcançabilidade mútua, gerado por meio do algoritmo HDBSCAN*. Com o intuito de atingir esse objetivo principal, delineamos os seguintes objetivos específicos:

- Realizar uma revisão do estado da arte em relação aos métodos variantes de KNN para construção de grafos;
- Implementar e disponibilizar o método construtor de grafos baseado na alcançabilidade mútua;
- Avaliar o desempenho do método por meio de métricas como acurácia, taxas de erro médio e desvio padrão, comparando os resultados obtidos com os de Sousa, Rezende e Batista (2013).

1.3 Organização do Trabalho

O restante do documento está organizado na seguinte forma: O Capítulo 2 apresenta a Revisão Bibliográfica/ Embasamento Teórico (com o referencial teórico e trabalhos relacionados). O Capítulo 3 apresenta a Metodologia ou Desenvolvimento (material e métodos). O Capítulo 4 apresenta os resultados obtidos, além das discussões dos resultados. Por fim, o Capítulo 5 apresenta as conclusões do trabalho e perspectivas de trabalhos futuros.

2 Fundamentação teórica

Neste capítulo, são introduzidos os conceitos fundamentais que constituem a base para a compreensão deste estudo.

2.1 Aprendizado semissupervisionado

Inicialmente, tratando-se do pilar fundamental de pesquisa desse trabalho, o aprendizado de máquina semissupervisionado realiza o treinamento de um modelo utilizando um conjunto de dados que hora são rotulados, hora não (CHAPELLE; SCHOLKOPF; ZIEN, 2006). Normalmente, ao lidar com bancos de dados, há alguma classificação que buscamos para cada instância do banco, por exemplo, ao tratar de casos de sucesso no tratamento de uma doença, a classificação buscada é se o paciente está curado ou não. Dessa forma, podemos classificar cada instância de um banco de dados de pacientes como curado ou não curado. No modelo de aprendizado semissupervisionado, uma parte dos dados teriam essa classificação.

No aprendizado semissupervisionado, o conjunto de dados $\mathbf{X} = (\mathbf{x}_i)_{i\in[n]}$ pode ser dividido em duas partes: os pontos $\mathbf{X}_l = (\mathbf{x}_1, \ldots, \mathbf{x}_l)$, para os quais os rótulos $\mathbf{Y}_l = (y_1, \ldots, y_l)$ são fornecidos, e os pontos $\mathbf{X}_u = (\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u})$, cujos rótulos não são conhecidos. Isso é conhecido como aprendizado semissupervisionado "padrão" (CHAPELLE; SCHOLKOPF; ZIEN, 2006).

2.1.1 Gaussian Random Fields

Para este estudo foi utilizado o *Gaussian Random Fields* (GRF), que é uma abordagem que pode ser vista semelhante a dos vizinhos mais próximos, a qual dados rotulados tenham dados adjuntos com mesma classificação (ZHU; GHAHRAMANI; LAFFERTY, 2003).

O funcionamento básico parte do princípio de que se tem l pontos rotulados e u pontos não rotulados, onde normalmente $l < u$ e $n = l + u$ sendo o total de pontos. Assume-se inicialmente que os rótulos são binários: $y \in 0, 1$. Considere um grafo $G = (V, E)$ com vértices V correspondentes aos n pontos do conjunto de dados $L = 1, ..., l$ correspondendo aos pontos rotulados com rótulos $y_1, ..., y_l$ e vértices $U = l + 1, ..., l + u$ correspondendo aos pontos não rotulados. A função do algoritmo é definir rótulos aos vértices U . Assume-se uma matriz $n \times n$ ponderada chamada W (ZHU; GOLDBERG, 2022).

A estratégia é computar uma função *real-valued* $f : V \to \mathbb{R}$ que defina os rótulos baseados em f. Intuitivamente, queremos que pontos não rotulados tenham rótulos semelhantes a seus vizinhos, o que leva ao uso da função de energia quadrática dada na equação 2.1

$$E(x) = \frac{1}{2} \sum_{i,j} w_{ij} \cdot (f(i) - f(j))^2 \tag{2.1}$$

Onde:

w_{ij} é ponderação associada à interação entre os pontos i e j

$f(i)$ função ou um conjunto de valores associados a cada ponto i do sistema

$f(j)$ função ou um conjunto de valores associados a cada ponto j do sistema

E para dar uma distribuição de probabilidade nas funções f nós formamos os *Gaussian Fields* usando a equação 2.2

$$p_\beta(f) = \frac{e^{-\beta E(f)}}{Z_\beta}, \tag{2.2}$$

Onde:

$E(f)$ é a energia da configuração f

β é o parâmetro de inversão da temperatura

Z_β é a função de partição.

Essa função harmônica satisfaz a condição de $\Delta f = 0$ em pontos não rotulados, e U é igual a f_1 em pontos rotulados. Δ é uma laplaciana (conceito matemático que descreve como uma quantidade varia em diferentes partes de um espaço) combinatória dada em forma de matriz $\Delta = D - W$ onde $D = diag(d_1)$ é a diagonal de $\sum_j w_{ij}$ e $W = [w_{ij}]$.

Além disso, uma função é dita harmônica quando o valor de f em cada instância não rotulada é a média dos f dos seus vizinhos, como dito na equação 2.3.

$$f(j) = \frac{1}{d_j} \sum_{i \sim j} w_{ij} f(i), \quad \text{para } j = l+1, \ldots, l+u \tag{2.3}$$

Para computar a solução em termos de operações de matriz nós dividimos a matriz ponderada W em 4 blocos depois da l-ésima coluna e linha como na matriz 2.4:

$$W = \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix} \tag{2.4}$$

Sobrando $f = \begin{bmatrix} f_l \\ f_u \end{bmatrix}$ e sendo $y_l = (y_1, ..., y_l)^T$ temos que a solução harmônica é dada pela equação 2.5 (ZHU; GHAHRAMANI; LAFFERTY, 2003):

$$\begin{aligned} f_l &= y_l \\ f_u &= -L_{uu}^{-1} L_{ul} Y_l \end{aligned} \tag{2.5}$$

2.2 Grafos

Para o fluxo deste trabalho será necessária a breve compreensão do conceito de grafos, extensivamente aplicado no campo da computação, tem como propósito representar informações por meio de vértices e arestas. Além disso, possibilita a observação de nuances que anteriormente não estavam explicitamente delineadas (ANSCOMBE, 1973). Na Figura 2.1, é apresentado um exemplo simples ilustrativo desse conceito.

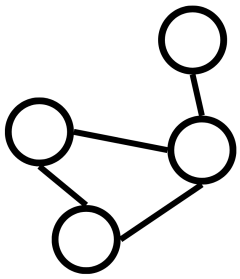


Figura 2.1 – Exemplo de grafo.

Fonte: Elaborado pelo autor.

Os grafos, segundo Goldbarg e Goldbarg (2012) funcionam como uma abstração, formalizando relações de interdependência existente entre elementos de um conjunto. Um conceito importante é a direcionalidade do grafo, um grafo é dito direcionado quando as relações entre os elementos de um conjunto N dependem da sua origem. Um exemplo em que grafos direcionados são úteis são em modelagens de circuitos elétricos em que o vértice de onde origina e de onde finaliza são pertinentes, como pode ser visto na figura 2.2.

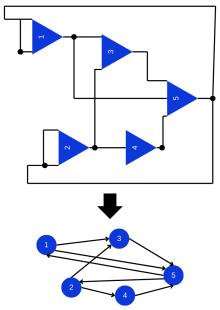
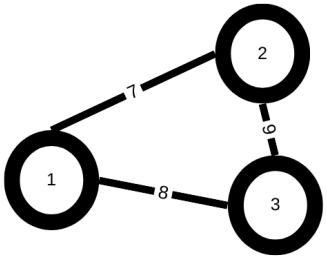


Figura 2.2 – Representação em grafo de um circuito elétrico.

Fonte: Elaborado pelo autor.

Outro conceito importante é o de grafos ponderados, note que no grafo da figura 2.1 não há valores associados às arestas, isso significa que sair de um ponto e chegar em outro não tem um custo ou ganho embutido. Várias vezes isso não é verdade, se fôssemos representar uma linha de ônibus intermunicipais cada caminho receberia um valor, ou seja, cada aresta (linha de ônibus) possui um valor. Um exemplo está na figura 2.3. E por fim, o conceito de grafos completos, segundo Erdös (1959), são aqueles todos os conjuntos de dois vértices estão conectados por uma aresta.

Ademais, os grafos, embora sejam convenientes aos humanos, não possuem uma estrutura computacional que se assemelhe a expressão geométrica de um grafo, necessitando a utilização de algumas possíveis estruturas alternativas para sua representação, como a matriz de adjacência (GOLDBARG; GOLDBARG, 2012). Representada por uma matriz quadrada com dimensão igual ao número de vértices em que cada valor da matriz corresponde ao peso daquela ligação, ou 0 e 1 em casos de matrizes não ponderadas, 0 caso não haja aresta entre os pontos, 1 caso haja. Um exemplo de uma matriz ponderada resultante de um grafo está na figura Figura 2.3.



	1	2	3
1	0	7	8
2	7	0	9
3	8	9	0

Figura 2.3 – Exemplo de uma matriz ponderada.

Fonte: Elaborado pelo autor.

2.2.1 Árvore geradora mínima

Outro conceito importante para a compreensão deste trabalho é o de Árvore Geradora Mínima (AGM). Uma AGM de um grafo G é um subgrafo gerador que é conexo e acíclico. Devido às propriedades de uma árvore, todo grafo conexo possui pelo menos uma árvore geradora. A AGM

é aquela que possui o menor custo dentre todas as possíveis em G (GOLDBARG; GOLDBARG, 2012). Em outras palavras, as AGMs são subgrafos de grafos conectados que vinculam todos os vértices minimizando o peso total das arestas incluídas. Características-chave como conectividade, ausência de ciclos e peso mínimo distinguem as Árvores Geradoras Mínimas.

Existem dois algoritmos amplamente conhecidos para a construção de AGMs: o algoritmo de Kruskal e o algoritmo de Prim. O algoritmo de Kruskal começa com vértices individuais considerados como componentes conexos, incorporando gradualmente arestas com o menor peso que evitam a formação de ciclos. Por outro lado, o algoritmo de Prim inicia com um único vértice e expande sistematicamente a AGM, selecionando a aresta de menor peso que estende a árvore (GOLDBARG; GOLDBARG, 2012). Na Figura 2.4, é possível visualizar um exemplo de Árvore Geradora Mínima a partir de um grafo.

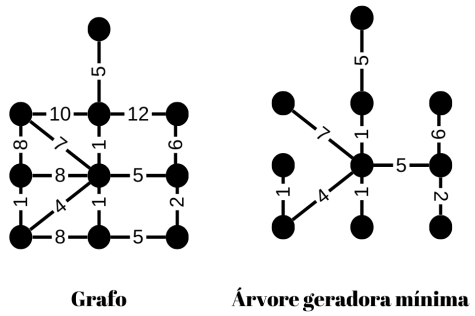


Figura 2.4 – Árvore geradora mínima.

Fonte: Elaborado pelo autor.

Este conceito vai ser importante pois, posteriormente, no uso do HDBSCAN* uma das etapas envolve a extração de uma AGM de um grafo completo gerado por uma das etapas do *framework*.

2.2.2 Métodos para construção de grafos

O algoritmo k-nearest neighbors (KNN) é reconhecido como um dos métodos clássicos e simples para a classificação de padrões. Sua aplicação pode resultar em resultados competitivos e, em determinados domínios, quando combinado com conhecimento contextual específico, pode alcançar avanços significativos e contribuir para o estado da arte. A regra KNN atribui a cada exemplo não rotulado a etiqueta mais frequente entre seus k vizinhos mais próximos no conjunto de treinamento. Assim, o desempenho desse algoritmo está diretamente relacionado à

escolha adequada da métrica de distância utilizada para identificar os vizinhos mais próximos. Na ausência de conhecimento contextual dos dados, a maioria dos classificadores KNN recorre à métrica euclidiana simples, conforme descrita na Equação 2.6:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_d - y_d)^2} \tag{2.6}$$

Onde:

$d(\mathbf{x}, \mathbf{y})$ é a distância euclidiana entre os pontos $\mathbf{x}$ e $\mathbf{y}$,

x_i são as coordenadas do ponto $\mathbf{x}$ ao longo da dimensão i, para $i = 1, 2, \ldots, d$,

y_i são as coordenadas do ponto $\mathbf{y}$ ao longo da dimensão i, para $i = 1, 2, \ldots, d$,

d é a dimensão dos pontos.

Em geral, a distância euclidiana é utilizada para medir as dissimilaridades entre exemplos representados como entradas de vetores (SUN; HUANG, 2010). Entretanto, o algoritmo principal proposto pode gerar grafos não simétricos (SOUSA; REZENDE; BATISTA, 2013), por isso são oferecidas 3 outras abordagens que serão tratadas no próximo tópico para resolução desse problema.

Sobre o *mutual* KNN, ele é definido como um grafo no qual existe uma aresta entre os vértices v_i e v_j se cada um deles pertencer aos k vizinhos mais próximos (em termos da métrica de similaridade original) do outro vértice. Por outro lado, um grafo KNN tem uma aresta entre os vértices v_i e v_j se um deles pertencer aos k vizinhos mais próximos do outro. Portanto, o grafo *mutual* KNN é um subgrafo do grafo KNN calculado a partir dos mesmos dados com o mesmo valor de k (OZAKI et al., 2011). De forma similar temos o *symmetric* KNN, a principal diferença entre o KNN tradicional e o KNN simétrico é que, no KNN simétrico, a atenção é voltada para identificar pares de pontos que são mutuamente vizinhos mais próximos. Isso significa que, se um ponto A é um dos k vizinhos mais próximos de um ponto qualquer B, e ao mesmo tempo, o ponto B é um dos k vizinhos mais próximos de A, então esses dois pontos formam um par simétrico. Por fim, o *symmetry-favored* KNN que é uma representação estrutural para representar as conexões entre vértices em um grafo direcionado ou não direcionado. Enquanto a matriz de adjacência binária é composta apenas por valores 0 e 1 para indicar a presença ou ausência de arestas, a matriz de adjacência não binária pode conter outros valores (SOUSA; REZENDE; BATISTA, 2013),

2.3 Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN*)

Para compreender o algoritmo HDBSCAN*, é essencial contextualizá-lo dentro do domínio do agrupamento de dados. Este último diz respeito à organização de um conjunto de

objetos, pontos ou elementos em grupos, de modo que os elementos dentro de cada grupo sejam similares entre si em algum aspecto específico (FACELI et al., 2021). Em geral, os algoritmos de agrupamento são categorizados como particionais ou hierárquicos, sendo que dentro de ambas abordagens encontramos aqueles baseados em densidade.

Um agrupamento baseado em densidade é um conceito na análise e mineração de dados, onde os objetos são agrupados com base na densidade de pontos próximos em um espaço de características. Nesse contexto, foram desenvolvidos diversos algoritmos para gerar *clusters*, nome dado a aglutinação de dados, baseados em densidade, sendo um deles de particular interesse para a construção de grafos: o HDBSCAN*. Este algoritmo, um método de agrupamento hierárquico, produz uma hierarquia completa de agrupamentos baseados em densidade. O HDBSCAN* facilita a extração de uma árvore de agrupamentos simplificada, composta pelos agrupamentos mais significantes. Sua operação inicia com o cálculo da distância "core" ($CoreDistance(\mathbf{x})$) para todos os objetos de dados, seguido pela construção de um grafo de alcançabilidade mútua.

O cálculo da *Core Distance* é dado pelo raio mínimo necessário para que uma instância qualquer X seja considerada um objeto *core* tendo em vista um número mínimo de pontos (*minPts*) próximos, definidos pelo usuário, incluindo o próprio ponto. O grafo de alcançabilidade mútua ou *Mutual Reachability Graph* (MRG) é um grafo completo, explicado na seção 2.2, em que os pontos são os objetos do *dataset* e os pesos das arestas é a distância de alcançabilidade entre os pontos, dado os pontos x_1 e x_2 essa distância é calculada pelo maior valor entre: *CoreDistance(*x_1*)*, *CoreDistance(*x_2*)* e a distância, calculada por alguma métrica normalmente a euclidiana, explicada na subseção 2.2.2, entre os dois pontos x_1 e x_2.

Esse grafo é então expandido para obter uma Árvore Geradora Mínima (AGM), adicionando auto-arestas com a distância do núcleo como pesos. A hierarquia HDBSCAN* é extraída como um dendrograma da AGM, em que cada nível representa uma solução de agrupamento distinta (CAMPELLO et al., 2015).

2.4 Detecção de *outliers* em um conjunto de dados

Tendo em vista que este trabalho necessita lidar com diversos conjuntos de dados, algumas informações são relevantes e necessárias para compreender o contexto do *dataset*. Sendo assim, urge a necessidade de estudar sobre os *outliers* e como detectá-los. *Outliers* são instâncias que divergem substancialmente do comportamento esperado ou padrão de um conjunto de dados, frequentemente caracterizadas por valores que se encontram além de limites estatisticamente significativos (DAVIES; GATHER, 1993). Segundo Ding, Zhang e He (2017) uma das melhores formas de encontrar *outliers* é utilizando *Local Outlier Factor* (LOF).

O LOF funciona da seguinte forma; Primeiramente, o LOF calcula a densidade local para cada instância, avaliando a proximidade em relação aos seus vizinhos. Essa densidade local representa a quantidade de pontos em uma vizinhança específica ao redor de cada ponto. Em

seguida, o LOF atribui uma pontuação LOF a cada ponto com base na densidade local. Essa pontuação quantifica a anormalidade de um ponto em relação aos seus vizinhos. Pode-se dizer que pontos com escores LOF mais altos são considerados mais propensos a serem *outliers*, ou seja, são menos consistentes com o comportamento geral do conjunto de dados (DING; ZHANG; HE, 2017).

2.5 Trabalhos Relacionados

Este trabalho se inicia utilizando metodologias disponibilizadas por Campello et al. (2015) para criação de grafos, com explicações detalhadas na seção 2.3. Complementando com a segunda etapa deste trabalho, que é a propagação dos rótulos nesses grafos, o estudo de Zhu, Ghahramani e Lafferty (2003) propõe uma abordagem para aprendizado semissupervisionado baseado em GRF. Dados rotulados e não rotulados são representados como vértices em um gráfico ponderado, com pesos de aresta codificando a similaridade entre instâncias. O problema de aprendizagem é então formulado em termos de um campo aleatório gaussiano nesse grafo, onde a média do campo é caracterizada em termos de funções harmônicas e é obtida eficientemente usando métodos matriciais ou propagação de crenças, seu trabalho gerou resultados promissores em *datasets* de classificação de digitais e de texto. Ambos os trabalhos antecessores motivaram a criação deste trabalho.

Em estudos que comparam métodos de construção de grafos para aprendizado semissupervisionado, o trabalho de Ozaki et al. (2011) iniciou um estudo que posteriormente veio a ser um dos basilares no ramo, em seu estudo ele expandiu o estado da arte utilizando o *mutual* KNN como forma de construção de grafos gerando resultados melhores que os que existiam e com menor custo computacional, que inspirou o trabalho de Sousa, Rezende e Batista (2013).

O trabalho de Sousa, Rezende e Batista (2013) fornece uma comparação empírica detalhada de cinco algoritmos de SSL baseados em grafos, combinados com três métodos de construção de grafos de adjacência e três métodos de geração de matrizes ponderadas. Sua avaliação experimental indicou que os algoritmos de SSL são fortemente afetados pelo valor do parâmetro de esparcificação do grafo e pela escolha dos métodos de construção do grafo de adjacência e geração de matrizes ponderadas. Mostraram também que os algoritmos que possuem parâmetros de regularização também dependem da configuração adequada desses parâmetros. Como resultado demonstraram a superioridade do mutKNN em relação aos grafos symKNN e symFKNN. Além disso, mostraram que o mutKNN tende a gerar erros mais suaves do que os grafos symKNN e symFKNN. De forma similar, optaram pelo RBF Kernel como método que melhor performou. Em consonância, em um trabalho similar de Zhu, Ghahramani e Lafferty (2003), foi introduzido uma abordagem para aprendizado semissupervisionado baseada em um modelo de campo aleatório gaussiano definido em relação a um grafo ponderado que representa dados rotulados e não rotulados, os métodos já avaliados na literatura podem ser encontrados

concatenados na tabela 2.1.

Tabela 2.1 – Esquema de métodos validados por Sousa, Rezende e Batista (2013)

Processo completo		
Construção dos grafos		Propagação
Construção do Grafo de Adjacência	**Geração de Matriz Ponderada**	**Propagação de rótulos**
MutKNN	RBF	GRF
symKNN	HM	LGC
symFKNN	LLE	LapRLS
		LapSVM
		RMGT

Fonte: Elaborado pelo autor.

Entretanto, embora fossem explorados muitos métodos e combinações de táticas, tanto de propagação como de construção de grafos, careceu na etapa da construção de um grafo de adjacências de algum método que leve em consideração uma base de dados densos e com pouca variância. Uma forma de lidar com isso é utilizando métodos baseados em densidade que definem *clusters* com base na densidade local dos pontos de dados, procurando por regiões densas no espaço de características e agrupando pontos que estão próximos uns dos outros e têm densidades similares. Ademais, como a árvore geradora mínima resultante do grafo de alcançabilidade mútua gera um grafo esparso, grafos esparsos tendem a ter menos arestas, o que pode levar a redução na complexidade computacional dos algoritmos baseados em grafos. Isso pode ser benéfico para algoritmos semissupervisionados, especialmente em grandes conjuntos de dados, sendo assim, permitindo utilizar esse método como etapa do método de construção de grafos, como mostrado na tabela 2.2.

Tabela 2.2 – Esquema de métodos abordados

Processo completo	
Construção dos grafos	Propagação
Construção do Grafo de Adjacência	**Propagação de rótulos**
AGM extraída do HDBSCAN*	GRF

Fonte: Elaborado pelo autor.

Considerando a lacuna identificada no cenário atual, que se refere à falta de classificações precisas em conjuntos de dados com baixa variância, conforme abordado pelos algoritmos mencionados na literatura por Sousa, Rezende e Batista (2013), este estudo tem como objetivo oferecer uma contribuição para superar tal desafio.

3 Desenvolvimento

Este trabalho combina a construção de grafos com um método de propagação de rótulos, que são técnicas utilizadas em aprendizado de máquina para propagar informações de rótulos conhecidos para pontos não rotulados ou parcialmente rotulados em um conjunto de dados (ISCEN et al., 2019). Uma pesquisa empírica foi implementada com as configurações descritas na Seção 2.5. O passo a passo do processo implementado nesse trabalho pode ser encontrado no Algoritmo 1 e a descrições dos passos nas seções que seguem.

Algorithm 1 Algoritmo de propagação de rótulos

procedure GRAPH-CONSTRUCTION-SSL
    Passo 1: Computar *core distance*.
        Raio mínimo necessário para que uma instância qualquer X seja considerado um objeto core, tendo em vista um número mínimo de pontos próximos.
    Passo 2: Criar grafo de alcançabilidade mútua.
        É grafo completo em que os pontos são os objetos do *dataset* e os pesos das arestas é a distância de alcançabilidade entre os pontos, dado os pontos x_1 e x_2 essa distância é calculada pelo maior valor entre: $CoreDistance(x_1)$, $CoreDistance(X_2)$ e $Distncia(X_1, X_2)$.
    Passo 3: Árvore geradora mínima.
        Criar uma AGM usando Kruskal a partir do grafo do Passo 2.
    Passo 4: Difundir rótulos.
        Difundir rótulos baseados em GRF.
    Passo 5: Avaliação
        Com base no menor erro médio encontrado de acordo com os parâmetros variados são encontrados os resultados.
end procedure

3.1 Base de dados utilizada

No artigo de Sousa, Rezende e Batista (2013) foram utilizados diferentes bases de dados: USPS, COIL2, DIGIT-1, G-241C, G-241N e TEXT. COIL2 é um conjunto de dados para classificação de imagens, USPS e DIGIT-1 são conjuntos de dados para reconhecimento de dígitos, G-241N e G-241C são conjuntos de dados para classificação de distribuições gaussianas e TEXT é um conjunto de dados para classificação de texto. A utilização de diversas bases de dados pode ser justificada ao alegar que o trabalho visa ambientar resultados em qualquer tipo de caso de entrada, e a integração dos resultados desses dados para gerar uma conclusão consistente, já que, em aplicações de mundo real, os dados podem estar distribuídos em departamentos, lojas, arquivos e muitas outras estruturas distintas (CASTRO; FERRARI, 2016). Nesses casos, um dos passos essenciais antes da escolha de uma técnica de aprendizado usando a base é a concatenação de todos os resultados obtidos anteriormente e a escolha de um deles.

O USPS, foi um conjunto de dados de referência criado a partir do conjunto de dígitos manuscritos USPS. Foram selecionado aleatoriamente 150 imagens de cada um dos dez dígitos. Os dígitos "2"e "5"foram atribuídos à classe +1, enquanto todos os outros formaram a classe -1. As classes, portanto, estão desequilibradas, com tamanhos relativos de 1:4 (CHAPELLE; SCHOLKOPF; ZIEN, 2006).

O COIL2 e a COIL-100 são um conjunto de imagens coloridas de 100 objetos capturados em diferentes ângulos. Foi reduzido a resolução das imagens para 16 × 16 pixels e selecionados aleatoriamente 24 objetos dos 100 disponíveis. Esses objetos foram divididos em seis classes, com quatro objetos em cada uma. Foram descartadas algumas imagens de cada classe, deixando 250 imagens em cada uma. Por fim, foi ocultado a estrutura das imagens usando um algoritmo específico (CHAPELLE; SCHOLKOPF; ZIEN, 2006).

Sobre o TEXT, sob o dataset Newsgroup foi aplicada a codificação tf-idf (frequência do termo - frequência inversa do documento) resultando em uma representação esparsa com 11.960 dimensões. Para o benchmark, 750 pontos de cada classe foram selecionados aleatoriamente e as características foram permutadas aleatoriamente (TONG; KOLLER, 2000).

Sobre o DIGIT1, esse conjunto de dados foi projetado para consistir em pontos próximos a uma variedade de baixa dimensionalidade inserida em um espaço de alta dimensionalidade, mas sem mostrar uma estrutura de cluster pronunciada. Portanto, partindo de um sistema que gera escritas artificiais (imagens) do dígito "1"(CHAPELLE; SCHOLKOPF; ZIEN, 2006).

Sobre o G-241C e G-241N, são datasets de geração de pontos clusterizados que são desenhados em um plano cartesiano, eles são a extração de pontos de gaussianas isotrópicas de variância unitária, isso é, refere-se ao processo de gerar pontos (dados) a partir de distribuições gaussianas isotrópicas específicas, onde isotrópico significa que a variância é a mesma em todas as direções (CHAPELLE; SCHOLKOPF; ZIEN, 2006), consequentemente acaba gerando *datasets*, que segundo Somanath, Rohith e Kambhamettu (2011) são densos, por ter pontos que fazem uma cobertura abrangente do espaço de características, não diferindo muito da média.

Abaixo pode ser encontrado uma tabela 3.1 com as propriedades dos conjuntos de dados organizados:

Tabela 3.1 – Propriedades básicas dos datasets.

Dataset	**Classes**	**Dimensão**	**Pontos**	**Comentário**
G-241C	2	241	1500	Artificial
G-241N	2	241	1500	Artificial
DIGIT	2	241	1500	Artificial
USPS	2	241	1500	Desbalanceado
COIL	6	241	1500	
TEXT	2	11960	1500	Esparço

Fonte: Elaborado pelo autor.

3.2 Construção de grafos

Inicialmente, foram injetados o conjunto de *datasets* disponibilizados na biblioteca *sslbookdata* em que compila todos *datasets* descritos por Chapelle, Scholkopf e Zien (2006) em sua obra de aprendizado semissupervisionado, um dos parâmetros que foram utilizados no algoritmo é a diminuição da dimensionalidade dos conjuntos de dados feito através da análise de componente principal em que normalizamos os dados do dataset e aplicamos a Análise de Componentes Principais (PCA) reduzindo de 241 atributos para 50.

O método HDBSCAN* realiza a construção de um grafo por meio de um conjunto de etapas cuidadosamente definidas. Na primeira etapa é calculada a distância central para cada objeto de dados no conjunto da *sslbookdata*, esse valor é chamado de *Core Distance*, e é dado pelo raio mínimo necessário para que uma instância qualquer seja considerada um objeto core, tendo em vista um número mínimo pontos próximos incluindo o próprio ponto. Para calcular esse raio mínimo testes com 2, 3, 4 e 5 pontos foram utilizados para experimentação e a métrica de distância dos vizinhos mais próximos foram: euclidiana, manhattan, minkowski e do cosseno.

Com base nessa primeira etapa, o algoritmo computa um Grafo de Alcançabilidade Mútua que é um grafo completo em que os pontos são, novamente, os objetos dos *datasets* e os pesos das arestas é a distância de alcançabilidade entre os pontos ligados pela aresta (CAMPELLO et al., 2015). A distância de alcançabilidade é o maior valor entre a *Core Distance* do objeto 1, *Core Distance* do objeto 2 e a distância euclidiana entre os dois pontos.

Criado esse grafo de alcançabilidade mútua, é computada então uma árvore geradora mínima utilizando o algoritmo de Kruskal a partir dele e essa será a entrada do algoritmo de propagação de rótulos.

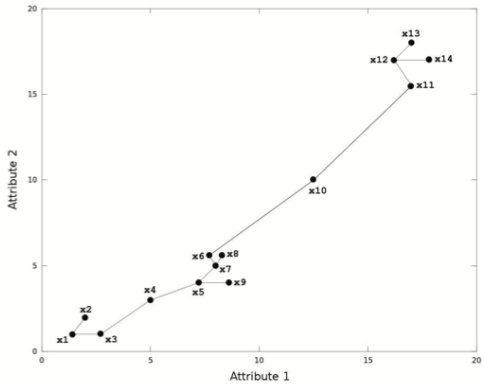


Figura 3.1 – Objetos de dados e arestas de uma AGM calculada sobre o espaço transformado de distâncias de alcance mútuo (CAMPELLO et al., 2015), p13.

3.3 Propagação de rótulos

Para propagação dos rótulos propostos foi utilizado o algoritmo de *Gaussian Random Fields* (GRF), em que, rotulados 10 pontos aleatórios da árvore geradora mínima do método de construção da seção anterior, exemplificada na imagem 3.2, ele seria capaz de difundir os rótulos para o conjunto de dados inteiro.

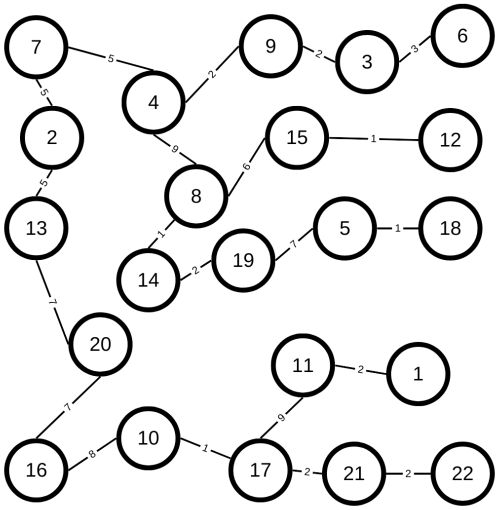


Figura 3.2 – Grafo de entrada para o algoritmo de propagação de rótulos.

Fonte: Elaborado pelo autor.

O algoritmo de propagação de rótulos começa com a inicialização dos parâmetros, incluindo a ordem dos objetos, os rótulos conhecidos e a matriz de pesos que representa as relações entre os objetos. Em seguida, é construída a matriz Laplaciana a partir da matriz de pesos, capturando as interações entre os objetos no conjunto de dados.

A ordem dos objetos é ajustada de acordo com a entrada fornecida, reordenando as linhas e colunas da matriz Laplaciana. Duas submatrizes são isoladas da matriz Laplaciana: uma que contém as relações entre objetos não rotulados e rotulados, e outra que contém apenas as relações entre objetos não rotulados. Um sistema linear é resolvido para obter os rótulos dos objetos não rotulados, levando em consideração a possível singularidade da matriz correspondente.

Os rótulos dos objetos rotulados são atribuídos diretamente aos resultados. Para os objetos não rotulados, os rótulos são atribuídos com base nos valores obtidos após a solução do sistema linear. Se o valor correspondente na solução for positivo, o objeto é rotulado como 1; se for negativo, o objeto é rotulado como -1. A função retorna uma lista que contém os rótulos atribuídos a todos os objetos, tanto rotulados quanto não rotulados, de acordo com a propagação de rótulos realizada pelo algoritmo.

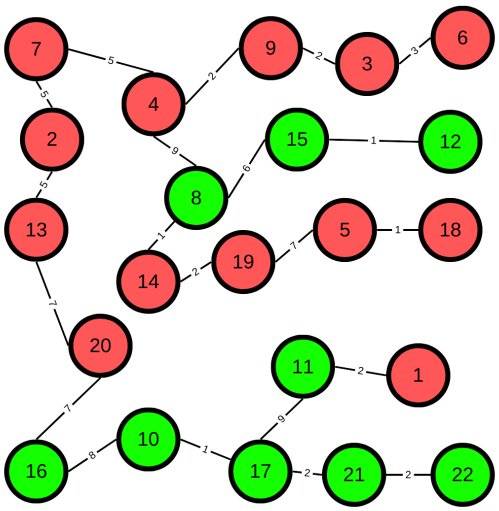
A entrada do algoritmo de GRF são 3 parâmetros computados anteriormente, disponíveis no Algoritmo 2 e que se referem a: 1) Ordem_objetos: Todos os objetos do *dataset* reordenados

de forma aos que foram rotulados aparecerem primeiro. 2) Yl: Uma lista dos 10 rótulos que foram aplicados no grafo. 3) W: A matriz de adjacência que vêm do grafo.

Algorithm 2 Algoritmo de Propagação de Rótulos

```
1: function PropagarRotulos(grafo, rotulos)
2:     resultado_rotulado ← Atribuir10RotulosGrafo(grafo)
3:     ordem_objetos ← ObjetosReordenadosComRotulos(grafo, resultado_rotulado)
4:     yl ← ValoresDos10Rotulos(resultadoRotulado)
5:     W ← TransformarEmMatrizDeAdjacencia(grafo)
6:     grf(ordem_objetos, yl, W)
7: end function
```

Figura 3.3 – Grafo rotulado após o algoritmo 2, sendo verde valores 1 e vermelho valores -1



Fonte: Elaborado pelo autor.

Através do grafo rotulado disponível na imagem 3.3 é possível extrair as informações disponíveis na tabela 3.3 e que servem de entrada para o algoritmo de Gaussian Random Fields.

Saída da função de propagação de rótulos																							
Yl	**Ordem_ Objetos**	**W**																					
1	8		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
1	10	**1**	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
1	15	**2**	0	0	0	0	0	0	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
1	16	**3**	0	0	0	0	0	3	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
1	21	**4**	0	0	0	0	0	0	5	9	2	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	1	**5**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	7	0	0	0	0
-1	4	**6**	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	7	**7**	0	5	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	13	**8**	0	0	0	9	0	0	0	0	0	0	0	0	0	1	6	0	0	0	0	0	0	0
-1	14	**9**	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	**10**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	1	0	0	0	0	0
	3	**11**	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0
	5	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	6	**13**	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0
	9	**14**	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0
	11	**15**	0	0	0	0	0	0	0	6	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	12	**16**	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	7	0	0
	17	**17**	0	0	0	0	0	0	0	0	0	1	9	0	0	0	0	0	0	0	0	0	2	0
	18	**18**	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	19	**19**	0	0	0	0	7	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
	20	**20**	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	7	0	0	0	0	0	0
	22	**21**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	2
		22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0

Tabela 3.2 – Tabela com a entrada do algoritmo de GRF, sendo yl os 10 rótulos do dataset, o Ordem_Objetos todos os objetos do dataset ordenados de acordo com seus rótulos e W a matriz de adjacência encontrada

Fonte: Elaborado pelo autor.

3.4 Medidas de Desempenho

A medida de desempenho a ser utilizada seguirá o mesmo padrão utilizado no trabalho base de Sousa, Rezende e Batista (2013) que é a taxa de erro médio e desvio padrão, o último não sendo classificatório para melhor desempenho ou não. A taxa de erro média escolhida é a menor obtida por uma combinação de um algoritmo SSL, um método de construção de grafo e um conjunto de dados para todos os valores de parâmetros iterado 50 vezes. A taxa de erro médio é calculado como a porcentagem do número de predições incorretas diante de todas as predições. Já o desvio padrão é uma medida de dispersão que quantifica o quanto os valores de um conjunto de dados tendem a se afastar da média.

4 Resultados

O ambiente computacional utilizado para execução dos algoritmos foi um processador Apple M2 PRO, 16 GB de memória RAM sob o sistema operacional macOS Ventura 13.2.1. O método proposto foi implementado usando Python versão 3.11.4 através do Jupyter Notebook. Devido a algumas possíveis aleatoriedades cada conjunto de dados rotulado foi executado de forma independente 50 vezes variando quais instâncias dos conjuntos de dados seriam rotuladas.

Novamente, os conjuntos de dados utilizados estão disponíveis na biblioteca sslbookdata[1] e são compostos por 6 *datasets* com características diversas. E o código utilizado pode ser encontrado em repositório público[2] no Github.

4.1 Resultados obtidos

Para o dataset USPS os melhores parâmetros obtidos foram o número mínimo de pontos para cálculo de core distance igual a 2, sem redução de dimensionalidade do dataset e métrica Minkowski. Alcançando uma taxa de erro médio de 17.8 e desvio padrão de 0.03.

O dataset DIGIT utilizou 5 pontos, com redução de dimensionalidade e métrica do Cosseno obtendo 17.67 de taxa de erro médio e 0.09 de desvio padrão. G-241C utilizou também 2 pontos, reduziu a dimensionalidade e também usou a métrica de Minkowski obtendo uma taxa de erro médio de 42.07 e desvio padrão de 0.02. O G-241N utilizou 4 pontos, reduziu dimensionalidade, teve mesma métrica que o anterior e obteu 45.87 de taxa de erro médio com 0.02 de desvio padrão. O dataset COIL utilizou 3 pontos, não reduziu a dimensionalidade, utilizou a mesma métrica da anterior e obteve uma taxa de erro médio de 41.13 com devsio padrão de 0.02.

Para o TEXT, último *dataset* utilizado, não foi feita a redução de dimensionalidade por ter 11960 atributos e assim como Sousa, Rezende e Batista (2013) buscamos manter essa característica de alto grau de complexidade, utilizamos 2 pontos, métrica euclidiana e obtivemos 41.13 de taxa média de erro e 0.09 de desvio padrão.

Todas as informações de resultados obtidos podem ser encontrados na tabela 4.1 que se segue.

[1] https://pypi.org/project/sslbookdata/
[2] https://github.com/rnlobao/graph-construction-SSL

Dataset	**Número de Pontos**	**Métrica**	**Reduziu Dimensionalidade**	**Taxa de erro média**	**Desvio Padrão**
USPS	2	Minkowski	Não	17.8	0.03
DIGIT	5	Cosseno	Sim	17.67	0.09
G-241C	2	Minkowski	Sim	42.07	0.02
G-241N	4	Minkowski	Sim	45.87	0.02
COIL	3	Minkowski	Não	41.13	0.02
TEXT	2	Euclidiana	Não	41.13	0.09

Tabela 4.1 – Tabela com os parâmetros usados no algoritmo e os resultados.

Fonte: Elaborado pelo autor.

4.1.1 Comparações com o estado da arte

Atualmente, os melhores resultados quanto à utilização de grafos para aprendizado semissupervisionado são de Sousa, Rezende e Batista (2013). Em seu trabalho usando o algoritmo de GRF ele encontrou os seguintes resultados ótimos para cada tipo de *dataset* podendo ser visualizados na tabela 4.2:

Dataset:	USPS	COIL	DIGIT	G-241N	G-241C	TEXT
Método de contrução	mutKNN-RBF	mutKNN-RBF	mutKNN-RBF	symKNN-RBF	symKNN-HM	mutKNN-RBF
Resultado	9.75 (4.50)	35.07 (3.82)	9.35 (4.51)	46.12 (7.61)	46.19 (7.25)	37.51 (6.85)

Tabela 4.2 – Resultados do atual estado da arte.

Fonte: Elaborado pelo autor.

Tendo esses valores em vista é possível averiguar que o algoritmo proposto com a contrução do grafo baseado na AGM do grafo de alcançabilidade mútua do algoritmo HDBSCAN* alcançou melhores resultados em 2 conjuntos de dados G-241C e G-241N, como pode ser visto na tabela 4.1.1 que compara os dois resultados:

Dataset	Resultado Atual Estado Da Arte	Resultados obtidos
USPS	**9.75 (4.5)**	17.80 (0.03)
COIL	**35.07 (3.82)**	41.13 (0.02)
DIGIT	**9.35 (4.51)**	17.67 (0.09)
G-241N	46.12 (7.61)	**45.87 (0.02)**
G-241C	46.19 (7.25)	**42.07 (0.02)**
TEXT	**37.51 (6.85)**	41.13 (0.09)

Tabela 4.3 – Comparação dos resultados de taxa de erro média com desvio padrão dessa pesquisa.

Fonte: Elaborado pelo autor.

Diante das informações dadas, é possível concluir que o método proposto foi capaz de produzir bons resultados com baixo desvio padrão para 2 *datasets* dos 6 propostos por Chapelle, Scholkopf e Zien (2006) em seu trabalho sobre algoritmos semissupervisionados e também utilizados por Sousa, Rezende e Batista (2013) em sua pesquisa. Os *datasets* em questão são o G-241C e G-241N que possuem características de extração de pontos de gaussianas isotrópicas de variância unitária para criar as classes, que se referem ao processo de gerar pontos, ou dados, a partir de distribuições gaussianas com variância em todas as direções (HAMMERSLEY; NELDER, 1955).

No contexto mencionado, a extração de pontos dessas gaussianas isotrópicas é realizada para criar classes ou grupos de dados. O processo pode envolver a geração de pontos aleatórios seguindo as características dessas gaussianas, e esses pontos são então atribuídos a diferentes classes ou *clusters*. Para ilustrar com um exemplo mais concreto, vamos supor duas gaussianas isotrópicas com centros a uma distância específica uma da outra. Ao extrair pontos dessas gaussianas, criam-se duas classes distintas, onde os pontos de cada classe são gerados a partir de uma gaussiana isotrópica com variância unitária. Isso pode ser parte do processo de criação de conjuntos de dados para fins de experimentação em aprendizado de máquina ou estudos estatísticos. Uma projeção bidimensional dos *datasets* está sendo mostrado na figura 4.1

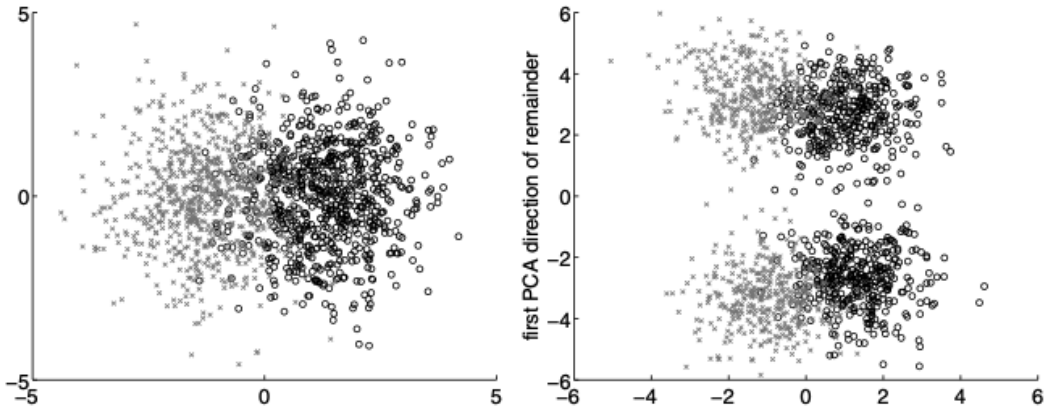


Figura 4.1 – Projeção 2d dos datasets G-241C e G-241N respectivamente (CHAPELLE; SCHOLKOPF; ZIEN, 2006), p378.

Ambos os *datasets* possuem duas classes distintas, têm distância entre o centro da gaussiana já definidas, a direção dos pontos é aleatória e o rótulo de cada classe representa a gaussiana a partir da qual o ponto foi desenhado. Ademais, uma outra característica importante difere os dois *datasets* com resultados superiores dos outros.

Essa característica é a densidade da base de dados, que foi calculada utilizando a variância vista na equação 4.1. Uma variância 1 sugere que, em média, os pontos de dados não estão muito longe da média, mostrando, caso tenha poucos *outliers*, que o *dataset* é denso.

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1} \tag{4.1}$$

Onde:

s^2 é a variância amostral.

n é o número de observações na amostra.

X_i são os valores individuais na amostra.

$\bar{X}$ é a média da amostra, calculada como $\frac{\sum_{i=1}^{n} X_i}{n}$.

Tendo isso em vista, a variância e número de *outliers* encontrados, utilizando o método de LOF com 20 vizinhos próximos, explicado na seção 2.4, para cada *dataset* pode ser visto na tabela 4.4:

Dataset	**Outliers**	**Variância**
G-241C	0	1.0
G-241N	0	1.0
DIGIT	0	0.5
USPS	1	0.86
COIL	87	5894.74
TEXT	368	8.34

Tabela 4.4 – Tabela com número de *outliers* e variância dos *datasets*.

Fonte: Elaborado pelo autor.

5 Conclusão

A rotulagem de conjuntos de dados parcialmente preenchidos é um problema conhecido no meio científico. Ele é amplamente útil em processos industriais e contextos acadêmicos, onde gostaríamos de rotular objetos em um conjunto de dados que, recolher os dados é mais fácil que classificá-lo de acordo com alguma classe. Neste trabalho foi proposto um método de rotulagem de conjuntos de dados semissupervisionados baseado em uma AGM do grafo de alcançabilidade mútua do algoritmo de HDBSCAN*.

Experimentos foram realizados considerando os 6 conjuntos de dados mais utilizados pelo meio de testes do aprendizado semissupervisionado, utilizando 50 iterações dos testes e variando os 3 parâmetros principais que eram o número mínimo de pontos para computar uma *Core Distance*, redução ou não da dimensionalidade dos conjuntos de dados, exceto TEXT, e a métrica utilizada para calcular a distância dos objetos dos conjuntos de dados. Dos *datasets* testados observa-se desempenho melhor, considerando taxa de erro média, que o atual estado da arte em 2 deles; G-241C e G-241N.

Conclui-se que o método proposto supera a pesquisa de Sousa, Rezende e Batista (2013) em dois *datasets* com características de padronização de dimensões, duas classes e com conjuntos de dados que possuem variância igual a 1, significando que os dados não são muito dispersos, e que, os pontos de dados não estão muito longe da média, podendo ser tratados dessa forma por não possuírem muitos *outliers*. Ou seja, o método proposto funciona melhor que o estado da arte para situações em que o conjunto de dados tem classificação binária e dados densos.

5.1 Trabalhos futuros

Próximas etapas da pesquisa podem se concentrar em trazer os resultados de taxa de erro médio e de desvio padrão utilizando as outras formas de propagação de rótulos citados por Sousa, Rezende e Batista (2013) sendo elas a LGC, LapRLS, LapSVM e RMGT. Esses algoritmos de propagação de rótulos baseados em grafos teriam como entrada o AGM do grafo de alcançabilidade mútua do HDBSCAN* proposto no capítulo 3.2.

Referências

ANSCOMBE, F. J. Graphs in statistical analysis. *The american statistician*, Taylor & Francis, v. 27, n. 1, p. 17–21, 1973.

ARAÚJO, B.; ZHAO, L. Data heterogeneity consideration in semi-supervised learning. *Expert Syst. Appl.*, v. 45, p. 234–247, 2016. Disponível em: <https://doi.org/10.1016/j.eswa.2015.09.026>.

CAMPELLO, R. J.; MOULAVI, D.; ZIMEK, A.; SANDER, J. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, ACM New York, NY, USA, v. 10, n. 1, p. 1–51, 2015.

CASTRO, L. de; FERRARI, D. Introdução à mineração de dados: Conceitos básicos. *Algoritmos e Aplicações, Saraiva*, 2016.

CHAPELLE, O.; SCHOLKOPF, B.; ZIEN, A. Semi-supervised learning. 2006. *Cambridge, Massachusettes: The MIT Press View Article*, v. 2, 2006.

DAVIES, L.; GATHER, U. The identification of multiple outliers. *Journal of the American Statistical Association*, v. 88, p. 782–792, 1993.

DING, T.; ZHANG, M.; HE, D. mei. A network intrusion detection algorithm based on outlier mining. p. 1229–1236, 2017.

ERDöS, P. Graph theory and probability. *Canadian Journal of Mathematics*, v. 11, p. 276–280, 1959.

FACELI, K.; LORENA, A. C.; GAMA, J.; ALMEIDA, T. A. d.; CARVALHO, A. C. P. d. L. F. d. Inteligência artificial: uma abordagem de aprendizado de máquina. 2021.

GOLDBARG, M.; GOLDBARG, E. *Grafos: conceitos, algoritmos e aplicações*. [S.l.]: Elsevier, 2012.

HAMMERSLEY, J.; NELDER, J. Sampling from an isotropic gaussian process. *Mathematical Proceedings of the Cambridge Philosophical Society*, v. 51, p. 652 – 662, 1955.

HINTON, G. E.; SEJNOWSKI, T. Unsupervised learning. p. 1304, 1999.

ISCEN, A.; TOLIAS, G.; AVRITHIS, Y.; CHUM, O. Label propagation for deep semi-supervised learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2019.

KOTSIANTIS, S.; ZAHARAKIS, I.; PINTELAS, P. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, v. 26, p. 159–190, 2006.

LIU, W.; CHANG, S.-F. Robust multi-class transductive learning with graphs. In: IEEE. *2009 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.], 2009. p. 381–388.

OZAKI, K.; SHIMBO, M.; KOMACHI, M.; MATSUMOTO, Y. Using the mutual k-nearest neighbor graphs for semi-supervised classification on natural language data. In: *Proceedings of the fifteenth conference on computational natural language learning*. [S.l.: s.n.], 2011. p. 154–162.

SOMANATH, G.; ROHITH, M.; KAMBHAMETTU, C. Vadana: A dense dataset for facial image analysis. *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, p. 2175–2182, 2011.

SOUSA, C. A. R. de; REZENDE, S. O.; BATISTA, G. E. Influence of graph construction on semi-supervised learning. In: SPRINGER. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*. [S.l.], 2013. p. 160–175.

SUN, S.; HUANG, R. An adaptive k-nearest neighbor algorithm. In: IEEE. *2010 seventh international conference on fuzzy systems and knowledge discovery*. [S.l.], 2010. v. 1, p. 91–94.

TONG, S.; KOLLER, D. Restricted bayes optimal classifiers. In: CITESEER. *AAAI/IAAI*. [S.l.], 2000. p. 658–664.

ZHANG, S.; TONG, H.; XU, J.; MACIEJEWSKI, R. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, SpringerOpen, v. 6, n. 1, p. 1–23, 2019.

ZHU, X.; GHAHRAMANI, Z.; LAFFERTY, J. D. Semi-supervised learning using gaussian fields and harmonic functions. In: *Proceedings of the 20th International conference on Machine learning (ICML-03)*. [S.l.: s.n.], 2003. p. 912–919.

ZHU, X.; GOLDBERG, A. B. *Introduction to semi-supervised learning*. [S.l.]: Springer Nature, 2022.