UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Rio Claro

Leonardo Tadeu Lopes

Métodos de Agrupamento baseados em Informações de Ranqueamento

Rio Claro

2021

Leonardo Tadeu Lopes

Métodos de Agrupamento baseados em Informações de Ranqueamento

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista "Júlio de Mesquita Filho", como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Orientador: Profº. Drº. Daniel Carlos Guimarães Pedronette

Rio Claro

2021

L864m

Lopes, Leonardo Tadeu
Métodos de Agrupamento baseados em Informações de Ranqueamento / Leonardo Tadeu Lopes. -- Rio Claro, 2021
94 f.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Geociências e Ciências Exatas, Rio Claro
Orientador: Daniel Carlos Guimarães Pedronette

1. Ciência da Computação. 2. Métodos de Agrupamento. 3. Aprendizado Não-supervisionado. 4. Aprendizado Auto-supervisionado. 5. Modelo de Ranqueamento. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Geociências e Ciências Exatas, Rio Claro. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Leonardo Tadeu Lopes

Métodos de Agrupamento baseados em Informações de Ranqueamento

Comissão Examinadora

- Prof. Dr. Daniel Carlos Guimarães Pedronette (Orientador)
  Departamento de Estatística, Matemática Aplicada e Computação
  Universidade Estadual Paulista - UNESP
- Prof. Dr. Guilherme Palermo Coelho
  Faculdade de Técnologia
  Universidade de Campinas - UNICAMP
- Profa. Dra. Verônica Oliveira de Carvalho
  Departamento de Estatística, Matemática Aplicada e Computação
  Universidade Estadual Paulista - UNESP

Resultado: Aprovado

Rio Claro - SP

26 de Agosto de 2021

Este trabalho é dedicado à toda minha família, em especial às grandes mulheres da minha vida, Maria José, Adolfina, Elisângela e à minha futura esposa, Lígia.

Agradecimentos

Primeiramente, agradeço à vida e a bela maneira com que seus caminhos, ora tortuosos, se encontram e se alinham de maneira próspera. Agradeço minha família, responsável pela importante base da minha personalidade. Sendo uma pessoa iluminada, obtive a dádiva de receber três mães maravilhosas em minha vida. À minha mãe, Maria José, obrigado por ser mãe, pai e fazer tudo em seu alcance pelo meu futuro. À minha mãe de coração, Adolfina, obrigado por me dar tanto amor verdadeiro. À minha madrinha, Elisângela, obrigado por sempre ficar ao meu lado, sem você eu não teria conseguido.

Agradeço também ao meu padrinho e sua esposa, Kleber e Cibele, obrigado por incentivarem meu amor pela tecnologia desde sempre. Ao meu cunhado Emerson, obrigado por ser exemplo de família. Ao meu primo, afilhado e amigo Enrico, obrigado por ser minha força na busca de me tornar uma pessoa melhor sempre. Agradeço em especial ao meu pai de coração, Benedito, você é meu maior exemplo. Obrigado por ser essa figura tão importante em minha vida.

Agradeço também aos meus sogros, Reynaldo e Ivone. Obrigado por terem me acolhido e por me apoiarem em todas as etapas desse processo.

Agradeço aos amigos que foram importantes nesta trajetória. À minha eterna república Várzea que me deu os melhores amigos que poderia ter. Em especial, agradeço a três grandes pessoas que foram inspiração para essa jornada: Juliana Oler, Northon Penteado e Claudio Santos. Vocês me mostraram ser possível e assim o foi.

Agradeço a meu orientador Daniel Pedronette, pela grande parceria e suporte durante a elaboração deste trabalho, e a todos os grandes professores com os quais tive o prazer de aprender. Também agradeço ao Laboratório de Inteligência Artificial Aplicada ao Petróleo (LIAAP) e a equipe com quem tive o prazer de trabalhar e aprender muito.

Por fim, agradeço a minha futura esposa Lígia. Você é força, inspiração, apoio, parceria, porto seguro e tantos outros significados que não podem ser expressos em palavras. Obrigado por dividir a vida e seus sonhos comigo.

Resumo

As contantes evoluções tecnológicas realizadas nas últimas décadas, nos mais diversos domínios do conhecimento, possibilitaram a produção de volume massivo de dados e grande parcela deste volume é armazenada de maneira digital. Neste cenário, há uma grande demanda por métodos de aprendizado de máquina que consigam realizar análise de dados de forma automática para diferentes tarefas. Porém, a criação de rótulos de treinamento exige grande esforço humano, sendo escassos, inexatos ou até mesmo indisponíveis em diversas áreas de aplicação. Visando sobrepor esta dificuldade, métodos de aprendizado semi-supervisionado, auto-supervisionado e não-supervisionado utilizam as informações disponíveis de maneiras únicas, visando aprender a partir de poucos ou nenhum rótulo. As técnicas de agrupamento são importantes métodos não-supervisionados que buscam separar um conjunto de dados em agrupamentos disjuntos, a partir da análise da similaridade ou distância entre seus elementos. Esta categoria de algoritmos é amplamente aplicada em diversas áreas do reconhecimento de padrões e novos métodos são propostos constantemente, demonstrando a demanda para novas abordagens. De maneira análoga, as técnicas de manifold learning são métodos não-supervisionados que exploram a estrutura dos dados visando obter melhores relações de similaridade entre os elementos. Apesar de possuírem objetivos similares, métodos de agrupamento que exploram técnicas de manifold learning não são comuns na literatura. Neste trabalho, duas técnicas de manifold learning foram aplicadas para criação de métodos de agrupamento com a utilização de grafos, componentes conexas, hipergrafos e redes neurais baseados em grafos. As metodologias propostas foram avaliadas em uma variedade de conjuntos de dados e comparadas com métodos clássicos e recentes da literatura. Além disso, análises visuais foram exploradas para ilustrar os efeitos das abordagens de manifold learning utilizadas. Os resultados obtidos são promissores, sendo comparáveis ou superiores em todos os cenários avaliados.

Palavras-chave: métodos de agrupamento. aprendizado de máquina. aprendizado não-supervisionado. aprendizado auto-supervisionado. manifold learning.

Abstract

The constant technological evolutions from the last decade, on the most diverse knowledge fields, enabled the production of massive amounts of information, from which most is stored in digital format. In this scenario, the demand for machine learning methods which can automatically perform data analysis on different task has grown. However, the creation of labels required for training those methods requires huge human effort, being scarce, inaccurate or even unavailable in several application areas. Aiming to surpass this challenge, semi-supervised, auto-supervised and unsupervised methods exploits the available information from unique perspectives while attempting to learn from few or no labels from the input data. Clustering techniques are important unsupervised methods that seek to separate a set of data into disjoint groups, based on the analysis of the similarity or distance between its elements. Additionally, clustering algorithms are widely applied in several areas of pattern recognition and novel methods are constantly proposed, demonstrating the demand for new approaches. In a similar approach, manifold learning techniques are unsupervised techniques that explore the data structure in order to obtain better similarity relationships between elements. However, despite having similar objectives, clustering methods that explore manifold learning techniques are not common in the literature. In this work, two manifold learning techniques were applied to create clustering methods using graphs, connected components, hypergraphs and graph-based neural networks. The proposed methodologies were evaluated on a variety of datasets and compared with classical and novel methods from the literature. Furthermore, visual analyzes were explored to illustrate the effects of the chosen manifold learning approaches. The results obtained are promising, being comparable or superior in all evaluated scenarios.

Keywords: clustering. machine learning. unsupervised learning. semi-supervised learning. manifold learning.

Lista de Ilustrações

Lista de Tabelas

Lista de Abreviaturas e Siglas

ACC	Acurácia
APPNP	Approximate Personalized Propagation of Neural Predictions
ARI	Adjusted Rand Index
ASC	Aspect Shape Context
Auto-encoders	Redes Neurais Codificadoras/Decodificadoras
BIRCH	Balanced Iterative Reducing and Clustering Using Hierarchies
C-ReckNN	Clustering through Reciprocal kNN Graph and Connected Components
CFD	Contour Feature Descriptor
CURE	Clustering Using Representatives
CLARA	Clustering Large Applications
DBSCAN	Density-Based Spatial Clustering of Applications With Noise
DENCUE	Density-based Clustering
EM-Clustering	Expectation Maximization Clustering
FINCH	First Integer Neighbor Clustering Hierarchy
FSSC	Fast Self-Supervised Clustering With Anchor Graph
FSSF	Fast Semi-Supervised Framework
GCN	Graph Convolutional Network
GDBSCAN	Generalized Density-Based Spatial Clustering of Applications with Noise
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
LHRR	Log-based Hypergraph of Ranking References
MI	Mutual Information
MLP	Multi Layer Perceptron

MST	Minimal Spanning tree
MUNEC	Mutual Neighbor-based Clustering Algorithm
NMI	Normalized Mutual Information
PAM	Particioning Around Medoids
ResNet	Residual Neural Network
RI	Rand Index
RNA	Redes Neurais Artificiais
RNC	Redes Neurais Competitivas
RNP	Redes Neurais Profundas
ROCK	Robust Clustering Using Links
S^2ConvSCN	Self-Supervised Convolutional Subspace Clustering Network
SDCN	Structural Deep Clustering Network
SGC	Simple Graph Convolution
SGCC	Self-supervised Graph Convolutional Clustering
SOM	Self Organizing Maps
Spectral	Spectral Clustering

Lista de símbolos

$\mathbf{A}$	Matriz de afinidades adjacências.
$\mathcal{C}$	Coleção de elementos.
$\mathbf{c}_i$	Componente conexa de índice i.
c_i	Agrupamento ao qual pertence o elemento com índice i.
c	Número de agrupamentos contidos no conjunto de dados.
$\mathcal{D}$	Matriz de afinidades $\mathbf{A}$ normalizada para definição da GCN.
E	Conjunto de arestas de um grafo ou hiperarestas de um hipergrafo.
e_i	Uma hiperaresta de índice i, relacionada ao elemento o_i.
G	Grafo.
G_r	Grafo de vizinhos recíprocos.
G_h	Hipergrafo.
$\mathbf{H}_b$	Matriz de incidência binária do hipergrafo.
$\mathbf{H}$	Matriz de incidência do hipergrafo.
$\mathbf{H}_s$	Matriz de incidência das hiperarestas de agrupamento.
$\mathbf{h}_i$	Linha i da matriz $\mathbf{H}_s$, representando as características do agrupamento i.
k	Tamanho do conjunto de vizinhança.
L	Tamanho das listas ranqueadas.
n	Tamanho da coleção.
$\mathcal{N}$	Conjunto de vizinhança.
$\mathcal{N}_r$	Conjunto de vizinhança recíproca.
o_i	Elemento pertencente a coleção, cujo índice é i.
$\mathfrak{R}$	Conjunto de representantes.
r_i	Representante de índice i.

ρ	Utilizado para indicar métricas de distância ou similaridade entre elementos.
$\mathcal{S}$	Conjunto de agrupamentos extraído do conjunto de dados.
$\mathcal{S}_i$	Grupo do conjunto de agrupamentos, com índice i.
$\mathcal{P}$	Conjunto de componentes conexas.
T, t	Número de iterações.
$\mathcal{T}$	Conjunto das listas ranqueadas de todas os elementos de uma coleção.
τ_i	Lista ranqueada do elemento de índice i.
$\tau_q(i)$	Posição do elemento i na lista ranqueada do elemento de consulta q.
v_i	Vértice de índice i.
V	Conjunto de vértices de um grafo ou hipergrafo.
w	Métrica de seleção de combinações de ranqueadores.
$\mathbf{X}$	Vetores de características para todos os elementos do conjunto e dados.
$\mathbf{x}_i$	Vetor de características do elemento com índice i
$\mathbf{W}$	Matriz de similaridades ou matriz de pesos (GCN).
$\mathcal{Y}$	Conjunto de rótulos.
$\mathbf{Z}$	Matriz de representações aprendida pela GCN.

Sumário

1 Introdução

Os notáveis avanços tecnológicos realizados nas últimas décadas em diversos domínios, como a internet, os dispositivos móveis, sistemas de vigilância, e a disponibilidade de servidores de grande poder de processamento e armazenamento desencadearam a produção de um volume massivo de dados diariamente [Jain, 2010]. A geração de dados ocorre de forma ubíqua por múltiplos agentes, incluindo desde indivíduos em aplicações pessoais a corporações, que também utilizam diversas técnicas para organização das informações produzidas em suas operações, visando aproveitá-las em tomadas de decisões futuras. Desta maneira, estima-se que a produção de informação diária de nossa sociedade passe o volume de mil Exabytes e deva aumentar nos próximos anos [Benabdellah; Benghabrit; Bouhaddou, 2019]. A maior parcela deste volume é armazenado de maneira digital, o que cria uma grande demanda de métodos automáticos para análise de dados em tarefas de classificação, recuperação de informação e agrupamento.

Neste contexto, dentre os vários propósitos da área de reconhecimento de padrões, pode-se destacar a análise de dados com foco em tarefas predição: dado um conjunto de dados de treinamento, é desejado predizer informações para amostras futuras desconhecidas. Esta tarefa também é comumente referenciada como *aprendizado* [Jain, 2010]. A categoria mais comum de aprendizado é o *aprendizado supervisionado*, em que uma grande quantidade de exemplos já rotulados é utilizado como conjunto de treinamento para métodos que buscam aprender relações entre os elementos fornecidos e seus respectivos rótulos, sendo capazes que aplicar este aprendizado em novos exemplos no futuro. Esta categoria de aprendizado logrou a obtenção de resultados formidáveis em diversas áreas de conhecimento, como computação visual e processamento de linguagem natural, principalmente após o desenvolvimento de *redes neurais de aprendizado profundo* [LeCun; Bengio; Hinton, 2015].

Contudo, a obtenção de rótulos para treinamento exige grande esforço humano, especialmente em aplicações reais. Quando disponíveis, os rótulos são escassos, inexatos ou até mesmo inexistentes em diversas áreas de aplicação [Xie et al., 2021]. Neste cenário, o *aprendizado semi-supervisionado* explora quantidades limitadas de rótulos visando propagar as informações aprendidas para conjuntos maiores não-rotulados. Dados representados em grafos são largamente utilizadas nesta categoria, principalmente após o desenvolvimento de *redes neurais baseadas em grafos* e, mais recentemente, *redes neurais convolucionais baseadas em grafos* [Hoi; Liu; Chang, 2010]. Também abordando a ausência de dados rotulados, o *aprendizado auto-supervisionado* consiste em uma vertente recente de métodos que realizam treinados supervisionados utilizando rótulos gerados automaticamente a partir dos próprios dados analisados. Redes convolucionais baseadas em grafo também são frequentemente utilizadas nesta categoria de métodos [Xie et al., 2021].

Por fim, o *aprendizado não-supervisionado* tem como objetivo extrair conhecimento de um conjunto de dados, sobre o qual não existem informações rotuladas. Neste cenário, somente a relação entre os elementos está disponível, sendo explorada a partir de diversas técnicas e análises de similaridade ou distância [Rui Xu; Wunsch, 2005]. Um dos principais métodos de aprendizado não-supervisionado é o *agrupamento*, em que um conjunto é particionado em grupos de elementos disjuntos somente com base nas relações disponíveis entre os elementos e sem nenhum conhecimento prévio sobre os mesmos. Posteriormente, os métodos de agrupamento também são capazes de prever a alocação de um novo elemento, não presente no conjunto original, em um dos grupos formados anteriormente [Saxena et al., 2017].

As técnicas de agrupamento são componentes essenciais de diversas áreas de análises de dados ou aprendizado de máquina e diversas técnicas foram desenvolvidas ao longo das últimas décadas [Saxena et al., 2017; Rui Xu; Wunsch, 2005]. Entretanto, não há uma técnica universal, que pode ser aplicada de forma eficaz a qualquer problema de forma indiscriminada. Dessa forma, a escolha do método de agrupamento é frequentemente direcionada pelo conjunto de dados do qual se deseja obter informação [Ros; Guillaume, 2019]. Também, assim como outras técnicas de reconhecimento de padrões, os métodos de agrupamento baseiam-se nos conjuntos de características e na escolha de uma métrica que irá guiar a separação dos dados a partir destas características. Esta separação é diretamente afetada pela qualidade das informações presentes nas características e pela relação desenvolvida pela métrica escolhida. Neste cenário, há numerosos e relevantes desafios de pesquisa na área. Apesar de serem tema de estudo há décadas, os métodos de agrupamento ainda são amplamente investigados e o desenvolvimento na área é contínuo [Saxena et al., 2017].

De maneira mais formal, o objetivo principal dos métodos de agrupamento é encontrar uma separação ótima para o conjunto explorado a partir da análise da relação de similaridade entre os elementos. Diversas técnicas tem sido exploradas recentemente para atingir esse objetivo em novos métodos de agrupamento, dentre elas podemos citar a análise de vizinhança recíproca [Ros; Guillaume, 2019; Ros et al., 2020; Sarfraz; Sharma; Stiefelhagen, 2019], a utilização de redes neurais de aprendizado profundo [Darlow; Storkey, 2020; Huang; Zhu, 2020], aplicação de redes convolucionais baseadas em grafo [Bo et al., 2020; Bianchi; Grattarola; Alippi, 2020; Tsitsulin et al., 2020] e abordagens auto-supervisionadas [Sadeghi; Armanfard, 2021; Wang et al., 2021; Zhang et al., 2019]. Em comum entre as diversificadas abordagens, destaca-se o uso e desafios associados à definição de similaridade entre elementos.

Em direção análoga, técnicas de *manifold learning* são métodos de aprendizado não-supervisionado que buscam explorar a estrutura dos dados para obter melhores relações de similaridade entre os elementos, principalmente utilizadas em tarefas de recuperação

de informação. Dentre as diversas abordagens utilizadas para *manifold learning*, técnicas baseadas em ranqueamento visam explorar a relação de similaridade entre os elementos codificadas em formato de listas ranqueadas, com o objetivo de calcular novas medidas de similaridade baseadas em relações mais globais do conjunto [Pedronette et al., 2019; Pedronette; Valem; Torres, 2021]. Estas novas medidas podem codificar informações valiosas para agrupar conjuntos de dados com maior eficácia. Todavia, técnicas de agrupamento que explorem os métodos de *manifold learning* baseado em ranqueamento não são comuns na literatura e seu uso apresenta uma relevante oportunidade de pesquisa.

Neste cenário, este trabalho tem como objetivo central investigar a utilização de métodos de *manifold learning* baseados em ranqueamento para o desenvolvimento de novos métodos de agrupamento. A capacidade dos métodos de *manifold learning* em estabelecer relações de similaridade mais eficazes é explorada para a geração de agrupamentos também mais eficazes. As principais contribuições contidas nessa dissertação são discutidas a seguir:

- Uma revisão bibliográfica sobre as técnicas de agrupamento é apresentada, discutindo os principais conceitos, abordagens, métodos clássicos e métricas de avaliação;
- Dois métodos de *manifold learning* da literatura [Pedronette; Gonçalves; Guilherme, 2018; Pedronette et al., 2019] são explorados para criação de novos métodos de agrupamento, não somente utilizando as relações de similaridade aprimoradas, como também as estruturas utilizadas para analisar os conjuntos de dados;
- O método *Clustering through Reciprocal kNN Graph and Connected Components (C-ReckNN)* [Lopes. et al., 2020] é proposto para a formação de agrupamentos por meio do uso de componentes conexas presentes em grafos de vizinhança recíproca;
- É proposto um novo método de agrupamento baseado em aprendizado auto-supervisionado, nomeado *Self-supervised Graph Convolutional Clustering (SGCC)*. A abordagem proposta utiliza um algoritmo de *manifold learning* baseado em hipergrafos para a criação de agrupamentos iniciais de alta confiança, utilizados como rótulos de treinamento para uma rede convolucional baseada em grafos;
- As duas metodologias propostas são avaliadas experimentalmente e comparadas com métodos clássicos e recentes da literatura, considerando diversos conjuntos de dados. Análises visuais também são conduzidas para ilustrar os efeitos da aplicação de técnicas de *manifold learning* em tarefas de agrupamento.

É proposto um novo método de agrupamento baseado em aprendizado auto-supervisionado, nomeado *Self-supervised Graph Convolutional Clustering (SGCC)*. A abordagem proposta utiliza um algoritmo de *manifold learning* baseado em hipergrafos para a criação de

agrupamentos iniciais de alta confiança, utilizados como rótulos de treinamento para uma rede convolucional baseada em g

O restante da dissertação está organizado da seguinte maneira: o Capítulo 2 apresenta a fundamentação teórica sobre os métodos de agrupamento, modelo de ranqueamento e técnicas de *manifold learning*, além de apresentar trabalhos recentes relacionados com as metodologias propostas. O Capítulo 3 apresenta o método de agrupamento *C-ReckNN* e o Capítulo 4 apresenta o método *SGCC*. O Capítulo 5 apresenta a avaliação experimental conduzida para aferir a eficácia de ambas as abordagens propostas. As considerações finais e trabalhos futuros são elencadas no Capítulo 6.

2 Fundamentação Teórica e Trabalhos Relacionados

Este capítulo discute aspectos de fundamentação teórica referentes aos principais temas explorados no decorrer deste trabalho. Inicialmente, a Seção 2.1 apresenta a definição e as características dos métodos de agrupamento. A Seção 2.2 descreve as categorias de métodos de agrupamento existentes, elencando abordagens e métodos clássicos das mesmas. A Seção 2.3 define métricas de avaliação utilizadas para verificar a qualidade dos grupos obtidos por estes métodos. A Seção 2.4 introduz o modelo de ranqueamento utilizado como base para as metodologias propostas, enquanto a Seção 2.5 introduz os métodos de *manifold learning*, técnicas de aprendizado não-supervisionado exploradas para criação de novos métodos de agrupamento. Por fim, a Seção 2.6 discute trabalhos relacionados recentes.

2.1 Aspectos Conceituais

As técnicas de agrupamento, *clustering* em inglês, são métodos de aprendizado não-supervisionado que visam separar um dado conjunto de itens em subconjuntos disjuntos, com o objetivo de minimizar uma medida de similaridade dos elementos presentes em cada agrupamento e maximizar esta mesma medida entre os agrupamentos [Benabdellah; Benghabrit; Bouhaddou, 2019; Saxena et al., 2017; Jain; Murty; Flynn, 1999]. Desta maneira, uma técnica de agrupamento tenta extrair os grupos naturais presentes em um conjunto de dados [Jain, 2010]. Esta extração pode ser realizada pelo uso de diversas técnicas e diferentes conjuntos de dados podem ter melhores resultados com diferentes técnicas de agrupamento [Rui Xu; Wunsch, 2005].

A área de estudo dos métodos de agrupamento é uma área importante do aprendizado de máquina, pois se propõe a analisar e separar dados sem nenhuma informação prévias sobre eles, diferentemente de outras técnicas de classificação supervisionada [Saxena et al., 2017]. Os estudos na área são abundantes e aplicados em diferentes frentes. A Figura 1 apresenta uma análise de trabalhos publicados na área entre os anos de 1993 e 2017, realizada por [Benabdellah; Benghabrit; Bouhaddou, 2019]. É possível analisar que, dentre os trabalhos selecionados, a maior porcentagem se refere a novas abordagens e aplicações dos métodos da literatura para resolução de problemas. Isso demonstra, mesmo após anos de estudo, que os métodos de agrupamento ainda são relevantes, que novos métodos conseguem ser propostos com os avanços em outras áreas do conhecimento e que a tarefa do agrupamento é altamente adaptável, sendo possível sua utilização em diversas áreas de

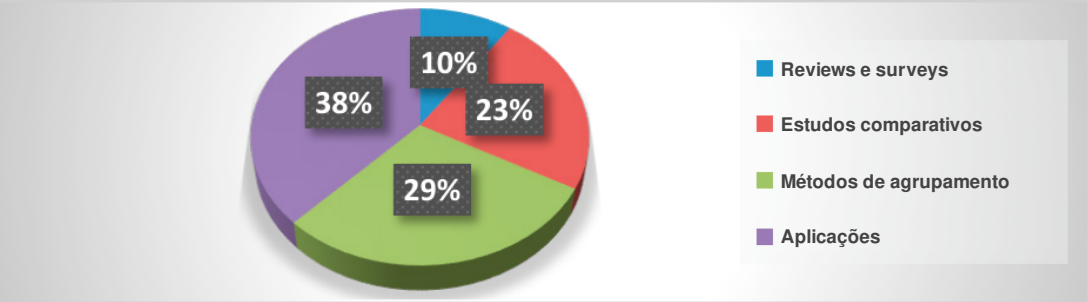


Figura 1 – Resultado da análise de artigos publicados na área de agrupamento entre os anos de 1994 e 2017. Traduzido de: [Benabdellah; Benghabrit; Bouhaddou, 2019]

atuação do conhecimento humano.

Em termos gerais, um processo de agrupamento pode ser definido em quatro etapas principais: *(i)* Seleção ou extração de características; *(ii)* Criação ou escolha do método de agrupamento e métrica de distância/similaridade; *(iii)* Validação dos agrupamentos; e *(iv)* Interpretação de resultados [Rui Xu; Wunsch, 2005; Jain; Murty; Flynn, 1999]. A *Seleção ou extração de características* é a etapa onde os valores de entrada para uma técnica de agrupamento serão definidos. A extração utiliza descritores para obtenção de características, como redes neurais profundas, ou técnicas de transformação para gerar melhores características a partir das originais. A seleção analisa um conjunto de características para selecionar as mais relevantes para o processo. Esta etapa é de grande importância, afetando diretamente a configuração de agrupamentos e as duas técnicas podem ser combinadas para obtenção de melhores resultados.

A etapa de *Criação ou escolha do método de agrupamento e medida de distância ou similaridade* está diretamente conectada a qual abordagem será escolhida para analisar o conjunto de dados recebido pelo método. Cada categoria dos métodos de agrupamento foi desenvolvida em torno de uma visão específica sobre qual a melhor forma e medida de distância/similaridade para análise de conjuntos de dados. Após a separação do conjunto de dados, a etapa de *Validação dos agrupamentos* utiliza-se de métricas objetivas para análise da qualidade dos grupos. Estas métricas podem ser relacionadas a características internas, externas ou realizar a comparação em dois agrupamentos diferentes. Por fim, após a separação e validação dos agrupamentos, a *Interpretação dos resultados* é o principal objetivo do processo. As informações contidas nos agrupamentos podem ser utilizadas por especialistas para tomada de decisões ou resolução de problemas.

É importante notar que o processo de agrupamento não é, necessariamente, realizado somente uma vez. Assim, o processo pode ser refeito para que uma melhor interpretação do conjunto seja obtida [Rui Xu; Wunsch, 2005]. Este trabalho realiza a análise das etapas *(ii)* e *(iii)* deste processo, as quais são detalhadas nas Seções 2.2 e 2.3, respectivamente.

2.2 Categorias de Métodos de Agrupamento

Conforme discutido na Seção 2.1, um conjunto de dados pode ser analisado através de diferentes perspectivas ou técnicas para obtenção dos agrupamentos. Desta maneira, os métodos de agrupamento podem ser divididos em cinco categorias: *(i)* Hierárquicos; *(ii)* Particionais; *(iii)* Baseados em Grafos; *(iv)* Baseados em Densidade; e *(v)* Baseados em Modelos de Redes Neurais [Benabdellah; Benghabrit; Bouhaddou, 2019; Saxena et al., 2017].

Cada categoria possui características únicas quanto à abordagem utilizada, parâmetros necessários e tipo de análises realizadas. As subseções a seguir detalham cada uma das categorias citadas acima e apresentam algoritmos clássicos para exemplificação das técnicas utilizadas.

2.2.1 Métodos Hierárquicos

Métodos hierárquicos analisam o conjunto de dados e resultam em uma estrutura representada por uma árvore binária, ou dendrograma, na qual a raiz representa todos os elementos em um único grupo e as folhas representam cada um dos objetos como um agrupamento unitário [Rui Xu; Wunsch, 2005; Saxena et al., 2017]. A Figura 2 apresenta um dendrograma resultante de um método hierárquico. É possível notar que este dendrograma pode ser analisado em diferentes níveis, onde cada nível apresenta uma separação final diferente para o conjunto de dados [Jain; Murty; Flynn, 1999].

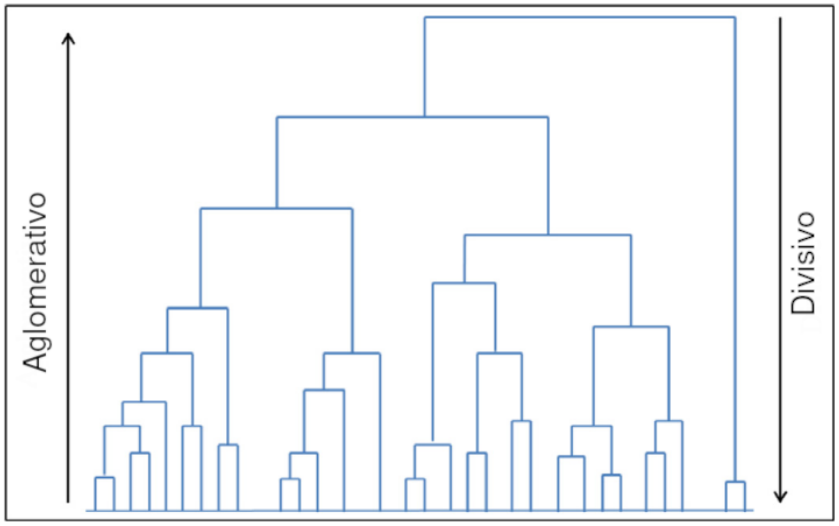


Figura 2 – Dendrograma obtido por um método de agrupamento hierárquico. Traduzido de: [Saxena et al., 2017]

Os métodos hierárquicos podem ser divididos em duas abordagens. Métodos *Divisivos* separam o conjunto de dados em agrupamentos menores, até que cada elemento esteja sozinho em seu agrupamento, realizando a construção do dendrograma de cima

para baixo. Por outro lado, os métodos *Aglomerativos* abordam cada um dos elementos presentes no conjunto como um agrupamento unitário, realizando união par-a-par até que todos os elementos estejam em um mesmo grupo. Esta abordagem realiza a criação do dendrograma de baixo para cima.

Por tratarem todo o conjunto de dados simultaneamente, estes métodos necessitam que as relações de similaridade, ou distância, entre os elementos estejam disponíveis em todas as etapas do processo e que, ao serem realizadas uniões ou separações, estes valores sejam atualizados para decisões futuras. A necessidade do armazenamento desta matriz de distâncias durante todo o processamento do conjunto é uma das principais dificuldades dos métodos hierárquicos [Sarfraz; Sharma; Stiefelhagen, 2019]. Outras dificuldades são a complexidade temporal encontrada durante as comparações par-a-par de agrupamentos e a sensibilidade a ruídos e elementos que não pertençam à distribuição geral do conjunto (conhecidos como *outliers*), sendo que, uma vez colocado em um agrupamento, um elemento não será realocado e poderá afetar as decisões futuras do algoritmo [Saxena et al., 2017].

Outra dificuldade, específica da abordagem divisiva, é o custo computacional para construção do dendrograma. Para um agrupamento com N elementos, existem $2^{N-1}-1$ configurações diferentes para sua separação em dois agrupamentos menores. Essas possibilidades devem ser analisadas para escolha da próxima separação em todos os passos do algoritmo.

De maneira geral, as etapas de um método hierárquico aglomerativo podem ser definidas como [Rui Xu; Wunsch, 2005]:

1. Inicie N agrupamentos unitários e calcule a matriz de similaridade/distância entre eles;
2. Escolha dois agrupamentos através de uma métrica de similaridade/distância e realize a união entre eles;
3. Atualize a matriz com o novo agrupamento formado;
4. Repita os passos *2* e *3* até que um critério de parada seja alcançado.

Os métodos hierárquicos também podem ser divididos em três categorias com base no tipo de ligação utilizado [Saxena et al., 2017]. Seja $\rho(o_i, o_j)$ uma função de distância entre dois objetos o_i e o_j. A ligação *Single-linkage* define a proximidade entre dois agrupamentos como a menor distância entre um objeto o_i pertencente ao agrupamento A e um objeto o_j pertencente ao agrupamento B:

$$\min \ \rho(o_i, o_j) : o_i \in A, o_j \in B \tag{2.1}$$

A ligação *Complete-linkage*, ao contrário da ligação anterior, utiliza a maior distância entre um elemento o_i pertencente a A e um elemento o_j pertencente a B para a definição de proximidade entre os dois agrupamentos:

$$\max \ \rho(o_i, o_j) : o_i \in A, o_j \in B \tag{2.2}$$

Por fim, a ligação *Average-linkage*, assume, como medida de proximidade entre dois agrupamentos, a média de distância de todos os elementos presentes em A para todos os elementos presentes em B:

$$\frac{1}{|A||B|} \sum_{o_i \in A} \sum_{o_j \in B} \rho(o_i, o_j), \tag{2.3}$$

Os métodos hierárquicos são amplamente utilizados, sendo possível listar diversas abordagens, como *Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH)* [Zhang; Ramakrishnan; Livny, 1996], *Clustering Using Representatives (CURE)* [Guha; Rastogi; Shim, 1998] e *Robust Clustering Using Links (ROCK)* [Guha; Rastogi; Shim, 2000].

2.2.2 Métodos Particionais

Métodos particionais, em contraste aos métodos hierárquicos, atribuem os objetos de um conjunto de dados a c agrupamentos de maneira simultânea e sem uma estrutura hierárquica [Jain, 2010; Rui Xu; Wunsch, 2005].

Estes métodos iniciam com a escolha, aleatória ou através de conhecimento prévio, de c centros podendo conter um ou mais elementos cada. A partir desta seleção, todos os elementos são alocados ao agrupamento com centro mais próximo, particionando o conjunto. Ao final de uma execução completa, o agrupamento é avaliado através de uma função, a qual é geralmente uma função de erro quadrática [Jain; Murty; Flynn, 1999], e o processo é repetido até que não haja variação de nenhum elemento.

Em teoria, o agrupamento ideal com base em um critério específico pode ser encontrado pela verificação de todas as combinações possíveis de separação. Porém, mesmo para conjuntos de dados muito pequenos, esta tarefa é computacionalmente inviável [Rui Xu; Wunsch, 2005]. Assim, estes algoritmos trabalham por meio de uma minimização de função e são executados múltiplas vezes. A melhor configuração encontrada após estas execuções é retornada como separação do conjunto de dados.

K-means [MacQueen, 1967] é o método particional mais utilizado e conhecido. Proposto em *1967*, ele ainda é relevante mesmo após décadas de sua criação [Jain, 2010]. Isso se deve, em grande parte, por sua simplicidade e facilidade de implementação para resolução dos mais diversos problemas. Os passos para a realização do *K-means* são:

1. Selecione c centros iniciais para separação dos dados;
2. Gere uma nova partição alocando cada um dos elementos no agrupamento com centro mais próximo;
3. Calcule os novos centros;
4. Repita os passos *2* e *3* até que todos os elementos se estabilizem em seus agrupamentos atuais;

Seja $\mathcal{C} = \{o_1, o_2, ..., o_n\}$ uma coleção de n elementos, representados por vetores d-dimensionais, que devem ser agrupados em um conjunto de c agrupamentos, $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_k\}$. O *K-means* irá procurar uma partição onde o erro quadrático entre os centros de cada agrupamento, obtidos pela média de suas características vetoriais, e seus respectivos elementos seja mínima [Jain, 2010]. Seja μ_i o centro do agrupamento i. A função $\mathcal{E}(\mathcal{S}_i)$ representa o erro quadrático entre μ_i e os elementos contidos em $\mathcal{S}_i$:

$$\mathcal{E}(\mathcal{S}i) = \sum_{o_j \in \mathcal{S}_i} ||o_j - \mu_i||^2. \tag{2.4}$$

O objetivo principal do método é minimizar a soma dos erros de todos os agrupamentos:

$$\mathcal{E}(S) = \sum_{i=1}^{c} \sum_{o_j \in \mathcal{S}_i} ||o_j - \mu_i||^2. \tag{2.5}$$

A Figura 3 representa as etapas do método em ação para realização de uma partição com 3 grupos. É possível notar que, a cada iteração, o número de itens que estão alocados incorretamente diminui e o centro dos agrupamentos, representados por círculos com a mesma cor do agrupamento, se desloca para o local ideal da partição.

O *K-means* requer dois parâmetros para execução: *(i)* O número c de agrupamentos que deverão ser gerados; *(ii)* A métrica de distância que deve ser utilizada para alocar os elementos nos agrupamentos, sendo geralmente utilizada a distância Euclidiana. O parâmetro c exerce um impacto muito grande na separação realizada pelo algoritmo, representando um ponto negativo do *K-means* e dos métodos particionais, em geral. Uma das abordagens para minimizar esse problema visa estimar o número ideal de agrupamentos para um conjunto de dados. Neste contexto, duas abordagens recentes tentam modelar o problema em um cenário onde os dados são vistos por ângulos diversos para encontrar picos onde poderiam estar os centros dos agrupamentos [Masud et al., 2018] e a utilização de um consenso entre diversos métodos para definição do número ideal de agrupamentos [Ünlü; Xanthopoulos, 2019].

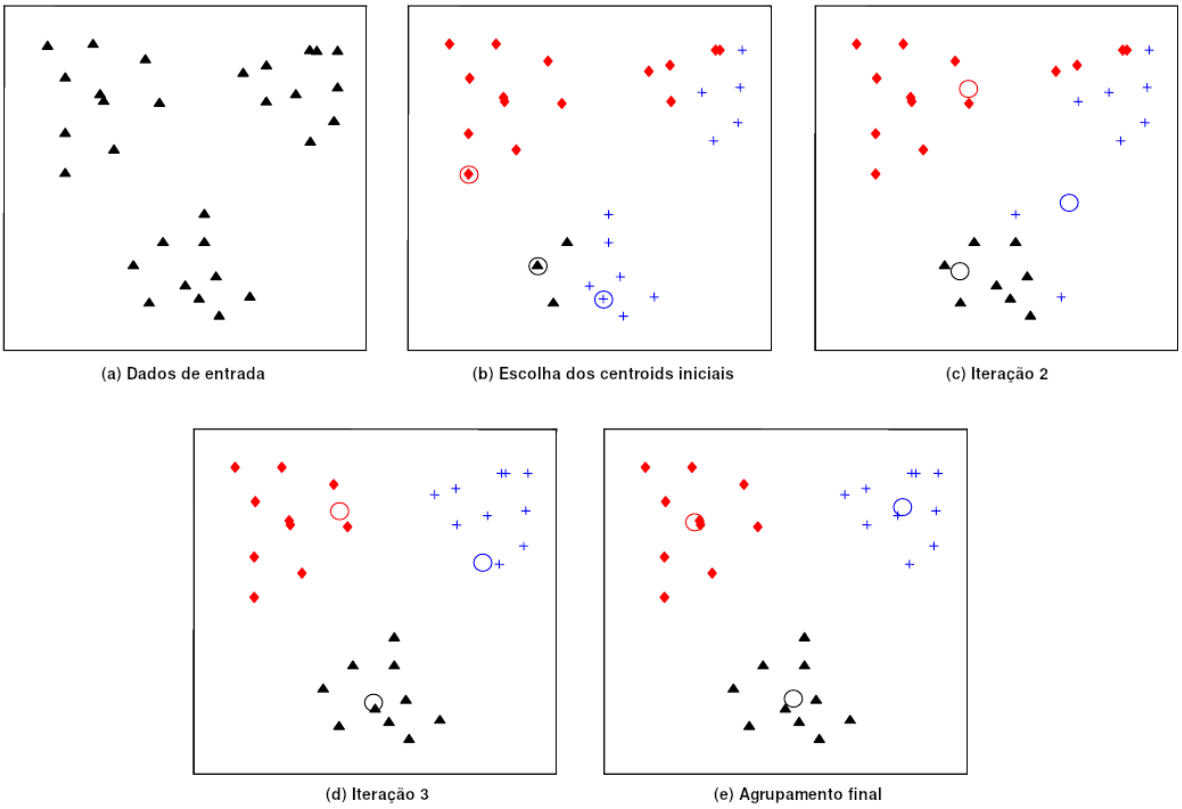


Figura 3 – Etapas de agrupamento do método *K-means*. Traduzido de: [Jain, 2010]

Duas abordagens conhecidas que aprimoram o *K-means* são o *Particioning Around Medoids (PAM)* [Kaufman; Rousseeuw, 1990], que elenca elementos como centros (*medoids*) dos agrupamentos, ao invés de utilizar uma média de todos os pontos, de maneira a minimizar o efeito de *outliers* no resultado, e o *Clustering Large Applications (CLARA)* [Kaufman; Rousseeuw, 1990] que visa trabalhar com conjuntos de dados muito grandes. Por um número definido de iterações, o método obtém uma amostra do conjunto e o separa utilizando o método *PAM*. Com os *c medoids*, o método faz o agrupamento do restante do conjunto. Esse processo é repetido e avaliado pela distância média entre cada elemento e o *medoid* de seu agrupamento.

2.2.3 Métodos Baseados em Grafos

Os métodos baseados em grafos abordam cada elemento do conjunto de dados como um vértice e a medida de distância/similaridade para outros elementos é representada nas arestas criadas com outros vértices deste grafo [Rui Xu; Wunsch, 2005]. Assim, o principal objetivo destes métodos é separar os nós em agrupamentos onde a densidade interna de arestas seja maior que a densidade externa, presente na ligação entre diferentes agrupamentos [Saxena et al., 2017]. A Figura 4 representa uma técnica de agrupamentos baseada em grafos, aplicada a um conjunto de dados. É possível notar que o método alocou os elementos em nós, realizou a ligação por meio das medidas de similaridade entre eles,

sendo encontrados três subgrafos com densidade suficiente para se tornarem agrupamentos.

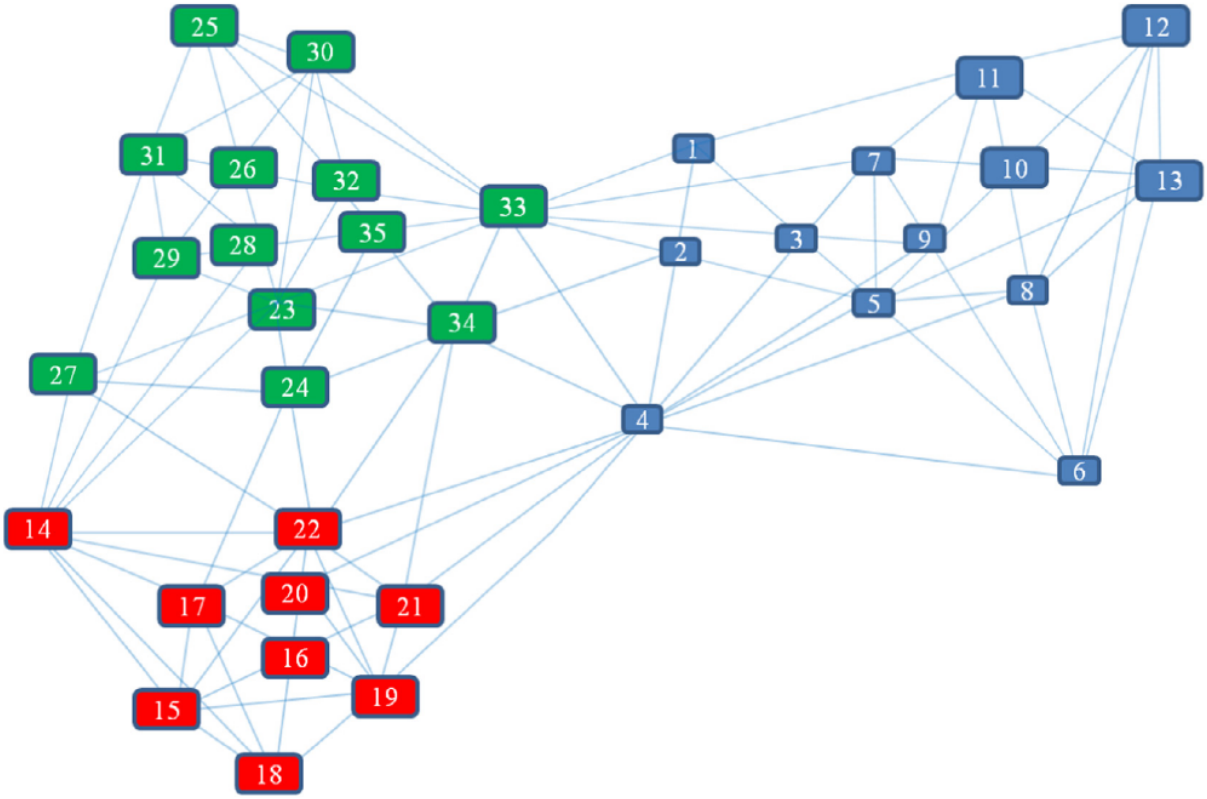


Figura 4 – Resultado de um método de agrupamento baseado em grafos. Fonte: [Saxena et al., 2017]

O *Spectral Clustering* [Donath; Hoffman, 1973; Von Luxburg, 2007] é o método mais conhecido baseado em grafos. Ele realiza o cálculo de uma matriz laplaciana e, com base nos autovetores e autovalores desta matriz, define quantos agrupamentos formar e quais seus centros iniciais. Seja G_s um grafo de similaridade onde cada objeto o_i do conjunto de dados é definido como um vértice, cada aresta entre os objetos o_i e o_j é uma aresta ponderada com valor $s(o_i, o_j)$ e $s(o_i, o_j)$ é a medida de similaridade entre o_i e o_j. A matriz de adjacência $\mathbf{A}$ pode ser definida como:

$$\mathbf{A}_{ij} = \begin{cases} 0, & \textit{se } o_i \textit{ e } o_j \textit{ não estão conectados} \\ s(o_i, o_j), & \textit{caso contrário} \end{cases} \tag{2.6}$$

Diversas técnicas podem ser utilizadas para criação deste grafo de similaridade, incluindo os grafos *threshold*, onde arestas são criadas somente se a similaridade entre os elementos está acima de um determinado limite (nomeado como *threshold*), grafos de vizinhos próximos e grafos completamente conectados. A partir de $\mathbf{A}$, a matriz de grau $\mathbf{D}$ pode ser definida como uma matriz diagonal onde cada diagonal $i = d_i$ pode ser definida como:

$$d_i = \sum_{j=1}^{n} s(o_i, o_j). \tag{2.7}$$

Utilizando as duas matrizes definidas anteriormente, a matriz laplaciana $\mathbf{L}$, em sua forma não-normalizada, pode ser obtida com a seguinte equação:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \tag{2.8}$$

A matriz laplaciana contém informações importantes sobre a estrutura do grafo de similaridade. A partir do cálculo de seus autovetores e autovalores é possível encontrar o número de componentes conexas e o respectivo agrupamentos dos elementos em torno das mesmas [Von Luxburg, 2007]. O número de autovalores de $\mathbf{L}$ iguais a 0 representa o número de componentes conexas presente no grafo de similaridade e os autovetores associados a eles servem como um indicador de quais elementos estão presentes em cada uma destas componentes. Por fim, o *Spectral* realiza o agrupamento de um conjunto de dados, recebendo como entrada a matriz de similaridade $\mathbf{M} \in \mathbb{R}^{nxn}$ e o número de agrupamentos c, a partir das seguintes etapas:

1. Construa o grafo de similaridade e obtenha a matriz de adjacência $\mathbf{A}$;
2. Compute a matriz laplaciana $\mathbf{L}$;
3. Compute os c primeiros autovetores $\mathbf{u}_1$, ... $\mathbf{u}_c$ de $\mathbf{L}$;
4. Seja $\mathbf{U} \in \mathbb{R}^{nxc}$ a matriz contendo os autovetores encontrados, distribuídos como colunas;
5. Para $i = 1, ..., n$, $\mathbf{x}_i \in \mathbb{R}^c$ representa a linha i da matriz $\mathbf{U}$ e o objeto o_i do conjunto de dados;
6. Agrupe os vetores $\mathbf{x}_i$ pelo método *K-means* nos agrupamentos $\mathcal{S}_1$, ... $\mathcal{S}_k$

A mudança da representação dos elementos presentes no conjunto de dados para os vetores $\mathbf{x}$ presentes na matriz $\mathbf{U}$, insere o contexto aprendido por meio do grafo de similaridade e permite que o *K-means* agrupe os elementos com maior precisão. Outros exemplos de métodos baseados em grafos são o *Minimal Spanning tree (MST)* [Zahn, 1971] e o Método de agrupamento baseado em conjuntos de vizinhança limitados [Urquhart, 1982].

Apesar de as aplicações de grafos para o problema de agrupamento serem muito versáteis, tendo sido amplamente estudadas e com novos métodos sendo explorados constantemente, o relacionamento representado é limitado à ligação entre pares de nós. Desta forma, enquanto a maioria das abordagens para métodos de agrupamento explora os relacionamentos de pares, muitas entidades reais apresentam relacionamentos mais complexos, onde a análise par-a-par pode levar a perda de informação presente no conjunto

de dados original. Nestes cenários, hipergrafos oferecem uma representação natural para estes relacionamentos mais complexos [Kumar et al., 2018].

Um hipergrafo é uma generalização da teoria dos grafos convencionais onde uma aresta pode conectar múltiplos vértices simultaneamente. Diferentemente da relação de vértices e arestas, os hipergrafos podem ser descritos como uma coleção de subconjuntos sobrepostos do conjunto de vértices. Estes subconjuntos são representados pelas hiperarestas [Kumar et al., 2018].

Assim, um hipergrafo pode ser definido como $\mathcal{G}_h = (V, E)$, onde V é o conjunto de vértices do hipergrafo e E é o conjunto de hiperarestas. Conforme mencionado acima, a hiperaresta $e_i \in E$ é um subconjunto dos vértices V, i.e., $e_i \subseteq V$. O hipergrafo pode ser representado por uma matriz de incidência $\mathbf{H}$ de tamanho $|V| \times |E|$, onde as entradas $h(v_i, e_i) = 1$ se o vértice v_i está inserido na hiperaresta e_i e $h(v_i, e_i) = 0$ em caso contrário [Purkait et al., 2017].

As primeiras abordagens da utilização de hipergrafos para tarefas de agrupamento buscaram adaptar o *Spectral* para a separação de hipergrafos, a partir da redução do hipergrafo para um grafo [Schölkopf; Platt; Hofmann, 2007]. Alguns trabalhos recentes para área de agrupamento com hipergrafos incluem um estudo para melhores métodos de obtenção de hiperarestas [Purkait et al., 2017] e um método de agrupamento que adapta o método de *Modularity Maximization* para operação a partir de hipergrafos [Kumar et al., 2018].

2.2.4 Métodos Baseados em Densidade

Na visão probabilística, valores podem ser representados de acordo com diversas distribuições de probabilidade. Desta maneira, métodos de agrupamentos baseados em misturas de densidade assumem que objetos podem ser derivados de diferentes distribuições probabilísticas ou de uma mesma distribuição com parâmetros diferentes e seu objetivo é identificar os agrupamentos e suas distribuições [Rui Xu; Wunsch, 2005; Saxena et al., 2017]. Esta abordagem é desenvolvida como uma aproximação paramétrica onde uma densidade desconhecida $p(x)$ do conjunto é definida como uma mistura de c densidades $p_i(x)$, cada uma pertencendo a um dos c agrupamentos contidos no conjunto de dados. Os parâmetros desta aproximação são estimados com base em um conjunto de dados recebido como amostra [Campello et al., 2019]. Um dos métodos mais conhecidos baseados em densidade probabilística é o *Expectation Maximization Clustering (EM-Clustering)* [Dempster; Laird; Rubin, 1977] que estima os parâmetros para c distribuições gaussianas com base conjunto fornecido.

Em uma abordagem alternativa, os métodos baseados em densidade são uma aproximação não-paramétrica onde agrupamentos são considerados regiões de vizinhança

com alta densidade de elementos, separados por regiões com baixa densidade. Estes métodos não requerem o número desejado de agrupamentos e não analisam a variação de elementos dentro de cada agrupamento [Campello et al., 2019]. Desta maneira, os agrupamentos obtidos por estes métodos não são baseados em valores de similaridade/distância par-a-par, por isso, não tendem a possuir formatos convexos, podendo adquirir formas arbitrárias.

O *Density-Based Spatial Clustering of Applications With Noise (DBSCAN)* [Ester et al., 1996] é um dos métodos baseados em densidade mais conhecido e aplicado em casos reais da literatura. Este método propõe seis definições e as utiliza para definir agrupamentos com base na densidade e nas ligações de vizinhança entre os elementos do conjunto. Primeiramente com base em um parâmetro ϵ, a vizinhança-ϵ ($\mathcal{N}_\epsilon$) é um conjunto de elementos definido como

$$\mathcal{N}_\epsilon(o_i) = o_j \in \mathcal{C} \mid \rho(o_i, o_j) < \ \epsilon, \tag{2.9}$$

onde $\mathcal{C}$ é o conjunto de dados e $\rho(o_i, o_j)$ é a medida de distância entre os pontos o_i e o_j. A partir desta vizinhança e com base em um segundo parâmetro $minPts$, o método diferencia os elementos do conjunto como *pontos principais* e *pontos de borda*. Desta maneira, *pontos principais* são elementos onde $|\mathcal{N}_\epsilon| \geq minPts$. Em uma segunda definição, um ponto o_j é *diretamente alcançável-via-densidade* do ponto o_i se:

1. $o_j \in \mathcal{N}_\epsilon(o_i)$;
2. $\mathcal{N}_\epsilon(o_i) \geq minPts$.

A partir desta relação simétrica entre pares de elementos, o conceito é estendido. Um ponto o_j é considerado *alcançável-via-densidade* de um ponto o_i se existe uma cadeia de pontos $o_1, ..., o_n$, onde $o_1 = o_i$, $o_n = o_j$ e todo elemento o_{k+1} nesta cadeia é *diretamente alcançável-via-densidade* do respectivo ponto o_k. Esta nova relação é assimétrica, porém, mesmo que dois pontos inseridos em um agrupamento A não sejam *alcançáveis-via-densidade* reciprocamente, deve haver um *ponto-principal* em A de onde os dois pontos são *alcançáveis-via-densidade*. Desta maneira, um ponto o_i está *conectado-via-densidade* a um ponto o_j se existe um ponto o_k, tal que ambos os pontos o_i e o_j sejam *alcançáveis-via-densidade* a partir o_k. A Figura 5 ilustra as relações descritas acima. Neste exemplo $minPts = 4$, ϵ é representado pelos círculos, A é um *ponto principal*, N é um ponto de ruído e os pontos B e C são *pontos de borda*.

Por fim, um agrupamento $\mathcal{S}_k \subset \mathcal{C}$ é um subconjunto não-vazio que satisfaz as seguintes condições

1. $\forall o_i, o_j$: *se* $o_i \in \mathcal{S}_k$ e o_j é *alcançável-via-densidade* a partir o_i, então $o_j \in \mathcal{S}_k$;

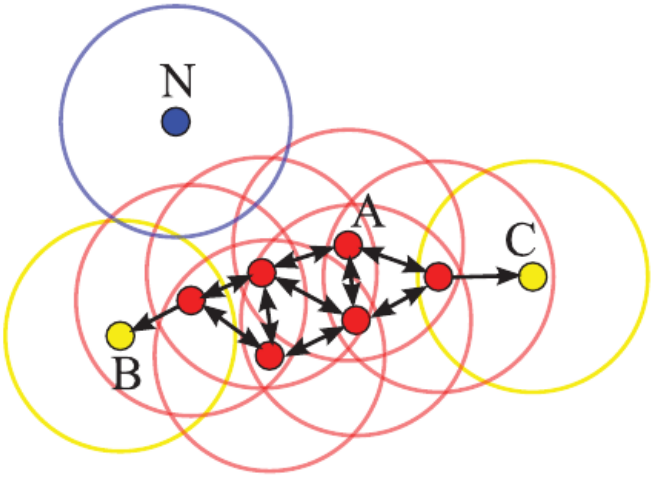


Figura 5 – Ilustração de relações analisadas pelo método *DBSCAN*. Fonte: [Schubert et al., 2017]

2. $\forall o_i, o_j \in \mathcal{S}_k$: o_i é *conectado-via-densidade* à o_j,

Após a obtenção dos agrupamentos, todos os pontos que não pertencerem à nenhum agrupamento serão considerados pontos de ruído. Desta maneira, o *DBSCAN* consegue recuperar um conjunto de agrupamentos $\mathcal{S}_1, ..., \mathcal{S}_c$ com base na análise de densidade do conjunto e sem a necessidade da definição de c. Um ponto negativo destes métodos é a definição dos parâmetros ϵ e $minPts$, a qual não é trivial e requer um conhecimento ou análise prévia do conjunto de dados.

Outros exemplos de métodos de densidade são *Density-based Clustering (DENCUE)* [Hinneburg; Keim, 1998] e, duas variações do método *DBSCAN*, *GDBSCAN* [Sander et al., 1998] e *HDBSCAN* [McInnes; Healy; Astels, 2017].

2.2.5 Métodos Baseados em Redes Neurais

Métodos baseados em modelos de redes neurais utilizam *Redes Neurais Artificiais (RNA)* para agrupar um conjunto de dados recebido como entrada. Durante anos, esta categoria foi dominada por métodos baseados em *Redes Neurais Competitivas (RNC)* [Rui Xu; Wunsch, 2005]. Uma *RNC* é um modelo de *RNA* composta por uma grade de neurônios que competem e cooperam entre si. Dessa maneira, a amostra de entrada é analisada e o somente o neurônio mais similar a ela é ativado. Após sua ativação, o neurônio vencedor e seus vizinhos próximos tem seus pesos atualizados, apresentando o fator de cooperação da rede [Vesanto; Alhoniemi, 2000].

Um dos métodos mais conhecidos e estudados desta categoria é o *Self Organizing Maps (SOM)* [Vesanto; Alhoniemi, 2000; Kohonen, 1998]. O *SOM* tem como objetivo representar um conjunto de dados multi-dimensional em uma grade, geralmente de duas

dimensões. Cada unidade desta grade representa um neurônio que é conectado aos seus adjacentes, o que aplica um conceito de topologia ao modelo. As amostras de entrada, compostas de vetores numéricos que representam as características dos elementos presentes no conjunto de dados, são ligadas a todos os neurônios por meio de pesos adaptáveis [Jain; Murty; Flynn, 1999]. As etapas para implementação e treinamento de um modelo de *SOM* são listadas a seguir [Rui Xu; Wunsch, 2005]:

1. Defina a topologia da rede (Quantos neurônios serão utilizados e qual sua disposição);
2. Inicialize os vetores protótipos $\mathbf{m}_i$ para $i = 1, ..., k$ aleatoriamente. Onde k é o número de neurônios;
3. Insira uma amostra $\mathbf{x}$ na rede; escolha o neurônio J mais próximo à $\mathbf{x}$ como vencedor, i.e., $J = \arg\min ||\mathbf{x} - \mathbf{m}_j||$;
4. Atualize os vetores protótipos de J e de seus adjacentes;
5. Repita os passos 3 e 4 com base em uma função de erro ou até que nenhum neurônio sofra mudanças significativas.
6. Após a finalização do processo, realize a separação do conjunto com base nos valores contidos nos neurônios para cada elemento.

Alguns pontos fracos deste algoritmo são à necessidade da definição das dimensões da grade, a possibilidade de um aprendizado errôneo da topologia do conjunto ou de treinamento excessivo, onde somente os elementos utilizados durante o aprendizado são classificados eficientemente nos conjuntos encontrados [Vesanto; Alhoniemi, 2000].

Nos últimos anos, estudos tem aplicado *Redes Neurais Profundas (RNP)* para realização de tarefas de agrupamento. Diversas arquiteturas de *RNP* foram adaptadas para realizar a separação de conjuntos, como *redes neurais residuais (ResNets)* e *redes neurais codificadoras/decodificadoras (Auto-encoders)*. Recentemente, redes neurais baseadas em grafos tem sido aplicadas para tarefas de agrupamento. Alguns exemplos de abordagens com a utilização destes modelos são discutidas na Seção 2.6. Mais informações sobre métodos baseados em *RNPs* podem ser encontradas em [Min et al., 2018].

2.3 Métricas de Avaliação

As métricas disponíveis para avaliação de métodos de agrupamento podem ser classificadas em duas categorias: *(i) métricas internas*, que analisam a relação diretamente entre os elementos agrupados para definição de uma pontuação para a configuração dos agrupamentos obtidos, *(ii) métricas externas*, que comparam os agrupamentos obtidos

em relação a outra configuração, a qual pode ter sido obtida por um método diferente, ou aos rótulos reais do conjunto [Saxena et al., 2017; Rui Xu; Wunsch, 2005]. Por serem utilizadas com maior frequência na literatura, esse trabalho apresenta métricas usadas para avaliação externa de métodos de agrupamento.

O *Mutual Information (MI)* [Strehl; Ghosh, 2002b; Strehl; Ghosh, 2002a] é uma métrica que visa identificar informações estatísticas compartilhadas entre dois agrupamentos, onde um deles é geralmente substituído pelas classes reais esperadas de um conjunto de dados. Sejam $\mathcal{S}$ e $\mathcal{S}'$ duas configurações de agrupamentos obtidas de uma coleção de elmentos $\mathcal{C}$, tal que $|\mathcal{C}| = n$. A entropia é a quantidade de incerteza relacionada a uma configuração de agrupamentos e pode ser definida para $\mathcal{S}$ pela equação:

$$H(\mathcal{S}) = -\sum_{i=1}^{n} P(i)\ log(P(i)), \tag{2.10}$$

onde $P(i) = |\mathcal{S}_i|/n$ é a probabilidade de que um objeto selecionado ao acaso entre no agrupamento $\mathcal{S}_i \in \mathcal{S}$. Da mesma forma, para $\mathcal{S}'$:

$$H(\mathcal{S}') = -\sum_{j=1}^{n} P'(j)\ log(P'(j)), \tag{2.11}$$

onde $P'(j) = |\mathcal{S}'_j|/n$ é a probabilidade de que um objeto selecionado ao acaso seja inserido no agrupamento $\mathcal{S}'_j \in \mathcal{S}'$. Desta forma, a informação mutual (MI em inglês) entre $\mathcal{S}$ e $\mathcal{S}'$ pode ser obtida por:

$$MI(\mathcal{S}, \mathcal{S}') = \sum_{i=1}^{n}\sum_{j=1}^{n} P(i,j)\ log(\frac{P(i,j)}{P(i)P'(j)}), \tag{2.12}$$

onde $P(i,j) = |\mathcal{S}_i \cap \mathcal{S}'_j|/n$ é a probabilidade de que um elemento selecionado ao acaso entre nas classes $\mathcal{S}_i$ e $\mathcal{S}'_j$, simultaneamente. Uma versão normalizada da métrica, denominada *Normalized Mutual Information (NMI)*, melhora o resultado para testes com número elevado de classes, podendo ser obtida pela equação:

$$NMI(\mathcal{S}, \mathcal{S}') = \frac{MI(\mathcal{S}, \mathcal{S}')}{Média(H(\mathcal{S}), H(\mathcal{S}'))}. \tag{2.13}$$

Por outro lado, o *Rand Index (RI)* [Hubert; Arabie, 1985] analisa a similaridade de dois agrupamentos por meio da manipulação de pares dos objetos classificados. Seja $\mathcal{S}$ a configuração de agrupamentos que representa as classes verdadeiras do conjunto, $\mathcal{S}'$ uma configuração de agrupamentos obtida através de um método de agrupamento, a o número de pares que está presente em uma mesma classe nos dois agrupamentos $\mathcal{S}$ e $\mathcal{S}'$ e b o número de pares que está presente em classes diferentes nos agrupamentos $\mathcal{S}$ e $\mathcal{S}'$, podemos definir o RI como:

$$RI(\mathcal{S}, \mathcal{S}') = \frac{a+b}{\mathcal{C}_2^n}, \tag{2.14}$$

onde $\mathcal{C}_2^n$ representa todos os possíveis pares no conjunto de dados $\mathcal{C}$ analisado, onde $|\mathcal{C}| = n$. A métrica *RI*, porém, não garante que uma formação de agrupamento gerada aleatoriamente receberá um valor próximo de 0. Desta maneira, o *Adjusted Rand Index (ARI)* é obtido por meio da subtração dos pares esperados pela métrica, representado por $E[RI]$:

$$ARI(\mathcal{S}, \mathcal{S}') = \frac{RI - E[RI]}{\arg\max RI - E[RI]}, \tag{2.15}$$

retornando um valor entre 0 e 1 que representa o grau de similaridade entre os agrupamentos.

As métricas de *Homegeneity, Completeness e V-Measure* [Rosenberg; Hirschberg, 2007] são três métricas complementares que analisam uma configuração de agrupamentos como um todo. *Homogeneity* representa a homogeneidade do agrupamento e o avalia de maneira que, na separação ideal, cada agrupamento contenha somente elementos de uma única classe. Por outro lado, *Completeness* visa avaliar a completude do agrupamento, de modo que todos os elementos de uma classe sejam colocados em um mesmo agrupamento. Por fim, o *V-measure* calcula uma média harmônica entre as duas métricas citadas acima. Seja $\mathcal{S}$ a configuração de agrupamentos que representa as classes verdadeiras do conjunto $\mathcal{C}$, com $|\mathcal{C}| = n$, $\mathcal{S}'$ uma configuração de agrupamentos deste conjunto obtida através de um método de agrupamento. Definimos A como uma tabela de contingência, tal que $A = a_{ij}$ onde a_{ij} é o número de membros da classe $\mathcal{S}_i$ e do agrupamento $\mathcal{S}'_j$, simultaneamente. Com base em A, $H(\mathcal{S}|\mathcal{S}')$ representa a entropia da distribuição de classe dado o agrupamento proposto e analisa o quão similar é o agrupamento $\mathcal{S}'$ do conjunto representante das classes $\mathcal{S}$ e pode ser definido por:

$$H(\mathcal{S}|\mathcal{S}') = -\sum_{j=1}^{|\mathcal{S}'|}\sum_{i=1}^{|\mathcal{S}|}\frac{a_{ij}}{n}\log\left(\frac{a_{ij}}{\sum_{k=1}^{|\mathcal{S}'|} a_{ik}}\right), \tag{2.16}$$

Em contraste, $H(\mathcal{S})$ representa a máxima redução entrópica provida pela distribuição, i.e., todos os agrupamentos representam as classes reais:

$$H(\mathcal{S}) = -\sum_{i=1}^{|\mathcal{S}|}\frac{\sum_{j=1}^{|\mathcal{S}'|} a_{ij}}{n}\log\frac{\sum_{j=1}^{|\mathcal{S}'|} a_{ij}}{n}. \tag{2.17}$$

Com base as duas medidas descritas acima, a *Homogeneity* é definida como:

$$hh = \begin{cases} 1 & se\ H(\mathcal{S}|\mathcal{S}') = 0, \\ 1 - \dfrac{H(\mathcal{S}|\mathcal{S}')}{H(\mathcal{S})} & caso\ contrário, \end{cases} \tag{2.18}$$

retornando um valor entre 0 e 1. Também com base nas mesmas medidas, a *Completeness* pode ser obtida por:

$$hc = \begin{cases} 1 & se\ H(\mathcal{S}|\mathcal{S}') = 0, \\ 1 - \dfrac{H(\mathcal{S}'|\mathcal{S})}{H(\mathcal{S}')} & caso\ contrário, \end{cases} \tag{2.19}$$

também retornando um valor entre 0 e 1. Por fim, a *V-measure* é obtida pelo cálculo da média harmônica entre as duas outras métricas:

$$V = 2 * \frac{hh * hc}{hh + hc}. \tag{2.20}$$

Visando avaliar a assertividade dos agrupamentos obtidos, *F-Measure* [Chinchor, 1992] é uma métrica clássica, aplicada a métodos de aprendizado supervisionado e baseada nas métricas de *Precision* e *Recall*. Ela pode ser aplicada para métodos de agrupamento por meio da utilização de pares de elementos, conforme descrito a seguir.

Seja $\mathcal{S}' = \{\mathcal{S}_1, \ldots, \mathcal{S}_c\}$ uma configuração de agrupamentos obtida através de um método M. Seja $\mathcal{S}$ um conjunto de agrupamentos, onde cada agrupamento representa uma das classes verdadeiras do conjunto de dados analisado. Seja $Pares(\mathcal{S})$ uma função que obtém todas as possíveis combinações de elementos par-a-par contidas em cada um dos grupos da configuração de agrupamentos $\mathcal{S}$. Dado um método de agrupamento M e um conjunto de classes reais, as medidas mencionadas acima são calculadas da seguinte maneira:

$$TP(TruePositives) = |Pares(\mathcal{S}') \cap Pares(\mathcal{S})| \tag{2.21}$$

$$FP(FalsePositives) = |Pares(\mathcal{S}') - Pares(\mathcal{S})| \tag{2.22}$$

$$FN(FalseNegatives) = |Pares(\mathcal{S}) - Pares(\mathcal{S}')| \tag{2.23}$$

$$Precision = TP/(TP + FP) \tag{2.24}$$

$$Recall = TP/(TP + FN) \tag{2.25}$$

$$F - Measure(F1 - Score) = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \tag{2.26}$$

Recentemente, métodos de agrupamento baseados em redes neurais tem utilizado a métrica de *Acurácia (ACC)* para avaliar os agrupamentos realizados [Min et al., 2018]. A

métrica *ACC*, assim como o *F-Measure*, é uma métrica de avaliação muito utilizada para tarefas de aprendizado supervisionado e analisa o número de acertos entre as classes reais e a classificação obtida pelos algoritmos de aprendizado de máquina.

No cenário dos métodos de agrupamentos, a acurácia pode ser obtida pela permutação dos rótulos da configuração de agrupamento, buscando atribuir cada grupo a uma classe específica de maneira a maximizar a pontuação, podendo ser definida pela seguinte equação:

$$ACC = \max_{m} \frac{\sum_{i=1}^{n} y_i = m(c_i)}{n}, \tag{2.27}$$

onde y_i representa a classe real do elemento o_i, c_i representa o grupo ao qual o elemento o_i foi alocado e $m(c_i)$ é uma função que mapeia todas as permutações um-a-um possíveis entre os grupos e as classes reais. A função $m(c_i)$ pode ser computada de maneira eficiente pela utilização do *Algoritmo Húngaro* [Kuhn, 1955].

2.4 Modelo de Ranqueamento

Seja $\mathcal{C} = \{o_1, o_2, \ldots, o_n\}$ uma coleção de objetos, em que n representa o número de objetos presentes na coleção. $\mathcal{D}$ representa um descritor de objetos, o qual pode ser definido como uma tupla $\mathcal{D} = (\epsilon, \rho)$ [Torres; Falcão, 2006], onde ϵ: $\hat{I} \rightarrow \mathbb{R}^d$ é uma função que extrai um vetor de características $v_{\hat{I}}$ de um objeto $\hat{I}$; e ρ: $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ é uma função que calcula a distância entre dois objetos conforme os vetores de características correspondentes. No decorrer deste texto, $\rho(i, j)$ é utilizado para descrever a distância entre dois objetos o_i e o_j.

Uma lista τ_q pode ser formalmente definida como uma permutação $(o_1, o_2, \ldots, o_L)$ do subconjunto $\mathcal{C}_L \subset \mathcal{C}$ obtido pela utilização de ρ como a função de distância. Esta lista contém os L objetos mais similares a um objeto o_q, de maneira que $|\mathcal{C}_L| = L$ e $L \ll n$. Para um objeto o_q, $\tau_q(i)$ é interpretado como a posição (ou classificação) do objeto o_i na lista ranqueada τ_q. Se o_i está posicionado antes de o_j na lista ranqueada do objeto o_q $(\tau_q(i) < \tau_q(j))$, então $\rho(q, i) \leq \rho(q, j)$.

Nos métodos apresentados neste trabalho, conjuntos $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_n\}$ são obtidas por meio da geração de listas ranqueadas para cada objeto presente em uma coleção $\mathcal{C}$ analisada, utilizando a distância Euclidiana como a função ρ. Estes conjuntos contêm informações de similaridade importantes sobre a coleção $\mathcal{C}$, as quais podem ser exploradas para criação de agrupamentos explorando métodos de *manifold learning*, discutidos na próxima seção.

2.5 Manifold Learning baseado em Ranqueamento

O processamento de conteúdo multimídia, como fotos e vídeos, em tarefas de recuperação de informações e aprendizado de máquina é frequentemente realizado por meio de descritores [Torres; Falcão, 2006]. Os descritores extraem características para obter uma representação numérica, frequentemente em formato vetorial. Também métodos de aprendizado profundo comumente utilizam estratégias de transferência de aprendizado para obtenção de estruturas análogas.

Tais representações vetoriais são utilizadas especialmente em tarefas de recuperação, para o cálculo de medidas de distância (ou similaridade) par-a-par. Porém, ao analisar somente pares de objetos, as informações contidas nas relações mais abrangentes entre os elementos é ignorada, o que pode levar a uma *lacuna semântica* [**??**] entre a representação vetorial obtida pelos descritores e as informações de similaridade entre os elementos codificadas em relações dos conjuntos de dados.

Neste cenário, métodos de *Manifold Learning* (*Aprendizado de Estrutura* em tradução livre) são algoritmos que analisam o conjunto de dados para encontrar novas medidas de similaridade de abordagem global, ou seja, que sejam capazes de codificar o relacionamento entre os elementos analisados [Yang; Prasad; Latecki, 2013]. Ao codificar as relações globais de similaridade dos conjuntos de dados, os métodos calculam medidas mais eficazes, que podem impactar positivamente em tarefas de recuperação e aprendizado de máquina.

Apesar de haver uma grande diversidade de estratégias e terminologias de *Manifold Learning*, duas abordagens têm recebido grande atenção na literatura: *(i) baseados em processos de difusão*: os quais utilizam caminhos aleatórios baseados em matrizes de afinidade recebidas como entrada [Donoser; Bischof, 2013; Yang; Koknar-Tezel; Latecki, 2009; Pedronette; Torres, 2017]; *(ii) baseados em ranqueamento*: métodos que exploram as informações em formato de ranqueamentos [Qin et al., 2011; Shen et al., 2012; Pedronette; Torres, 2013]. Embora ambas as estratégias sejam eficazes, os métodos baseados em processos de difusão comumente requerem alto poder computacional, enquanto os métodos baseados em ranqueamento são menos custosos.

A Figura 6 ilustra o efeito da aplicação de uma técnica de *manifold learning* [Pedronette; Gonçalves; Guilherme, 2018] baseada em ranqueamento no conjunto *Two Moons*. Nesta figura, os elementos i e c pertencem a mesma classe, apresentada em azul, e o elemento j pertence a outra classe, apresentada em vermelho. É possível notar que a medida de distância $\rho(i, j)$ seria inferior a medida $\rho(i, c)$ caso a distância Euclidiana fosse escolhida para uma comparação par-a-par, possivelmente ocasionando em um erro de classificação dos respectivos elementos. Ao analisar todas as relações do conjunto por meio da criação de componentes conexas baseadas nos ranqueamentos dos elementos, o método de *manifold learning* consegue obter uma nova medida ρ que considera a estrutura dos

dados, tal que $\rho(i, j) > \rho(i, c)$.

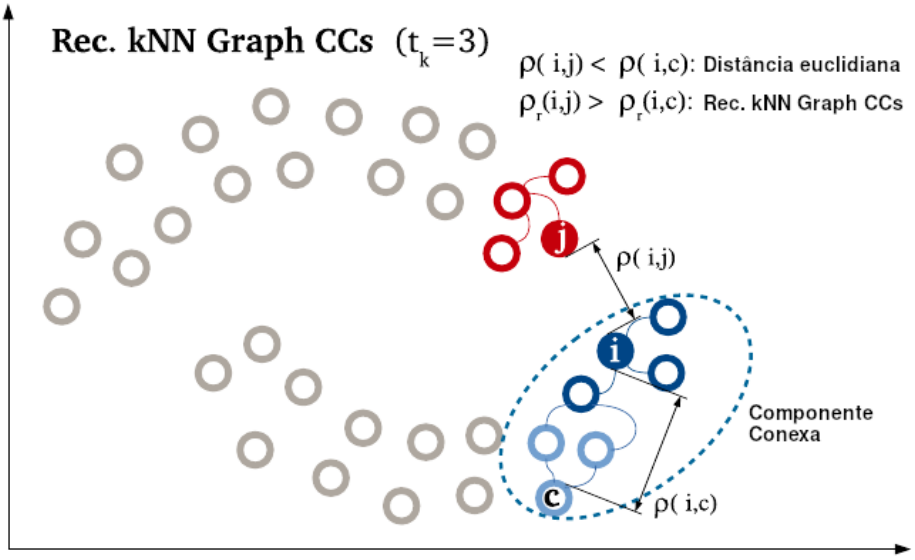


Figura 6 – Representação da análise de distâncias realizada pelo algoritmo *Reciprocal kNN Graph and Connected Components*. Traduzido de: [Pedronette; Gonçalves; Guilherme, 2018]

Este trabalho tem como foco a utilização de métodos de *manifold learning* baseados em ranqueamento, conforme definição apresentada na Seção 2.4. Além de utilizar as relações aprimoradas encontradas pelos métodos escolhidos, informações intermediárias utilizadas para obtenção dessas relações são adaptadas na formação de agrupamentos disjuntos, para derivação de novos métodos de agrupamento.

2.6 Trabalhos Relacionados

Conforme mencionado anteriormente, os métodos de agrupamento continuam em constante evolução, com novas abordagens sendo propostas regularmente na literatura. Dentre os métodos identificados pelo levantamento bibliográfico realizado para este trabalho, algumas abordagens que se relacionam de maneira mais próxima com os métodos propostos foram selecionadas e organizadas em três categorias. A Seção 2.6.1 apresenta métodos que exploram relações de vizinhança. A Seção 2.6.2 descreve métodos que utilizam redes convolucionais baseadas em grafo e a Seção 2.6.3 lista métodos recentes que utilizam abordagens auto-supervisionadas.

2.6.1 Análise de Vizinhança

A análise de vizinhança é uma técnica amplamente utilizada para criação de métodos de agrupamento, inclusive de métodos clássicos da literatura [MacQueen, 1967;

Murtagh, 1983; Schubert et al., 2017]. Analisar as vizinhanças presente entre elementos do conjunto auxilia na descoberta dos agrupamentos naturais do conjunto de dados.

Neste cenário, dois métodos recentes exploram as relações de vizinhança para formulação de agrupamentos. O *Mutual Neighbor-based Clustering Algorithm (MUNEC)* [Ros; Guillaume, 2019] é um método de agrupamento hierárquico aglomerativo que explora as vizinhanças recíprocas de tamanho $k = 1$ para realizar uniões iterativas até que uma condição de parada seja alcançada. Inicialmente, todos os elementos são considerados como agrupamentos unitários sendo unidos com seus vizinhos recíprocos, até que todos os agrupamentos tenham dois ou mais elementos.

A partir desta primeira formação, cada agrupamento passa a ser definido por dois indicadores internos: *(i)* n representa o número de distâncias entre dois vizinhos no agrupamento, onde o número total de elementos é definido por $n + 1$; *(ii)* d representa a média de distância entre dois vizinhos no agrupamento. Ambos os valores são inicializados como $d = 0$ e $n = 0$ para agrupamentos unitários e como $n = 1$ e $d = d(c_i, c_j)$ para agrupamentos com pares. A cada iteração do algoritmo, os agrupamentos que são vizinhos são aglomerados e os valores dos descritores internos são atualizados conforme as seguintes equações:

$$n = n_1 + n_2 + 1, \qquad d = \frac{1}{n}(n_1 d_1 + n_2 d_2 + d_{1,2}) \tag{2.28}$$

Este primeiro estágio de uniões sem restrição é realizado até que todos os agrupamentos contenham dois ou mais elementos. No segundo estágio de uniões, um índice de similaridade s, para dois agrupamentos i e j é definido como:

$$s_{i,j} = \frac{min(d_i, d_{i,j})}{max(d_i, d_{i,j})}. \tag{2.29}$$

Os agrupamentos são aglomerados com base em um limitador $SecureThres = 0.95$, de maneira que a união somente será realizada se $s < SecureThres$. Este segundo estágio de uniões termina quando nenhuma vizinhança recíproca satisfazer o limitador de união.

Em uma última etapa de uniões, três heurísticas são definidas, com base em um grau de homogeneidade entre os agrupamentos, para guiar as aglomerações finais: *(i)* O grau de homogeneidade do par a ser unificado deve ser maior que um limitador $h_t h$. Este limitador é posteriormente decaído a cada iteração, *(ii)* A vizinhança em torno dos agrupamentos unificados é analisada para permitir a aglomeração e *(iii)* a heterogeneidade entre os dois agrupamentos é medida e comparada a um limitador u. Os agrupamentos são aglomerados até que nenhum par satisfaça as heurísticas definidas.

O *First Integer Neighbor Clustering Hierarchy (FINCH)* [Sarfraz; Sharma; Stiefelhagen, 2019] é outro método recente que explora a relação entre os primeiros vizinhos

dos elementos presentes no conjunto de dados para agrupá-los. A principal proposta deste algoritmo é encontrar diferentes configurações de agrupamentos hierárquicos aglomerativos para o conjunto a partir de um método sem parâmetros. Para isso, o algoritmo calcula listas ranqueadas de distância para cada elemento do conjunto de dados e computa uma matriz de adjacência $\mathbf{A}$ definida por:

$$\mathbf{A}_{i,j} = \begin{cases} 1 & se\ j = \tau_i^1\ ou\ \tau_j^1 = i\ ou\ \tau_i^1 = \tau_j^1, \\ 0 & caso\ contrário, \end{cases} \tag{2.30}$$

onde τ_i^1 representa o primeiro vizinho do elemento i. Esta matriz de adjacência liga cada ponto i ao seu primeiro vizinho, $j = \tau_i^1$, reforça a simetria dado que a relação de vizinhança não é simétrica, $\tau_j^1 = i$, e assume que existe uma relação de similaridade entre dois pontos que compartilham o mesmo primeiro vizinho, $\tau_i^1 = \tau_j^1$. Esta matriz de adjacência é esparsa e representa um grafo não direcionado onde os vértices são os elementos e as arestas são definidas conforme a matriz de adjacência A. A partir deste grafo, as componentes conexas são encontradas e unidas em um único agrupamento.

Após a primeira etapa de uniões, o método recupera uma partição dos dados representada como uma configuração de agrupamentos do conjunto. A partir desta partição o processo é iterado, onde cada agrupamento é considerado como um elemento a ser unido. O primeiro vizinho é calculado para cada agrupamento, dando origem a uma nova matriz de adjacência $\mathbf{A}$ que guiará uma nova etapa de uniões. Este processo é iterado até que restem somente 2 agrupamentos no conjunto.

Em uma segunda abordagem, caso um número específico de agrupamentos seja informado, o *FINCH* realiza o processo mencionado acima até encontrar a menor partição onde $c_P > c$, sendo c_P o número de agrupamento na partição P e c o número esperado de agrupamentos. A partir desta partição encontrada e do cálculo de sua matriz de adjacência $\mathbf{A}$, o método une somente a relação de primeiro vizinho mais próxima, i.e., com menor distância entre os vizinhos, a cada iteração até que o número c de agrupamentos seja encontrado.

Ambos os métodos descritos exploram as vizinhanças para direcionar as uniões, aplicando abordagens diferentes para definir o fim das uniões. Porém, nenhum dos algoritmos explora as informações contidas nas relações de vizinhança para aprimorar a similaridade dos elementos. Um dos métodos proposto neste trabalho utiliza as vizinhanças de maneira análoga, com base nas características das estruturas de dados selecionadas, porém usa os métodos de *manifold learning* para descobrir relações mais eficazes antes de conduzir as uniões dos elementos em agrupamentos.

2.6.2 Redes Neurais Baseadas em Grafos

Diversas abordagens recentes tem aplicado *RNP* para criação de métodos de agrupamento, porém, grande parte desses métodos são focados na classificação não-supervisionada de imagens [Darlow; Storkey, 2020; Huang; Zhu, 2020]. Para isso, a representação dos pixels das imagens de entrada, em forma de matrizes, são processados por *Redes Neurais Convolucionais (CNN em inglês)*. Esta vertente de métodos foge do escopo proposto para esse trabalho, o qual visa agrupar elementos com base em vetores de características.

Apesar disso, métodos recentes de agrupamento tem utilizado *Redes Neurais Baseadas em Grafos (GCN em inglês)*, modelos de redes neurais que adaptam a operação de convolução, utilizadas pelas *CNNs*, para aplicação com base em matrizes de adjacência. A definição das *GCNs* é apresentada na Seção 4.3. Dois métodos desta vertente foram selecionados.

A *MinCutPool* [Bianchi; Grattarola; Alippi, 2020] é uma camada de *Pooling*, responsável por reduzir os mapas de características entre camadas de *Passagem de Mensagem (MP em inglês)* que trabalham propagando rótulos durante as operações da *GCN*. O modelo proposto nesse trabalho parte do pressuposto que as representações originais dos vértices do grafo de entrada já contém uma possível separação inicial do conjunto. Explorando essa afirmação, uma camada de *Perceptron Multi-Camadas (MLP em inglês)* é adicionada no fim da arquitetura de rede para classificar cada elemento para uma das c classes esperadas.

Para realizar o treinamento do modelo, a arquitetura proposta é treinada a partir do gradiente descendente com base em uma função de custo combinada $\mathcal{L} = \mathcal{L}_c + \mathcal{L}_o$. Nessa equação $\mathcal{L}_c$ otimiza os parâmetros presentes nas camadas *MinCutPool* por meio da avaliação dos rótulos obtidos da camada *MLP*, de maneira não-supervisionada. O componente $\mathcal{L}_o$ é utilizado para evitar a minimização completa de $\mathcal{L}_c$, incentivando que os rótulos de agrupamento sejam ortogonais e que a composição de cada agrupamento seja similar. O modelo é treinado integralmente com base na função de custo combinada $\mathcal{L}$ por um número definido de épocas. Após o treinamento, todo o conjunto é classificado nos c agrupamentos definidos.

O *Structural Deep Clustering Network (SDCN)* [Bo et al., 2020] é um algoritmo de agrupamentos que combina uma *GCN* com uma rede *Auto-encoder* para agrupar os elementos de um conjunto de entrada. Em um primeiro momento, o *Auto-encoder* é treinado a partir do conjunto de entrada. Os modelos *Auto-encoder* são *RNPs* que processam a informação de entrada, reduzindo sua dimensionalidade para posteriormente aprender a recriar o elemento recebido através de operações que retornam as características para a dimensão original. Ao utilizar o próprio elemento como rótulo para esse treinamento, o *Auto-encoder* aprende a codificar e decodificar o conjunto de dados de maneira não-

supervisionada.

Após o treinamento do *Auto-encoder*, um grafo de vizinhos recíprocos é criado a partir dos vetores de características do conjunto de dados. O grafo e as características dos elementos são utilizados como entrada para uma *GCN* com o mesmo número de camadas do *Auto-encoder* treinado previamente. Cada camada da *GCN* recebe a representação aprendida pela respectiva camada do modelo de *Auto-encoder* antes de aplicar sua função de ativação. Desta maneira, as representações aprendidas auxiliam na classificação dos elementos, realizada pela *GCN*. O modelo completo é treinado por uma função de custo composta que busca aproximar as representações dos objetos com base em seu centro, calculado como a média das representações dos elementos do agrupamento.

Ambos os métodos descritos acimas aplicam as *GCN* para tarefas de agrupamento por meio da formulação de funções de custo que integram a tarefa de separação dos elementos ao processo de treinamento do modelo, buscando ambos os objetivos durante o processo. O *SGCC* utiliza uma abordagem diferente que utiliza a *GCN* como uma estrutura de aprendizado semi-supervisionado, onde os rótulos utilizados durante o treinamento extraídos do conjunto por meio de *manifold learning*.

2.6.3 Métodos Auto-supervisionados

O aprendizado auto-supervisionado é uma linha de pesquisa recente que utiliza de estruturas criadas para treinamento por meio de aprendizado supervisionado ou semi-supervisionado, realizando estes treinamentos com rótulos obtidos por técnicas não-supervisionadas ou pela comparação entre elementos do conjunto, sem utilização de nenhuma informação prévia sobre os mesmos [Xie et al., 2021].

O *Self-Supervised Convolutional Subspace Clustering Network* ($S^2ConvSCN$) [Zhang et al., 2019] é um modelo de *RNP* auto-supervisionado que combina um módulo de *CNN* para extração de características, um módulo de *Auto-expressão (Self-expression em inglês)* em que os elementos são representados como a combinação linear dos outros elementos do conjunto e um módulo de agrupamento *Spectral* que produz uma configuração de agrupamentos para o conjunto.

O sistema de auto-supervisão é incorporado na função de custo para treinamento da rede por meio da aplicação de um módulo de camadas *Completamente Conectadas (FC em inglês)* que classifica as representações aprendidas pelo módulo *CNN* e as compara com os agrupamentos aprendidos pelo módulo *Spectral*. Além disso, um método de camadas *Decodificadoras (Decoders)* é utilizado para guiar o treinamento do módulo de *Self-expression*. Ao final do processo de treinamento, $S^2ConvSCN$ realiza a separação do conjunto de dados em agrupamentos e extrai uma nova representação para os elementos, a partir do módulo de *Self-expression*.

Por fim, o *Fast Self-Supervised Clustering With Anchor Graph (FSSC)* [Wang et al., 2021] é um algoritmo de agrupamento auto-supervisionado que integra um algoritmo de aprendizado semi-supervisionado *Fast Semi-Supervised Framework (FSSF)* com métodos não-supervisionados para separação do conjunto de dados.

Em um primeiro momento, uma variante hierárquica do método *K-Means* é aplicado no conjunto de dados e c elementos representantes são selecionados com base no resultado obtido pelo algoritmo. Cada representante escolhido é associado a um dos c agrupamentos desejados. Em uma segunda etapa, um novo vetor de representação é computado para cada elemento do conjunto, com base nos representantes escolhidos na etapa anterior. Estas novas representações são aplicadas ao *FSSF* que realiza a propagação dos rótulos dos representantes, separando o conjunto nos c agrupamentos encontrados.

As duas abordagens mencionadas acima utilizam diferentes abordagens para implementação do aprendizado não supervisado, enquanto o método $S^2ConvSCN$ [Zhang et al., 2019] direciona o aprendizado por meio da comparação entre uma representação dos dados e uma configuração de agrupamento, ambas obtidas de maneira não-supervisionada, o *FSSC* [Wang et al., 2021] utiliza um método de agrupamento para selecionar representantes e propaga os rótulos por meio de aprendizado não-supervisionado.

Há pontos em comum entre os métodos *FSSC*, $S^2ConvSCN$ e a abordagem auto-supervisionada proposta neste trabalho, o *SGCC*. Ambos os métodos aprendem novas representações para os elementos do conjunto durante o processo de agrupamento. Etapa que também é realizada de maneira opcional durante o *SGCC*. O *FSSC* também se assemelha ao método proposto na abordagem de selecionar elementos representantes para utilização de um método de aprendizado semi-supervisionado, responsável por separar o conjunto. Porém, o *SGCC* realiza as duas tarefas utilizando técnicas mais atuais, em que são utilizados um modelo de hipergrafo e redes neurais baseadas em grafos, respectivamente.

3 Agrupamento baseado em Grafo Recíproco e Componentes Conexas

O método *Clustering through Reciprocal kNN Graph and Connected Components (C-ReckNN)*[1] é apresentado neste capítulo. Primeiramente, a Seção 3.1 apresenta uma visão geral das etapas envolvidas no método proposto. A Seção 3.2 introduz o *Reciprocal kNN Graph and Connected Components* [Pedronette; Gonçalves; Guilherme, 2018], método de *manifold learning* utilizado. Por fim, a Seção 3.3 detalha como o *C-ReckNN* utiliza o método de *manifold learning* descrito na seção anterior para separar um conjunto de dados em agrupamentos.

3.1 Visão Geral

O *C-ReckNN* [Lopes. et al., 2020] é um método de agrupamento hierárquico aglomerativo que utiliza a aplicação uma técnica de *manifold learning* [Pedronette; Gonçalves; Guilherme, 2018], que explora componentes conexas contidas em um grafo de vizinhos recíprocos, para obtenção uma separação mais eficiente de agrupamentos para o conjunto de dados.

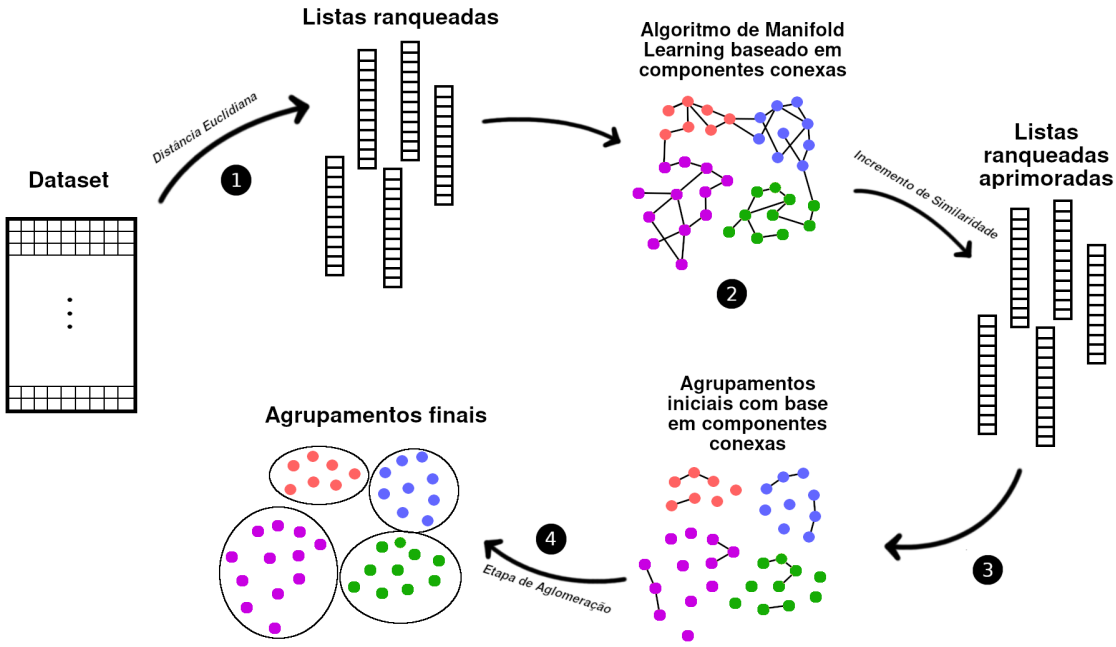


Figura 7 – Fluxo de dados implementado no *C-ReckNN*

[1] Implementação disponível em <https://github.com/lopes-leonardo/crecknn>

A Figura 7 representa o processo completo da metodologia proposta e as quatro principais etapas realizadas. Primeiramente, a Etapa **1** utiliza a distância Euclidiana para obtenção de listas ranqueadas do conjunto de dados, conforme descrito na Seção 2.4. Na Etapa **2**, detalhada na Seção 3.2, o método de *manifold learning* é aplicado nas listas ranqueadas do conjunto. Durante esta etapa, uma nova métrica de distância, capaz de analisar a estrutura natural do conjunto de dados, e novas listas ranqueadas são obtidas.

Estas informações são utilizadas pelo *C-ReckNN* para separar o conjunto em agrupamentos explorando as componentes conexas encontradas pelo método de *manifold learning*. Essa separação é realizada nas Etapas **3** e **4**, detalhadas na Seção 3.3.

3.2 Grafo de Vizinhos Recíprocos e Componentes Conexas

O algoritmo *Reciprocal kNN Graph and Connected Components* [Pedronette; Gonçalves; Guilherme, 2018] usa um conjunto de listas ranqueadas $\mathcal{T}$, descrito na Seção 2.4, para capturar a estrutura do conjunto de dados. Como dito anteriormente, estas listas ranqueadas contém uma informação mais profunda sobre o relacionamento entre os objetos em comparação com medidas tradicionais de distância, como a distância Euclidiana, que só comparam pares.

Para aproveitar esta informação, o algoritmo explora a relação de vizinhos próximos existente nos ranqueamentos deste conjunto de listas. Nesta relação, um elemento o_j é considerado vizinho próximo de outro elemento o_i se estiver presente nas k primeiras posições do ranqueamento do mesmo, onde k representa o tamanho da vizinhança explorada.

Porém, diferentemente das medidas de similaridade ou distância, onde todo $\rho(i,j) = \rho(j,i)$, o relacionamento de vizinhos próximos não é simétrico e, embora seja mais confiável, define um número menor de relações entre os objetos, restringindo a análise de similaridade. Desta maneira, o fato de um elemento o_j estar presente na vizinhança próxima de o_i não garante que o_i esteja na vizinhança próxima de o_j [Qin et al., 2011]. Este número de relações diminui se analisarmos relacionamentos de vizinhança recíproca, onde os dois elementos devem estar entre as k primeiras posições de suas respectivas listas ranqueadas.

Um grafo de vizinhos próximos recíprocos, onde cada aresta representa uma relação de vizinhança recíproca entre dois objetos, é normalmente esparso, provendo pouca informação sobre a estrutura interna dos dados. Neste cenário, as componentes conexas deste grafo podem ser exploradas para expandir a vizinha dos itens que a compõem.

Baseado nestes conceitos, o algoritmo *Reciprocal kNN Graph and Connected Components* modela a informação contida nas listas ranqueadas para gerar, a partir da criação e manipulação de um grafo de vizinhos próximos recíprocos, uma nova medida de distância visando aumentar a separabilidade dos objetos com base na distribuição do conjunto de

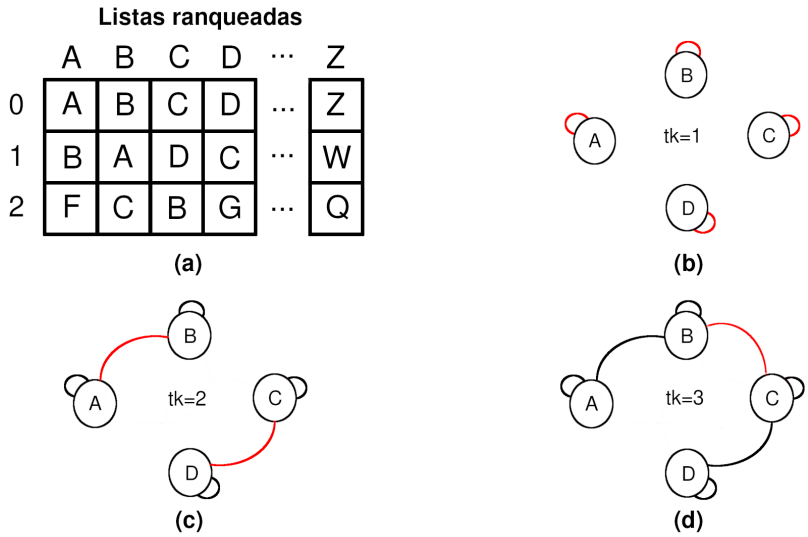


Figura 8 – Representação do processo de criação do grafo de vizinhança recíproca. *(a)* Conjunto de listas ranqueadas utilizada como entrada. *(b)* Primeira iteração. *(c)* Segunda iteração. *(d)* Terceira iteração.

dados.

A Figura 8 apresenta as etapas para formulação do grafo de vizinhança recíproca. A partir de um conjunto de listas ranqueadas, apresentado em *(a)*, o algoritmo itera sobre as vizinhanças recíprocas. Esta iteração é representada por t_k, que inicia com valor $t_k = 1$ sendo incrementado até o tamanho k da vizinhança. Na primeira iteração, representada em *(b)*, é criada uma aresta de auto-relação para todos os elementos do conjunto, isto ocorre, pois, cada elemento sempre ocupa a primeira posição de sua lista ranqueada.

Na segunda iteração, apresentada em *(c)*, são criadas arestas para as relações de vizinhança recíproca contidas na segunda posição dos rankings do conjunto. Neste exemplo, o elemento A é ligado ao elemento B e o elemento C é ligado ao elemento D. Estas ligações aplicam um aumento de similaridade entre estes elementos, o qual será discutido mais adiante.

Finalmente, na terceira iteração, representada em *(d)*, são criadas arestas para relação de vizinhança recíproca contida na terceira posição das listas ranqueadas. Nesta etapa, somente os elemento B e C recebem uma nova aresta, porém, pela relação contida nas componentes conexas, os elementos A e D também receberão incrementos de similaridade. Ao final do procedimento, pelo fato dos elementos A e D terem sido conectados entre si antes da conexão com os próximos elementos de suas respectivas listas ranqueadas, F e G, haverá uma reordenação destas listas, baseada na distribuição natural do conjunto de dados.

O algoritmo *Reciprocal kNN Graph and Connected Components* utiliza quatro etapas de processamento para explorar um conjunto de listas ranqueadas recebidas como

entrada, as quais são descritas nos itens a seguir.

A. Normalização das Listas Ranqueadas

Como mencionado anteriormente, as listas ranqueadas não são simétricas. Desta maneira, aumentar a simetria entre as vizinhanças de tamanho k tende a beneficiar a eficácia dos resultados [Jegou et al., 2010].

Para isso, normalização das listas ranqueadas de entrada é proposta para obter um valor simétrico de similaridade entre os elementos. A similaridade combinada ρ_n é definida como:

$$\rho_n(i,j) = \tau_i(j) + \tau_j(i) + max(\tau_i(j), \tau_j(i)), \tag{3.1}$$

onde $\tau_i(j) \leq L$, $\tau_j(i) \leq L$. Como ambos os valores de $\tau_i(j)$ e $\tau_j(i)$ possuem valor máximo de L, o valor da similaridade combinada normalizada será de no máximo $\rho_n(i,j) \leq 3 \times L$.

Baseado nos novos valores de similaridade obtidos por ρ_n, as listas ranqueadas são atualizadas, através de um algoritmo de ordenação estável, até as L primeiras posições. Este novo conjunto $\mathcal{T}$ é utilizado para os próximos estágios do *manifold learning*.

B. Grafo de Vizinhos Recíprocos

O Grafo de vizinhos recíprocos é um grafo unidirecional $G_r = (V, E)$, onde V é a coleção de vértices, $V = \mathcal{C}$, onde cada objeto é representado por um nó. O conjunto de arestas E é obtido com base na vizinha recíproca de tamanho k, sendo considerados diferentes valores para k. Inicialmente, um conjunto de vizinhança é definido como:

$$\mathcal{N}(q,k) = \{\mathcal{S} \subseteq \mathcal{C}, |\mathcal{S}| = k \wedge \forall o_i \in \mathcal{S}, o_j \in \mathcal{C} \setminus \mathcal{S} : \tau_q(i) < \tau_q(j)\}.$$

Assim, $\mathcal{N}(q,k)$ representa um conjunto de objetos contidos nas k primeiras posições da lista τ_q, onde $\forall o_i \in \mathcal{N}(q,k), \tau_q(i) \leq k$. Com base nesta formalização, o conjunto de vizinhanças recíprocas pode ser definido como:

$$\mathcal{N}_r(q,k) = \{o_i \in \mathcal{N}(q,k) \wedge o_q \in \mathcal{N}(i,k)\}. \tag{3.2}$$

Para cada iteração de k, representada no algoritmo por t_k, as arestas do Grafo de k vizinhos próximos recíprocos podem ser obtidas pela seguinte equação:

$$E = \{(o_q, o_j) \mid o_j \in \mathcal{N}_r(q, t_k)\}. \tag{3.3}$$

Desta maneira, uma aresta será criada a partir de o_q até o_j se os dois objetos são vizinhos recíprocos até as t_k primeiras posições das respectivas listas ranqueadas.

C. Componentes Conexas

Como mencionado acima, as arestas criadas pelo relacionamento de vizinhança recíproca normalmente produzem grafos esparsos. Assim, as componentes conexas do grafo gerado são utilizadas para analisar a geometria natural do conjunto de dados por meio da expansão da vizinhança de similaridade.

Estas componentes conexas podem ser obtidas por meio de algoritmos de busca em grafos, como os algoritmos de Busca em Profundidade e Busca em Largura. O resultado desta análise para todo o grafo é um conjunto de componentes conexas $\mathcal{P} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$, tal que $\{\mathbf{c}_1 \cup \mathbf{c}_2 \cup \cdots \cup \mathbf{c}_m\} = \mathcal{P}$ and $\{\mathbf{c}_1 \cap \mathbf{c}_2 \cap \cdots \cap \mathbf{c}_m\} = \emptyset$.

É importante notar que o limiar t_k está diretamente ligado ao número de componentes conexas m: quanto maior o valor de t_k, mais conectado o grafo se torna diminuindo m [Pedronette; Gonçalves; Guilherme, 2018].

D. Distância Baseada no Grafo Vizinhos Recíprocos

Evidências sobre a similaridade entre os objetos estão codificadas tanto no conjunto de arestas E quanto no conjunto de componentes conexas $\mathcal{P}$. Estas duas fontes de informação são utilizadas para calcular uma pontuação de similaridade entre pares de objetos.

O Grafo G_r é atualizado para diferentes tamanhos de vizinhanças recíprocas e, para cada valor $t_k \leqslant k$, a pontuação de similaridade é incrementada de maneira que valores maiores são dados aos vizinhos nas primeiras posições das listas ranqueadas.

A pontuação de similaridade de arestas $w_e(i, j)$, entre os objetos o_i e o_j, é formalmente definida com base na conectividade do grafo.

Seja $E(q)$ o conjunto de nós para os quais o objeto o_q possui arestas dado um valor t_k. Cada par de objetos (o_i, o_j) contido em $E(q)$ representa um aumento de similaridade entre eles, sendo que os dois objetos possuem arestas para o_q. Desta maneira, $w_e(i, j)$ pode ser definido como:

$$w_e(i,j) = \sum_{t_k=1}^{k} \sum_{q \in \mathcal{C} \wedge i,j \in E(q)} (k - t_k + 1). \tag{3.4}$$

De maneira análoga, a informação fornecida pelas componentes conexas define a pontuação de similaridade de componentes $w_c(i, j)$. Esta pontuação representa um aumento de similaridade entre os objetos o_i e o_j quando eles estão na mesma componente conexa, a qual também é definida considerando diferentes valores de t_k:

$$w_c(i,j) = \sum_{tk=1}^{k} \sum_{i,j \in C_l} (k - t_k + 1). \tag{3.5}$$

Quanto mais cedo a conexão entre o_i e o_j for encontrada em G_r, maior serão os valores de $w_e(i,j)$ e $w_c(i,j)$, destacando a estrutura geométrica presente no conjunto de dados.

As duas pontuações de similaridade são consideradas para a definição de uma pontuação final $w(i,j)$:

$$w(i,j) = w_e(i,j) + w_c(i,j). \tag{3.6}$$

Finalmente, a distância baseada no grafo de k vizinhos recíprocos [Pedronette; Gonçalves; Guilherme, 2018], ρ_r, é inversamente proporcional à pontuação de similaridade e é obtida conforme a equação a seguir:

$$\rho_r(i,j) = \frac{1}{1 + w(i,j)}. \tag{3.7}$$

Baseado nesta nova distância ρ_r, um conjunto mais efetivo de listas ranqueadas $\mathcal{T}_r$ é obtido e pode ser explorado para a análise do conjunto de dados.

3.3 Método de agrupamento

Os métodos de agrupamento, como mencionado na Seção 2.1, buscam separar os objetos de um conjunto com o objetivo de maximizar a similaridade dos objetos contidos em um grupo e de minimizar a similaridade entre os grupos obtidos.

Desta maneira, a utilização de informações mais confiáveis obtidas por meio de uma técnica de *manifold learning*, pode produzir melhores agrupamentos do que técnicas que utilizam de métricas de distância convencionais, como a distância Euclidiana.

A partir desta hipótese, a metodologia proposta nesta implementação busca utilizar o conjunto de listas ranqueadas $\mathcal{T}_r$ e a função de distância ρ_r, obtidas pelo método *Reciprocal kNN Graph and Connected Components*, de duas maneiras: *(i)* Explorando as componentes conexas geradas a partir de vizinhanças recíprocas de baixa profundidade para obter agrupamentos iniciais com alta confiabilidade; *(ii)* Fazendo uso de ρ_r para escolher agrupamentos similares para aglomeração.

A. Agrupamentos Iniciais

As componentes conexas obtidas a partir do grafo de vizinhos recíprocos, conforme descritas na Subseção 3.2, podem ser utilizadas diretamente como agrupamentos. Tal que, um objeto o_i estará contido em uma e somente uma componente conexa e a união de todas as componentes conexas é equivalente ao conjunto de dados original.

Desta maneira, a simples saída baseada nas componentes conexas recuperadas em uma certa iteração t_k representa o agrupamento da estrutura do conjunto de dados.

Porém, mesmo que a relação de vizinhança recíproca forneça arestas mais confiáveis entre os objetos e componentes conexas, ainda é possível que arestas incorretas sejam incluídas. Assim, para conjuntos de dados com listas ranqueadas ineficientes, o algoritmo tende a unir componentes conexas não-similares, levando a um processo de agrupamento também ineficiente.

Para minimizar as uniões incorretas, as componentes conexas obtidas a partir de vizinhanças recíprocas com baixa profundidade, isto é, com valores baixos para t_k, são exploradas. Baseada em um novo parâmetro, c_k, que representa o tamanho das vizinhanças recíprocas para a geração dos agrupamentos iniciais, a vizinhança recíproca de o_q é redefinida utilizando o conjunto de listas ranqueadas $\mathcal{T}_r$ obtido pelo método de *manifold learning*:

$$\begin{aligned}\mathcal{N}_c(q, c_k) = \{\mathcal{S} \subseteq \mathcal{C}, |\mathcal{S}| \leq c_k \wedge \forall o_i \in \mathcal{S} : \\ {\tau_r}^q(i) \leq c_k \wedge {\tau_r}^i(q) \leq c_k\},\end{aligned} \tag{3.8}$$

onde ${\tau_r}^q(i)$ representa a posição de o_i na lista ranqueada de o_q, $\mathcal{T}_r(q)$.

Então, um novo grafo G_c é definido. Do qual o conjunto de arestas E_c pode ser obtido por:

$$E = \{(o_q, o_j) \mid o_j \in \mathcal{N}_c(q, c_k)\}. \tag{3.9}$$

Um novo conjunto de componentes conexas $\mathcal{P}_c = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$ é extraído a partir G_c. Este conjunto de componentes conexas dá origem a um conjunto de agrupamentos $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m\}$, onde m representa o número grupos obtidos do conjunto de dados e cada agrupamentos $\mathcal{S}_i \in \mathcal{S}$ é criado a partir da componente conexa $\mathbf{c}_i \in \mathcal{P}_c$ correspondente. O conjunto $\mathcal{S}$ é composto majoritariamente por grupos unitários e por poucos grupos não-unitários.

Enquanto os grupos unitários representam objetos que não estão inseridos em vizinhanças recíprocas de profundidade c_k, os grupos não-unitários representam agrupamentos confiáveis que serão utilizados para aprimorar a aglomeração em busca da separação final do conjunto de dados.

B. Agrupamentos Finais

Os agrupamentos iniciais $\mathcal{S}$, embora muito confiáveis, tendem a incluir poucos elementos. Desta forma, uma etapa de união aglomerativa é aplicada para inclusão de novos elementos e obtenção da separação final do conjunto de dados. Para isso, a metodologia proposta itera sobre $\mathcal{S}$, descrito no item anterior.

A cada iteração, o menor agrupamento, representado por $\mathcal{S}_A$, é unido ao agrupamento mais próximo no conjunto. Para calcular a distância entre dois agrupamentos $\mathcal{S}_A$ e $\mathcal{S}_B$, a função $d(\mathcal{S}_A, \mathcal{S}_B)$ é definida como:

$$d(\mathcal{S}_A, \mathcal{S}_B) = \frac{1}{|\mathcal{S}_A||\mathcal{S}_B|} \sum_{a \in \mathcal{S}_A} \sum_{b \in \mathcal{S}_B} \rho_r(a, b), \tag{3.10}$$

onde ρ_r representa a função de distância aprimorada obtida pelo algoritmo de *manifold learning*.

A função $d(\mathcal{S}_A, \mathcal{S}_B)$ calcula a distância entre os agrupamentos $\mathcal{S}_A$ e $\mathcal{S}_B$ com base no tipo de ligação *average linkage*. Esta abordagem foi escolhida para explorar os agrupamentos iniciais, visto que estes agrupamentos apresentam alta confiabilidade e a utilização de todos os membros para o cálculo da distância leva o algoritmo a escolher melhores pares para união.

Baseado em $d(\mathcal{S}_A, \mathcal{S}_B)$, o agrupamento mais próximo à $\mathcal{S}_A$, representado por $f(\mathcal{S}_A)$, é recuperado:

$$f(\mathcal{S}_A) = \underset{\mathcal{S}_B \in \mathcal{S} \setminus \{\mathcal{S}_A\}}{\arg\min}\ d(\mathcal{S}_A, \mathcal{S}_B). \tag{3.11}$$

Com os dois agrupamentos encontrados, $\mathcal{S}_A$ é atualizado pela adição dos elementos de seu agrupamento mais próximo, $f(\mathcal{S}_A)$:

$$\mathcal{S}_A = \mathcal{S}_A \cup f(\mathcal{S}_A). \tag{3.12}$$

A seguir, o conjunto $\mathcal{S}$ é atualizado pela remoção de $f(\mathcal{S}_A)$:

$$\mathcal{S} = \mathcal{S} \setminus f(\mathcal{S}_A). \tag{3.13}$$

Tabela 1 – Descrição de parâmetros do método *C-ReckNN*.

Termo	**Descrição**
k	Principal parâmetro utilizado no método. Delimita o tamanho da vizinhança recíproca explorada pelo método de *manifold learning* e o número mínimo de objetos contidos em cada agrupamento final.
L	Define o tamanho considerado em cada lista ranqueada, isto é, quantas posições serão normalizadas e reordenadas durante as etapas do método.
c_k	Tamanho da vizinhança recíproca utilizada para criação dos agrupamentos iniciais do conjunto.

Este processo é repetido até que a condição de parada, a qual é baseada no parâmetro k, seja alcançada:

$$\forall \mathcal{S}_i \in \mathcal{S} : |\mathcal{S}_i| \leq k. \tag{3.14}$$

Após a união dos agrupamentos, a metodologia proposta recupera um agrupamento hierárquico aglomerativo com conexão *average linkage* [Saxena et al., 2017].

Por fim, a Tabela 1 resume os parâmetros esperados pela metodologia proposta. São eles: *(i)* k vizinhança explorada, *(ii)* L delimitador do tamanho das listas ranqueadas, *(iii)* c_k vizinhança explorada para obtenção dos agrupamentos iniciais.

4 Agrupamento Auto-Supervisionado via Redes Convolucionais baseadas em Grafos

Este capítulo apresenta o método *Self-supervised Graph Convolutional Clustering (SGCC)*[1]. A Seção 4.1 apresenta uma descrição geral do método e das etapas envolvidas no processo de agrupamento. A Seção 4.2 detalha o *Log-based Hypergraph of Ranking References (LHRR)* [Pedronette et al., 2019], método de *manifold learning* utilizado. A Seção 4.3 apresenta as redes neurais baseadas em grafos e lista variantes utilizadas na criação deste método. Por fim, a Seção 4.4 detalha como o *SGCC* explora as informações fornecidas pelo método de *manifold learning* para criação de agrupamentos iniciais confiáveis e como é realizado o treinamento da rede neural baseada em grafo para a definição dos agrupamentos finais do conjunto de dados.

4.1 Visão Geral

O *SGCC* utiliza informações obtidas por um método de *manifold learning* baseado em hiper-grafos [Pedronette et al., 2019] para identificar e realizar a aglomeração de uma parcela do conjunto em grupos, criando uma configuração inicial de agrupamentos para o conjunto. Esta configuração inicial é, posteriormente, utilizada como rótulo, em conjunto com outras informações, para treinar uma rede neural baseada em grafos, que realiza o agrupamento final do conjunto de dados.

A Figura 9 apresenta o processo realizado pelo método proposto para identificar os agrupamentos e separar os elementos presentes no conjunto de dados de entrada. Inicialmente, uma métrica de distância é utilizada para formular listas ranqueadas para cada elemento do conjunto a partir de vetores de características, como os extraídos por *CNN*s. As listas ranqueadas obtidas são utilizadas como entrada para o *LHRR*, método de *manifold learning* que realiza a criação de um hipergrafo baseado na relação de vizinhança entre os itens. O *LHRR* fornece três estruturas de dados importantes para o método proposto: *(i)* a matriz de incidência, a qual contém as relações formadas entre os elementos, além de pontuação para cada uma destas relações, *(ii)* a pontuação de peso das hiperarestas contidas no hipergrafo e *(iii)* novas listas ranqueadas aprimoradas para o conjunto de entrada, utilizadas para criação de um grafo de vizinhos recíprocos.

Explorando as informações fornecidas, uma nova medida de estimativa de eficácia é definida para a criação de um ranqueamento que contém uma ordenação do conjunto, onde os primeiros elementos possuem relações mais confiáveis. Essa nova medida, denominada

[1] Implementação futuramente disponível em <https://github.com/lopes-leonardo/sgcc>

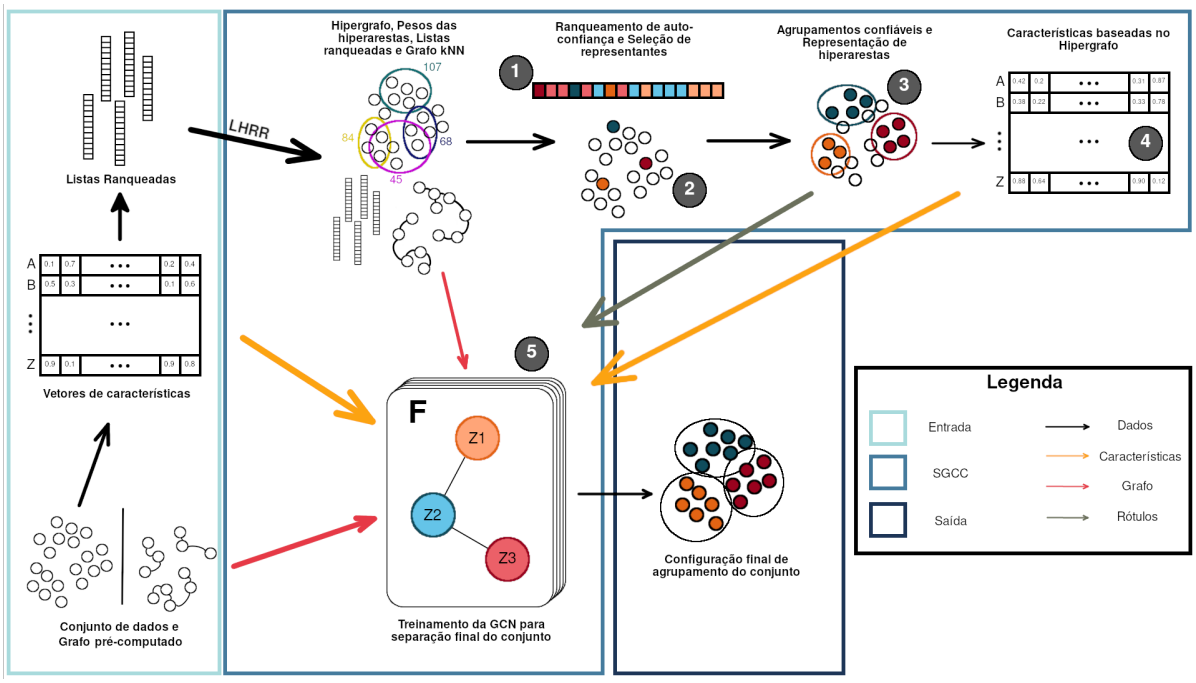


Figura 9 – Fluxo de dados implementado no *SGCC*

pontuação de auto-confiança é aplicada para identificar elementos do conjunto que representarão cada um dos agrupamentos contidos nos dados. Após a seleção de um elemento representativo para cada agrupamento, um novo conjunto de hiperarestas é formulado por meio da cópia das hiperarestas dos itens selecionados. Cada uma destas hiperarestas representa seu respectivo agrupamento, sendo incrementada conforme novos elementos são aglomerados a ele.

Após a criação dos agrupamentos unitários e de suas respectivas hiperarestas, uma parcela do conjunto é aglomerada iterativamente. Nesta etapa, duas funções de similaridade distintas podem ser utilizadas para seleção do agrupamento em que cada item será aglomerado. Após cada união, a hiperaresta do agrupamento é incrementada com os valores contidos na hiperaresta do elemento aglomerado. Ao fim das uniões, o método proposto recupera uma configuração inicial de agrupamentos confiáveis, a qual contém uma parcela do conjunto de dados definida com base em um parâmetro recebido como entrada pelo algoritmo, e um conjunto de hiperarestas que representam esses agrupamentos.

Em uma etapa opcional, o conjunto de hiperarestas de agrupamento pode ser utilizado para formulação de uma nova coleção de vetores de características para o conjunto de dados. Estes novos vetores de características são obtidos por meio do produto interno entre a hiperaresta de cada elemento do conjunto e as hiperarestas dos agrupamentos. Esta nova representação dos elementos de entrada tende a ser menor que os vetores de características utilizados inicialmente, diminuindo o tempo de processamento e o uso de memória para o treinamento do modelo de rede neural.

Por fim, o grafo de vizinhos recíprocos obtido das listas ranqueadas aprimoradas e os vetores de características do conjunto de dados são utilizados como entrada para um modelo de rede neural baseado em grafos, o qual é treinado com base na configuração inicial dos agrupamentos. Após o treinamento, o modelo é utilizado para agrupar todos os elementos, obtendo uma configuração final auto-supervisionada de agrupamentos do conjunto de dados.

4.2 Hipergrafo de Referências de Rankings baseadas em Logaritmos

O *Log-based Hypergraph of Ranking References (LHRR)* [Pedronette et al., 2019] busca encontrar uma relação global de similaridade entre os elementos do conjunto explorando informações estruturais do conjunto de dados. Para isso, o método constrói um hipergrafo baseado nas relações presentes no ranqueamento dos elementos, recebido como entrada do algoritmo.

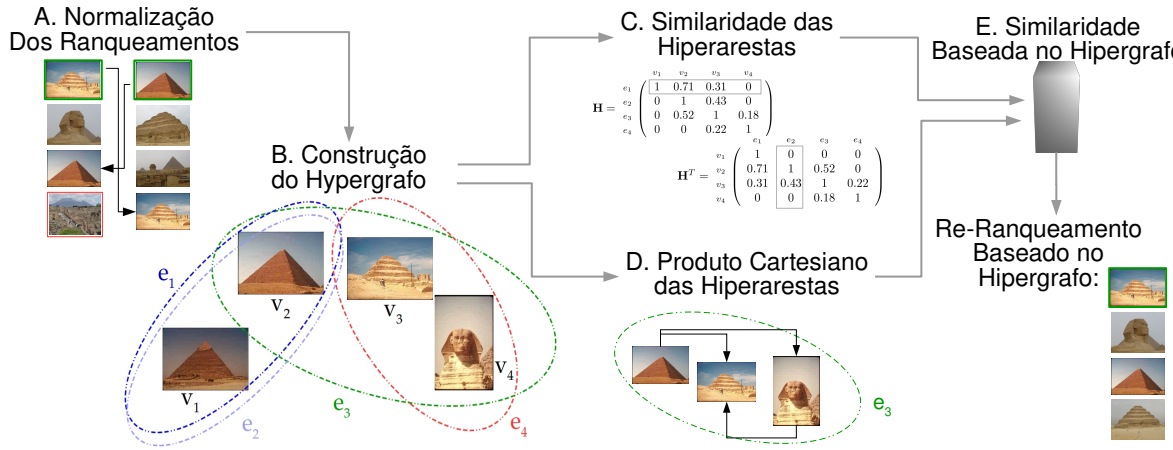


Figura 10 – Etapas de construção e análise das relações do hipergrafo. Traduzido de: [Pedronette et al., 2019].

A Figura 10 apresenta o fluxo de processamento executado pelo *LHRR*. O processo, que resulta na obtenção de uma nova medida de similaridade para o conjunto de dados, pode ser dividido em cinco etapas, descritas a seguir:

A. Normalização das Listas Ranqueadas

Com objetivo de aprimorar a simetria entre as vizinhanças do conjunto de listas ranqueadas $\mathcal{T}$, recebido como entrada, uma nova medida de similaridade normalizada ρ_n é definida como:

$$\rho_n(i,j) = 2L - (\tau_i(j) + \tau_j(i)) \tag{4.1}$$

onde $L \leqslant n$ e $\tau_i(j)$ representa a posição do elemento o_j na lista ranqueada do elemnto o_i.

Baseada em ρ_n, o conjunto $\mathcal{T}$ é ordenado até as L primeiras posições, por um algoritmo de ordenação estável.

B. Construção do Hipergrafo

Um hipergrafo é uma generalização do grafo tradicional, onde cada hiperaresta é um subconjunto do conjunto de vértices, podendo, portanto, conectar mais que dois elementos simultaneamente [Huang et al., 2010].

Dado um hipergrafo $G_h = (V, E, w)$, V representa o conjunto finito dos vértices e E representa o conjunto das hiperarestas, o qual pode ser definido como uma família de subconjuntos de V, de maneira que $\bigcup_{e \in E} = V$. Cada vértice $v_i \in V$ está associado a um objeto $o_i \in \mathcal{C}$, onde $\mathcal{C}$ representa o conjunto de dados explorado. Para cada hiperaresta e_i, existe um peso positivo $w(e_i)$, o qual representa a confiabilidade dos relacionamentos estabelecidos por e_i.

Uma hiperaresta e_i é descrita como *incidente* sobre um vértice v_j quando $v_j \in e_i$. Desta maneira, um hipergrafo pode ser representado por uma matriz de incidência $\mathbf{H}_b$ de tamanho $|E| \times |V|$:

$$h_b(e_i, v_j) = \begin{cases} 1, & \text{se } v_j \in e_i, \\ 0, & \text{caso contrário.} \end{cases} \tag{4.2}$$

Porém, a matriz $\mathbf{H}_b$ representa uma associação binária entre o conjunto de vértices V e o conjunto de hiperarestas E. Esta associação não é suficiente para representar cenários onde é desejável um grau de incerteza nas relações representadas. Para sobrepor esta limitação, hipergrafos probabilísticos representam a possibilidade de um vértice pertencer a uma hiperaresta. Seja $r : E \times V \to \mathbb{R}^+$ uma função que representa esta probabilidade, a matriz de incidência contínua $\mathbf{H}$ pode ser definida como:

$$h(e_i, v_j) = \begin{cases} r(e_i, v_j), & \text{se } v_j \in e_i, \\ 0, & \text{caso contrário.} \end{cases} \tag{4.3}$$

Ao início do processamento, uma hiperaresta e_i é definida para cada objeto $o_i \in C$. Seja $o_x \in \mathcal{N}(i, k)$ um vizinho do objeto o_i e $o_j \in \mathcal{N}(x, k)$ um vizinho do objeto o_x. A medida $r(e_i, v_j)$, que representa o grau de pertencimento do vértice v_j a hiperaresta e_i, é calculada como:

$$r(e_i, v_j) = \sum_{o_x \in \mathcal{N}(i,k) \wedge o_j \in \mathcal{N}(x,k)} w_p(i, x) \times w_p(x, j), \tag{4.4}$$

onde $w_p(i, x)$ é a função que determina um peso de relevância do objeto o_x de acordo com sua posição na lista ranqueada $\mathcal{T}_i$, sendo definida como:

$$w_p(i, x) = 1 - \log_k \tau_i(x). \tag{4.5}$$

A função $w_p(i, x)$ atribui o valor 1 para a primeira posição e rapidamente decai ao decorrer das demais posições do ranqueamento. Atuando desta maneira, ela atribui um alto valor para as primeiras posições do ranqueamento, onde a efetividade é superior.

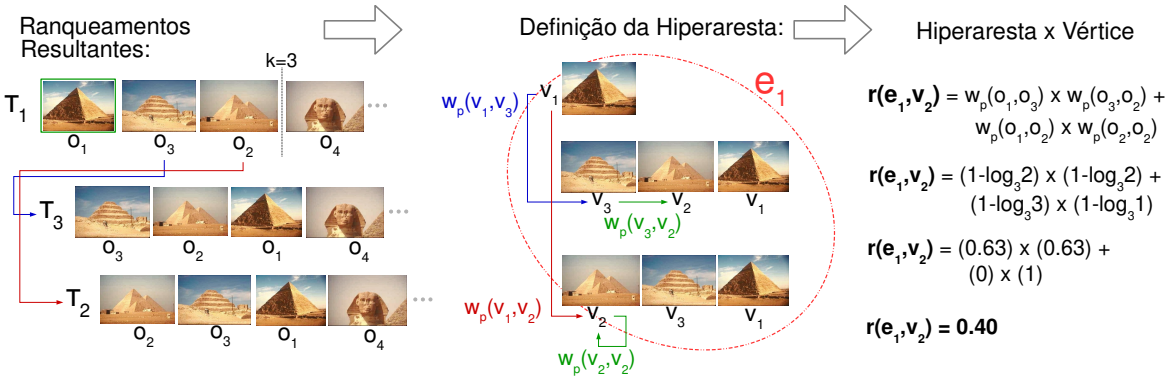


Figura 11 – Ilustração da definição de uma hiperaresta baseada em referências de ranqueamento com vizinhança de tamanho $k = 3$. Traduzido de: [Pedronette et al., 2019].

A Figura 11 ilustra como a referência do ranqueamento é explorada para definir a hiperaresta e_1 e como é calculado o relacionamento entre a hiperaresta e um vértice. No exemplo, a hiperaresta e_1 é definida com base na lista ranqueada $\mathcal{T}_1$ e em suas respectivas referências para $\mathcal{T}_2$ e $\mathcal{T}_3$. A relação entre o vértice v_j e a hiperaresta e_1 é representada por $r(e_1, v_j)$, a qual é calculada a partir dos pesos obtidos da função w_p.

O peso de uma hiperaresta $w(e_i)$ representa a confiança dos relacionamentos estabelecidos por ela. Para calcular $w(ei)$, um conjunto de vizinhança de hipergrafo $\mathcal{N}_h$ é definido. Dado uma hiperaresta e_i, o conjunto $\mathcal{N}_h$, contendo os vértices com os maiores valores $h(e_i, \cdot)$, é formalmente definido como:

$$\begin{aligned}\mathcal{N}_h(q, k) = \{\mathcal{S} \subseteq e_q, |\mathcal{S}| = k \wedge \forall o_i \in \mathcal{S}, o_j \in e_q - \mathcal{S} : \\ h(q, i) > h(q, j)\}.\end{aligned} \tag{4.6}$$

Por fim, o peso de um hiperaresta $w(e_i)$ pode ser definido como:

$$w(e_i) = \sum_{j \in \mathcal{N}_h(i,k)} h(i, j). \tag{4.7}$$

C. Similaridade das Hiperarestas

Os hipergrafos representam um poderoso modelo para representar relacionamentos de alta-complexidade, porém, em certas circunstâncias como a elaboração de ranqueamentos, é necessário extrair informação de similaridade par-a-par. O *LHRR* explora a informação de similaridade existente nas hiperarestas para calcular uma matriz de similaridade par-a-par $\mathbf{S}$. Esta similaridade é calculada com base em duas hipóteses diferentes [Pedronette et al., 2019], as quais são combinadas para obtenção da similaridade final entre os elementos.

A primeira hipótese é que objetos similares apresentam listas ranqueadas similares e, consequentemente, hiperarestas similares. A partir da informação contida na matriz de incidência $\mathbf{H}$, a similaridade entre duas hiperarestas e_i, e_j, pode ser calculada pela soma dos valores h multiplicados nos vértices correspondentes. Esta operação pode ser calculada pela multiplicação da matriz de incidência $\mathbf{H}$ pela sua transposta:

$$\mathbf{S}_h = \mathbf{H}\mathbf{H}^T \tag{4.8}$$

A segunda hipótese atesta que, objetos similares são supostamente referenciados pelas mesmas hiperarestas. Desta maneira, para calcular a similaridade par-a-par entre dois vértices v_i e v_j, os valores h nas hiperarestas correspondentes devem ser multiplicados. Esta operação pode ser calculada pela multiplicação da transposta da matriz $\mathbf{H}$ por sua versão original:

$$\mathbf{S}_v = \mathbf{H}^T\mathbf{H} \tag{4.9}$$

Como ambas as similaridades entre hiperarestas e entre vetores contém informações relevantes e complementares, elas são combinadas por meio da multiplicação elemento por elemento $s(i,j) = s_h(i,j) \times s_v(i,j)$. Desta maneira, a matriz de similaridade par-a-par pode ser calculada pelo produto de Hadamard:

$$\mathbf{S}_p = \mathbf{S}_h \circ \mathbf{S}_v \tag{4.10}$$

D. Produto Cartesiano entre as Hiperarestas

Como discutido anteriormente, uma hiperaresta contém um subconjunto de vértices. Com o objetivo de extrair o relacionamento par-a-par direto entre um conjunto de elementos definidos por uma hiperaresta, o produto cartesiano é aplicado. Sejam duas hiperarestas e_q, $e_i \in E$, o produto cartesiano entre elas é definido como:

$$e_q \times e_i = \{(v_x, v_y) : \ v_x \in e_q \wedge v_y \in e_i\}. \tag{4.11}$$

Seja ${e_q}^2$ o produto Cartesiano entre os elementos de uma mesma hiperaresta e_q. Para cada par de vértices $(v_i, v_j) \in {e_q}^2$, um relacionamento de similaridade $p : E \times V \times V \rightarrow \mathbb{R}^+$ é estabelecido e pode ser definido como:

$$p(e_q, v_i, v_j) = w(e_q) \times h(e_q, v_i) \times h(e_q, v_j). \tag{4.12}$$

Por fim, a matriz $\mathbf{P}_c$ é construída considerando os relacionamentos contidos em todas as hiperarestas, sendo calculada como:

$$\mathbf{P}_c(i, j) = \sum_{e_q \in E \wedge (v_i, v_j) \in {e_q}^2} p(v_i, v_j) \tag{4.13}$$

E. Medida de Similaridade Baseada no Hipergrafo

As informações de similaridade presentes nas hiperarestas e nos produtos Cartesianos são distintas e complementares, representando a distribuição natural do conjunto de dados. Desta maneira, as duas informações são exploradas para o cálculo de uma matriz de afinidade $\mathbf{W}$, a qual combina $\mathbf{P}_c$ e $\mathbf{S}_p$, da seguinte maneira:

$$\mathbf{W} = \mathbf{P}_c \circ \mathbf{S}_p \tag{4.14}$$

Baseado nos valores da matriz de afinidade $\mathbf{W}$, um novo conjunto de listas ranqueadas é calculado para os dados explorados. Como o *LHRR* representa a entrada e saída do método em formato de listas ranqueadas, o algoritmo pode ser repetido iterativamente, em busca de melhores resultados. Seja o sobrescrito $^{(t)}$ a representação da iteração atual, o conjunto de listas ranqueadas $\mathcal{T}^{(t+1)}$ é computado com base na matriz de similaridade $\mathbf{W}^{(t)}$. Neste cenário, a representação inicial das características do conjunto define a entrada $\mathcal{T}^{(0)}$.

Após o algoritmo *LHRR* ser executado por T iterações, as listas ranqueadas de saída $\mathcal{T}^{(t+1)}$, o conjunto $E^{(t)}$ de hiperarestas e seus respectivos pesos codificam informações de similaridade relevantes sobre o conjunto de dados. Tais informações são exploradas pelo método *SGCC* proposto, conforme descrição apresentada na Seção 4.4.

4.3 Redes Neurais baseadas em Grafos

Estratégias baseadas em aprendizado profundo (*deep learning*) têm atingido resultados de alta eficácia em diversas áreas do conhecimento, como a visão computacional e o processamento de linguagem natural. Dentre as várias arquiteturas existentes, pode-se destacar os resultados obtidos pelas Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNNs) [Khan et al., 2020]. Contudo, as CNNs são comumente aplicadas em

estruturas de dados de vizinhança organizadas em grades Euclidianas, como imagens. Mais recentemente, esforços recentes tem sido realizados para aplicar técnicas de aprendizado profundo em dados representados como grafos [Cai; Zheng; Chang, 2018], que permitem arranjos de vizinhança em cenários mais complexos e não Euclidianos. Neste cenário, as *Graph Convolutional Networks (GCN)* são um relevante modelo de redes neurais, introduzido em [Kipf; Welling, 2017], que visa, de maneira simplificada, aprender uma representação (*embedding*, em inglês) para cada nó presente no grafo de entrada através de uma agregação iterativa das representações de seus vizinhos. Essa abordagem permite que a estrutura do grafo de entrada seja codificada diretamente em um modelo de rede neural.

Conforme citado acima, [Kipf; Welling, 2017] utilizam uma *GCN* de duas camadas para classificação supervisionada e semi-supervisionada, a partir de um grafo representado por uma matriz simétrica de adjacências $\mathbf{A}$. Seu modelo pode ser descrito como uma função que recebe vetores de características $\mathbf{X}$ e uma matriz de adjacências $\mathbf{A}$ como entrada:

$$\mathbf{Z} = f(\mathbf{X}, \mathbf{A}), \tag{4.15}$$

Neste cenário, $\mathbf{Z}$ denota uma matriz de representações definida como $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n]^T \in \mathbb{R}^{n \times c}$, sendo cada $\mathbf{z}_i$ uma representação de c dimensões aprendida para o nó v_i. A função que define o cálculo de $\mathbf{Z}$ é formalmente definida a seguir. Inicialmente, em uma etapa de pré-processamento, os graus das matrizes são calculados como $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$, em que $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ e $\tilde{\mathbf{D}}$ é a matriz de grau de $\tilde{\mathbf{A}}$. Assim, o modelo GCN de duas camadas é representado pela função $f(\cdot)$, definida como:

$$\mathbf{Z} = log(softmax(\hat{\mathbf{A}} ReLU(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(0)})\mathbf{W}^{(1)})) \tag{4.16}$$

Os pesos da rede neural para a entrada da camada escondida H, que realiza o mapeamento para pesos internos, são definidos pela matriz $\mathbf{W}^{(0)} \in \mathbb{R}^{d \times H}$. Após isso, a camada $\mathbf{W}^{(1)} \in \mathbb{R}^{H \times c}$ faz a conversão da camada escondida para uma matriz de saída. Ambas as matrizes são treinadas com a utilização do gradiente descendente baseado no erro de entropia cruzada sobre todos os nós rotulados, os quais podem ser todos os nós ou somente parte deles, durante o aprendizado semi-supervisionado.

Por fim, a função de ativação *softmax* é aplicada e determina uma distribuição de probabilidade sobre as c classes para cada linha, onde a soma destas probabilidades resulta em 1. Após a aplicação da função logarítmica sobre estas probabilidades, a classe com valor menos negativo é atribuída ao nó v_i no espaço de representação dado pelo vetor $\mathbf{z}_i$, resultando em um dos valores contidos no conjunto $\mathcal{Y} = \{y_1, y_2, ;y_c\}$ de rótulos.

Após o sucesso da primeira GCN, diversos modelos de redes neurais baseadas em grafo têm sido propostos [Klicpera; Bojchevski; Günnemann, 2019; Bianchi et al., 2021; Veličković et al., 2018; Wu et al., 2019]. Neste trabalho, além da GCN proposta

em [Kipf; Welling, 2017], outras duas redes foram selecionadas com base em trabalhos recentes [Pedronette; Latecki, 2021] e resultados obtidos em um conjunto inicial de experimentos: *(i) Simple Graph Convolution (SGC)* [Wu et al., 2019], um modelo de GCN simplificada com a remoção das ativações não-lineares e com a aplicação de normalização nos pesos presentes entre camadas consecutivas, e *(ii) Approximate Personalized Propagation of Neural Predictions (APPNP)* [Klicpera; Bojchevski; Günnemann, 2019], que explora o relacionamento entre as GCNs e ranqueamento, propondo uma estratégia de propagação baseada em um modelo personalizado de ranqueamento.

4.4 Método de Agrupamento

O *SGCC* utiliza informações obtidas através do método de *manifold learning LHRR* para separar um conjunto de dados de entrada. Em um primeiro momento, o *LHRR* é executado utilizando k como tamanho da vizinhança e $L = 4 \times k$, que define a profundidade de exploração das listas ranqueadas de entrada. Após a execução do método de *manifold learning* por t iterações, as hiperarestas, seus respectivos pesos e as listas ranqueadas aprimoradas obtidas são processadas por cinco etapas, ilustradas na Figura 9 e descritas nos itens a seguir.

1. Pontuação de Auto-confiança da Hiperaresta

Conforme mencionado na Subseção 4.2, o hipergrafo gerado pelo *LHRR* contém, para cada objeto $o_i \in C$, um vértice v_i e uma hiperaresta e_i relacionados entre si. Da mesma maneira, todo vértice $v_i \in V$ possui um grau de pertencimento $r(e_j, v_i)$ para toda hiperaresta $e_j \in E$ e toda hiperaresta $e_i \in E$ possui um peso $w(e_i)$ que representa sua confiabilidade baseada nos graus de pertencimento de todos os vértices que a compõem.

Estes três indicadores apresentam informações complementares sobre o relacionamento entre os elementos e podem ser explorados para elencar elementos de alta confiabilidade, ou seja, que apresentam informações eficazes de ranqueamento e, por conseguinte de similaridade, em relação a outros elementos do conjunto de dados.

O grau de pertencimento de um vértice à sua própria hiperaresta, definido por $r(e_i, v_i)$, está associado ao número de aparições do vértice v_i nas primeiras posições das listas ranqueadas de seus respectivos vizinhos. Assim, baseada no peso da hiperaresta e no grau de pertencimento descrito acima, a pontuação de auto-confiança $s : V \to \mathbb{R}^+$ é definida, para todos os objetos do conjunto, como:

$$s(o_i) = w(e_i) \times r(e_i, v_i). \tag{4.17}$$

Baseado na pontuação definida s, a lista ranqueada $\tau_s = (o_1, o_2, ..., o_n)$ pode ser definida como uma permutação do conjunto $\mathcal{C}$ obtida por um algoritmo de ordenação

estável, de maneira que $\forall o_i, o_j \in \tau_s, \tau_s(o_i) < \tau_s(o_j) \Leftrightarrow s(o_i) > s(o_j)$. Esta nova lista ranqueada é utilizada para definir a ordem em que os elementos do conjunto são explorados durante as próximas etapas do algoritmo.

2. Seleção de Elementos Representantes

A lista ranqueada τ_s representa uma ordenação do conjunto de dados em termos de confiabilidade das informações de ranqueamento e similaridade. Dessa forma, os elementos que apresentam informações de similaridade e ranqueamento confiáveis são alocados nas primeiras posições. Contudo, os diferentes grupos do conjunto de dados podem apresentar níveis de heterogêneos de representatividade, ou seja, as primeiras posições da lista τ_s podem não representar todas as classes contidas no conjunto. Desta maneira, somente considerar os elementos na ordem obtida pela função s pode resultar em uma configuração ineficaz de agrupamentos para o conjunto explorado.

Para sobrepor essa limitação, uma seleção de elementos representantes é proposta com o objetivo de definir objetos com alta confiabilidade e suficientemente distintos para direcionar a definição dos grupos nas etapas seguintes. O conjunto $\mathcal{R} = (o_1, o_2, \ldots, o_c)$ é um subconjunto de $\mathcal{C}$, tal que $|\mathcal{R}| = c$, que contém os representantes selecionados para o conjunto de dados, em que cada $o_i \in \mathcal{R}$ é selecionado conforme a seguinte equação:

$$o_i = \underset{o_j \in \mathcal{C} \setminus \mathcal{R}}{\arg\max} \frac{s(o_j)}{1 + \sum_{o_k \in \mathcal{R}_{i-1}} r(e_k, v_j)}, \tag{4.18}$$

onde $\mathcal{R}_{i-1} = (o_1, \ldots, o_{i-1})$ é o conjunto de elementos representantes escolhidos em iterações anteriores. A Equação 4.18 pode ser descrita da seguinte maneira: selecione o próximo candidato que tenha uma alta pontuação de auto-confiança (numerador) e baixa similaridade com os elementos escolhidos em iterações anteriores (denominador).

O conjunto $\mathcal{R}$ é iniciado com o primeiro elemento contido na lista ranqueada τ_s, tal que $\tau_s(o_i) = 0$, e $c - 1$ iterações são realizadas para selecionar o restante dos representantes. Após a formulação de $\mathcal{R}$, o conjunto $\mathcal{S}$ pode ser definido como um conjunto de agrupamentos, tal que $|\mathcal{S}| = c$ e $\forall \mathcal{S}_i \in \mathcal{S}$, $\mathcal{S}_i = \{r_i \in \mathcal{R}\}$. Desta maneira, um agrupamento unitário é criado para cada um dos elementos selecionados pela Equação 4.18.

3. Agrupamentos Iniciais Confiáveis e Representação dos Grupos

A hiperaresta é uma poderosa representação de relacionamento entre múltiplos elementos do conjunto. Por conseguinte, tais estruturas também são exploradas para representar os agrupamentos, com base nas hiperarestas de todos os seus elementos. Para isso, as *hiperarestas de agrupamento* podem ser definidas como a matriz de incidência $\mathbf{H}_s$ de valores reais com tamanho $|S| \times |V|$, representando o grau de pertencimento dos

elementos do conjunto para cada um dos agrupamentos criados, podendo ser descrita como:

$$h_s(\mathcal{S}_i, v_j) = \sum_{e_k \in \mathcal{S}_i} h(e_k, v_j), \tag{4.19}$$

onde os valores de $\mathbf{H}_s$ são atualizados a cada nova aglomeração realizada nesta etapa do processamento.

Após a criação dos agrupamentos, uma parcela do conjunto é aglomerada iterativamente com o objetivo de gerar uma configuração inicial que será posteriormente utilizada como rótulo de treinamento para a *GCN*. Assim, para cada item o_i aglomerado nesta etapa, a função $nc : C \rightarrow \mathcal{S}$ obtém o agrupamento destino, sendo este o mais similar a o_i. Duas equações $nc(o_i)$ são propostas explorando aspectos diferentes das informações contidas nas hiperarestas do agrupamento. Estas equações são detalhadas nos itens *A* e *B*, enquanto o restante da etapa de aglomeração é descrito no item *C*.

A. Média Ponderada de Pertencimento

O grau de pertencimento ao agrupamento, contido na matriz $\mathbf{H}_s$, representa uma somatória de todos os valores de similaridade entre as hiperarestas dos elementos do agrupamento e outro item do conjunto de dados. Porém, mesmo que um elemento o_i esteja presente somente em uma parte das hiperarestas que formam o agrupamento, o valor de $h_s(\mathcal{S}_j, v_i)$ será alto caso estas relações sejam de alta similaridade. Este comportamento pode ocasionar aglomerações incorretas em conjunto com listas ranqueadas pouco confiáveis.

Além disso, utilizar somente o valor h_s para identificação do melhor agrupamento destino para um determinado elemento o_i, pode ocasionar a concentração de grande parte do conjunto em um único agrupamento com muitos elementos, o que não é o cenário ideal. Visando evitar este comportamento, uma *Média de Pertencimento* é proposta.

Seja o_i o próximo elemento do conjunto a ser aglomerado, o agrupamento mais similar à o_i, $\mathcal{S}_j$, pode ser definido pela seguinte equação:

$$nc(o_i) = \underset{\mathcal{S}_k \in \mathcal{S}}{\arg\max} \ \frac{h_s(\mathcal{S}_k, o_i)}{|\mathcal{S}_k|}. \tag{4.20}$$

B. Produto Intero das Hiperarestas

Explorando a similaridade direta entra as hiperarestas dos elementos e dos agrupamentos, uma segunda equação para definir o destino da aglomeração pode ser definida como o *Produto Interno das Hiperarestas*. Neste cenário, seja $\mathbf{h}_j = \{h_s(\mathcal{S}_j, o_1), \ldots, h_s(\mathcal{S}_j, o_n)\}$ um vetor contendo o grau de pertencimento de todos os elementos do conjunto em relação ao agrupamento $\mathcal{S}_j$, onde $n = |\mathcal{C}|$, o agrupamento $\mathcal{S}_j$ mais similar ao elemento o_i pode ser

definido pela seguinte equação:

$$nc(o_i) = \underset{\mathcal{S}_k \in \mathcal{S}}{\arg\max} \ \mathbf{h}_j \cdot e_i. \tag{4.21}$$

Diferentemente da Equação 4.20, esta abordagem considera a relação do elemento o_i com todos os elementos do conjunto e não somente com os elementos contidos no agrupamento analisado. Porém, não há um mecanismo para considerar o tamanho atual dos conjuntos durante a aglomeração.

C. Aglomeração Ordenada

Utilizando uma das abordagens descritas acima, uma parcela do conjunto é aglomerada para formação de agrupamentos iniciais. Assim, seja $q = \lfloor n \times p \rceil$ o tamanho da parcela do conjunto classificada nesta etapa, onde $n = |\mathcal{C}|$ e $p \in [0.1, 0.9]$ é um parâmetro recebido como entrada pelo algoritmo, $\mathcal{C}_a$ é o conjunto que contém todos os elementos separados para aglomeração, podendo ser definido como:

$$\mathcal{C}_a = \{\mathcal{C}_a \subseteq \mathcal{C} \setminus \mathcal{R}, |\mathcal{C}_a| = q - c \wedge \forall o_i \in \mathcal{C}_a, o_j \in \mathcal{C} \setminus \mathcal{C}_a : s(o_i) > s(o_j)\}. \tag{4.22}$$

Os itens contidos em $\mathcal{C}_a$ são aglomerados iterativamente com base em uma das duas abordagens propostas para a equação $nc(o_i)$. Desta maneira, cada agrupamento inicial $\mathcal{S}_j \in \mathcal{S}$ pode ser definido pela seguinte equação:

$$\mathcal{S}_j = \bigcup_{o_i \in \mathcal{C}_a \bigwedge nc(o_i) = \mathcal{S}_j} \{o_i\} \tag{4.23}$$

É importante notar que a função $nc(o_i)$ utiliza o estado atual de todos os agrupamentos para encontrar o destino mais similar ao item aglomerado. Desta maneira, esta etapa utiliza a ordenação contida na lista ranqueada τ_s para direcionar as uniões realizadas, explorando a confiança das hiperarestas dos primeiros elementos para ajudar nas aglomerações posteriores.

Após a aglomeração de todos os itens contidos no conjunto $\mathcal{C}_a$, o *SGCC* recupera uma configuração inicial de alta confiabilidade de agrupamentos contendo uma parcela dos elementos do conjunto de entrada, a qual será utilizada como rótulo de treinamento para a separação final realizada pela *GCN*.

4. Representação Baseada no Hipergrafo

Ao processar grandes conjuntos de dados, métodos de aprendizado de máquina podem apresentar limitações em função de capacidade de processamento ou quantidade de memória disponível no ambiente de execução. Assim, métodos de redução de dimensionalidade buscam encontrar novas representações (*embeddings* em inglês) de menor

dimensionalidade sem perder as informações fornecidas pelo conjunto de características original [Maćkiewicz; Ratajczak, 1993; Wang; Cui; Zhu, 2016; Ou et al., 2016]. Inspirado por estes trabalhos, o *SGCC* propõe uma etapa opcional para geração de novos vetores de características para os elementos do conjunto de entrada.

Após a formulação da configuração inicial, as hiperarestas de agrupamento contém uma representação contextual dos elementos inseridos em seu respectivo grupo. Estas hiperarestas podem ser exploradas de maneira similar à apresentada na Equação 4.21 para criação de vetores de características com c dimensões. Neste cenário, seja $\mathbf{X}$ uma matriz de características de tamanho $|C| \times |S|$ e $\mathbf{x}_i = \{x(o_i, \mathcal{S}_1), \ldots, x(o_i, \mathcal{S}_c)\}$ a linha da matriz $\mathbf{X}$ referente ao elemento o_i, a *Matriz de Representação Baseada no Hipergrafo* pode ser descrita como:

$$x(o_i, S_j) = e_i \cdot \mathbf{h}_j \tag{4.24}$$

5. Agrupamento Auto-Supervisionado

Os agrupamentos de alta confiabilidade obtidos em etapas anteriores são utilizados como conjunto de treinamento para um modelo de rede neural baseada em grafos. Dessa forma, o modelo obtido é utilizado para classificar as demais amostras, selecionando os agrupamentos iniciais aos quais serão inseridas. Conforme mencionado na Seção 4.3, modelos de *GCN* recebem uma matriz de características e um grafo e retornam uma nova representação de probabilidade de pertencimento entre c classes para cada um dos elementos, como definido na Equação 4.15. Com a aplicação de uma função *softmax*, estas representações podem ser usadas para tarefas de classificação.

Neste cenário, duas opções podem ser utilizadas para os vetores de características $\mathcal{X}$, as características originais do conjunto ou as representações baseadas no hipergrafo, descritas pela Equação 4.24. De maneira análoga, a matriz de adjacências $\mathbf{A}$ pode ser construída a partir de um grafo de vizinhos recíprocos, conforme definido na Equação 3.3, com vizinhança de tamanho k, baseado nas listas ranqueadas obtidas do método de *manifold learning*, ou recebidas como entrada do algoritmo em casos específicos como o de conjuntos de dados de redes de citação.

Por fim, os agrupamentos contidos em $\mathcal{S}$ formam o conjunto de rótulos $\mathcal{Y}$, o qual não possui rótulos para todos os elementos do conjunto de dados, conforme utilizado no cenário de aprendizado semi-supervisionado. O treinamento é realizado com a utilização do gradiente descendente baseado no erro de entropia cruzada dos rótulos $\mathcal{Y}$. Após o treinamento do modelo, uma nova inferência é realizada para obtenção da configuração final de agrupamentos do conjunto de entrada $\mathcal{C}$.

Com objetivo de sumarizar os parâmetros apresentados no decorrer desta seção, a Tabela 2 contém uma descrição dos quatro parâmetros recebidos como entrada pelo

Tabela 2 – Descrição de parâmetros do método *SGCC*.

Parâmetro	**Descrição**
c	Número esperado de agrupamentos de saída. Este parâmetro define em quantos grupos o conjunto de entrada será dividido e, consequentemente, quantos elementos serão selecionados como líderes durante as etapas de processamento.
k	Vizinhança recíproca explorada pelo método *LHRR* e utilizada para criação do grafo recíproco baseado as listas ranqueadas obtidas a partir do mesmo.
t	Número de iterações para realização do *LHRR*.
p	Parcela do conjunto de entrada separado em grupos para o posterior treinamento da GCN, entre 0.1 e 0.9.

SGCC. São eles: *(i)* c número esperado de agrupamentos, *(ii)* k vizinhança explorada, *(iii)* t iterações realizadas do método de *manifold learning* e *(iv)* p parcela do conjunto aglomerada para treinamento da *GCN*. Ao fim do processo, o *Self-Supervised Graph Convolutional Clustering* obtém uma separação auto-supervisionada do conjunto de dados.

5 Avaliação Experimental

A avaliação experimental dos métodos propostos neste trabalho é apresentada neste capítulo. A Seção 5.1 descreve os conjuntos de dados utilizados. A Seção 5.2 discute o protocolo experimental aplicado e os parâmetros utilizados. A Seção 5.3 apresenta os resultados obtidos na avaliação quantitativa dos métodos propostos. A Seção 5.4 apresenta uma análise visual e mais qualitativa dos resultados.

5.1 Conjuntos de Dados

Durante a avaliação experimental, três grupos de conjuntos de dados foram utilizados para explorar diferentes aspectos das metodologias propostas. A Tabela 3 apresenta as informações de quatro conjuntos de dados de imagens com tamanhos variando de 1.360 até 11.788 imagens de temas diversos. Estes conjuntos são frequentemente utilizados para tarefas de recuperação de imagens. Para aplicação dos métodos de agrupamento, as imagens são representadas por vetores de características extraídos por descritores. De acordo com o conjunto de dados, foram considerados descritores tradicionais (baseados em propriedades como forma), ou utilizando estratégias de aprendizado de máquina, como redes neurais convolucionais treinadas em cenários de transferência de aprendizado.

Tabela 3 – Conjuntos de dados de imagens utilizados para análise experimental.

Base de dados	Tamanho	Tipo	Descrição Geral
MPEG-7 [Latecki; Lakamper; Eckhardt, 2000]	1.400	Formas	Conjunto de dados composto de 70 classes de diferentes formas, sendo consideradas diferentes posições e orientações.
Flowers [Nilsback; Zisserman, 2006]	1.360	Flores	Conjunto Composto de 17 espécies de flores com 80 imagens de cada espécie apresentando variações de luz e posição.
Corel5k [Liu; Yang, 2013]	5.000	Objetos / Cenas	Composto de 50 categorias com 100 imagens cada, incluindo conteúdos de cena diversicados como fogos de artifícios, imagens microscópicas, árvores, etc.
CUB200 [Wah et al., 2011]	11.788	Pássaros	Conjunto de 200 espécies de pássaros em diferentes posições e ambientações.

O segundo grupo contém pequenos conjuntos de dados com representação em duas dimensões, utilizados para avaliação visual sem a necessidade da aplicação de técnicas de

redução de dimensionalidade. A Tabela 4 apresenta as informações sobre os três conjuntos de dados selecionados, os quais apresentam formas e número de agrupamentos variados.

Tabela 4 – Conjuntos de dados utilizadas para análise visual.

Base de dados	Tamanho	Descrição
Spirals [Fränti; Sieranoja, 2018]	312	Base de dados composta por 3 classes, contendo 104 pontos cada.
Jain's Toys [Fränti; Sieranoja, 2018]	373	Base de dados clássica para a área de agrupamento, contém 2 classes e um total de 373 pontos de 2 dimensões.
Two-Circles [Pedregosa et al., 2011]	500	Base de dados gerada sinteticamente com auxílio do framework python *scikit-learn*. É formado por 500 pontos, distribuídos em 2 classes que formam círculos concêntricos.

O terceiro grupo contém conjuntos de dados de redes de citação, usados frequentemente em tarefas de aprendizado semi-supervisionado e agrupamento. A Tabela 5 descreve as três redes de citação escolhidas para este trabalho, as quais variam de 2.708 a 19.717 elementos, com grafos de 4.732 a 44.338 arestas e vetores binários de características variando de 500 até 3.703 dimensões.

Tabela 5 – Conjunto de dados de redes de citação utilizadas para avaliação experimental.

Base de dados	Tamanho	Tipo	Descrição Geral
Cora [Sen et al., 2008]	2.708	Publicações Científicas	Conjunto composto por 7 classes de diferentes publicações, contendo um grafo de citação com 5.429 arestas.
CiteSeer [Sen et al., 2008]	3,312	Publicações Científicas	Rede de citação com trabalhos de 6 categorias, representado em um grafo com 4,732 arestas.
PubMed [Sen et al., 2008]	19.717	Publicações Científicas	Composto por 3 categorias de publicações do banco de dados PubMed, contém um grafo com 44.338 arestas.

5.2 Protocolo Experimental e Definição de Parâmetros

Os métodos *C-ReckNN* e *SGCC* foram comparados com métodos clássicos e recentes disponíveis na literatura, com datas de publicação variando de 1957 a 2020. Esta seleção visou encontrar métodos relevantes que utilizassem diferentes abordagens para separação dos elementos. Todos os algoritmos selecionados para comparação foram utilizados com valores de parâmetros pré-definidos pelos autores e, quando possível, o número exato de classes foi definido como o número desejado de agrupamentos. Os métodos que reportam

resultados com abordagem probabilística foram executados por 10 repetições e a média dos resultados obtidos é reportado nas tabelas de comparação.

Para a execução do *C-ReckNN* foi utilizado $c_k = 3$ para todos os experimentos. O tamanho da vizinhança explorada foi de $k = 15$ para o conjunto *MPEG-7*, devido ao tamanho reduzido de suas classes, e de $k = 50$ para os demais conjuntos.

O método *SGCC* foi explorado com diversas configurações para aplicação em uma grande variedade de conjuntos de dados. Primeiramente, um experimento foi realizado para encontrar um valor padrão para o parâmetro p. A Figura 12 apresenta os resultados deste experimento, onde o conjunto de dados *Corel5K* foi separado utilizando um valor fixo de $k = 50$ e variando o valor de $p \in [0.1, 0.9]$ em dois cenários com valores de $t = 1$ e $t = 2$. O valor $p = 0.5$ é representando pela linha pontilhada em todos os experimentos. É possível notar que utilizando $t = 1$, as métricas avaliadas sobem conforme o valor de p aumenta. Porém, na execução com $t = 2$, retratada na coluna da direita, as métricas decaem conforme o valor de p aumenta. Visando avaliar todas as possíveis configurações do método proposto, o parâmetro $p = 0.5$ é escolhido como um equilíbrio entre os resultados obtidos para cada valor do parâmetro t.

Dois cenários de avaliação foram explorados: *(i) utilizando $k = 50$ em todos os experimentos*, onde este valor é escolhido como uma indicação com resultados competitivos para todos os conjuntos, e *(ii) variando k com valores entre 10 e 100*, reportando os melhores resultados encontrados. Estes dois cenários são testados em quatro combinações diferentes de abordagem do método, variando a equação de aglomeração (função $nc(o_i)$), utilizada durante a etapa de aglomeração ordenada, e avaliando a utilização das características originais do conjunto em comparação com as representações baseadas no hipergrafo (apresentada na Equação 4.24). Em ambos os cenários, o parâmetro t foi avaliado com valores 1 e 2.

Para a etapa de treinamento da *GCN*, foram utilizados três modelos: *(i)* a *Graph Convolutional Network (GCN)* original proposta por [Kipf; Welling, 2017], *(ii)* a *Simple Graph Convolution (SGC)* [Wu et al., 2019] e *(iii)* a *Approximate Personalized Propagation of Neural Predictions (APPNP)* [Klicpera; Bojchevski; Günnemann, 2019]. As três redes escolhidas foram treinadas em todos os cenários explorados na avaliação do *SGCC*.

Em todos os experimentos, os modelos de *GCN* utilizam 32 camadas ocultas e taxa de aprendizagem de 10^{-3}, a qual é reduzida pela metade a cada 100 épocas. Além disso, o modelo é treinado por 400 épocas para as características originais do conjunto e por 800 épocas para as características baseadas no hipergrafo, o aumento no número de épocas se deve ao fato das novas representações serem de menor dimensionalidade. Os valores reportados são a média e desvio padrão de 10 repetições e toda repetição possui um mecanismo de parada precoce, caso a época atinja um acerto de 100% nos rótulos de treinamento.

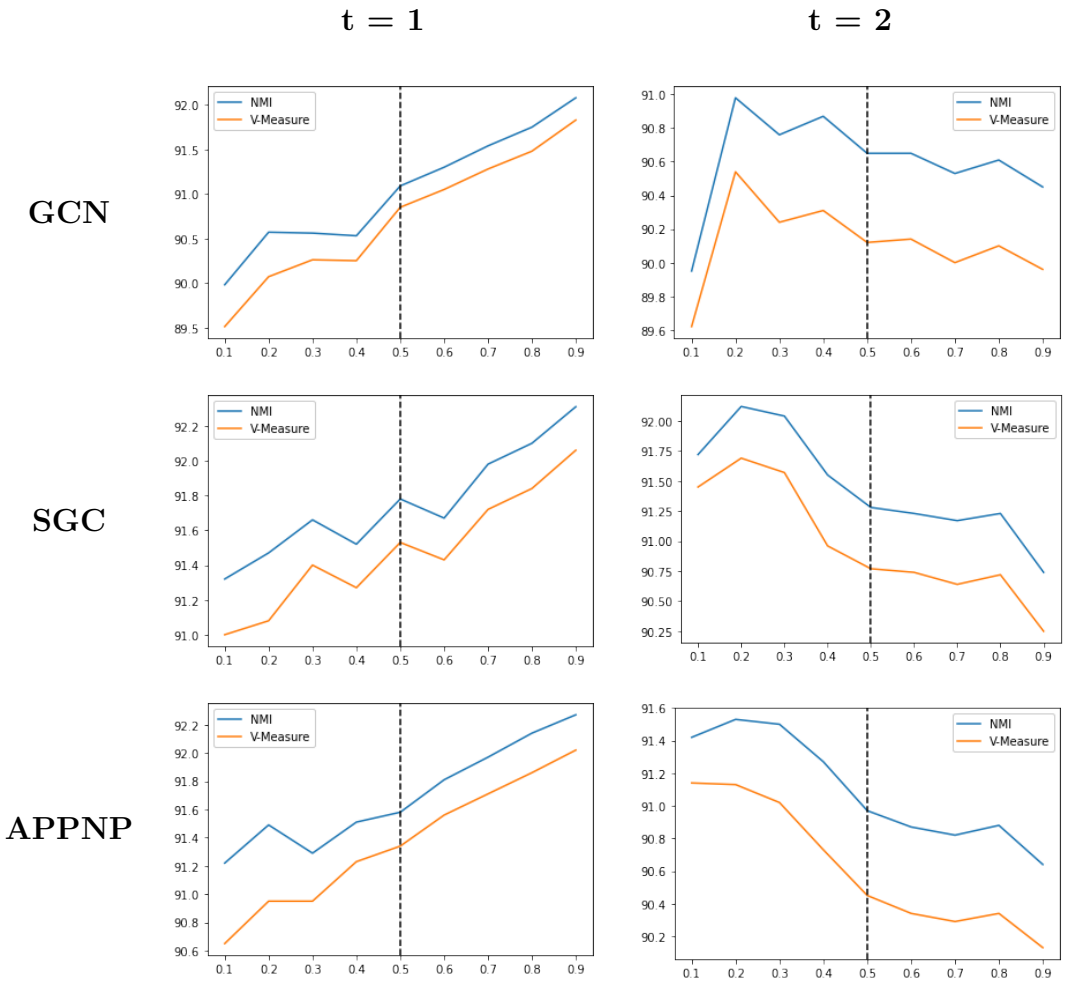


Figura 12 – Avaliação do impacto do parâmetro p no método *SGCC*. O experimento foi realizado utilizando o conjunto *Corel5K* e $k = 50$. Os resultados de *NMI* e *V-Measure* são apresentados para os três modelos de redes neurais baseadas em grafo (linhas) e com valores $t = 1$ e $t = 2$ (colunas).

Para avaliação dos agrupamentos gerados, foram utilizadas as métricas *Adjusted Rand Index (ARI)*, *Normalized Mutual Information (NMI)* e *V-Measure*. O *F-Measure* foi utilizado nos experimentos do *C-ReckNN* e, observando a tendência de trabalhos recentes, principalmente baseados em *GCNs*, a métrica de *Acurácia (ACC)* foi utilizada nos experimentos do *SGCC*. Todas as métricas listadas apresentam valores entre 0 e 100 onde, quanto maior o resultado, melhor o agrupamento obtido.

5.3 Resultados

Nesta seção são apresentados os resultados quantitativos obtidos pelos métodos propostos. A Seção 5.3.1 apresenta os resultados e comparações realizadas com o *C-ReckNN* e a Seção 5.3.2 apresenta os experimentos realizados para avaliação do método *SGCC*.

5.3.1 C-RecKNN

O *C-ReckNN* foi comparado com os métodos *K-means* [MacQueen, 1967], *Agglomerative* [Murtagh, 1983], *Affinity Propagation* [Frey; Dueck, 2007] e *FINCH* [Sarfraz; Sharma; Stiefelhagen, 2019]. Durante os experimentos, foram utilizados os conjuntos de dados *MPEG-7*, com descritores de forma *Aspect Shape Context (ASC)* [Ling; Yang; Latecki, 2010] e *Contour Feature Descriptor (CFD)* [Pedronette; Torres, 2010], *Flowers* e *Corel5K*, ambos com características extraídas pelo descritor de cores *Auto Color Correlograms (ACC)* [Huang et al., 1997] e pela rede neural *ResNet-152* [He et al., 2015].

A avaliação foi separada em dois experimentos: *(i)* As distâncias pré-computadas para o conjunto de dados *MPEG-7* foram utilizadas em métodos de agrupamento que aceitam matrizes de distância como entrada; *(ii)* Os vetores de características dos conjuntos *Flowers* e *COREL5K* foram utilizados em métodos de agrupamento que aceitam somente vetores de características como entrada. A Tabela 6 apresenta os resultados obtidos para o conjunto de dados *MPEG-7* e a Tabela 7 apresenta os resultados para os conjuntos *Flowers* e *COREL5K*.

Tabela 6 – Comparação entre o *C-ReckNN* e métodos da literatura para o conjunto *MPEG-7*.

Conjunto	Desc.	Método	F-Meas.	ARI	NMI	V-Meas.
MPEG-7	CFD	Agglom.	51.31	50.42	90.43	86.76
		FINCH	47.45	46.50	87.07	83.72
		Aff. Prop.	03.53	00.89	66.32	19.24
		C-ReckNN	**91.04**	**90.91**	**96.99**	**96.76**
	ASC	Agglom.	60.60	59.94	91.43	88.81
		FINCH	63.47	62.86	91.52	87.52
		Aff. Prop.	06.22	03.74	61.03	35.82
		C-ReckNN	**82.69**	**82.43**	**96.60**	**95.30**

É possível verificar que o *C-ReckNN* obteve resultados equiparáveis ou superiores aos outros métodos analisados em todos os testes. Os resultados obtidos foram superiores em todas as métricas para o conjunto *MPEG-7* e nas métricas *ARI* e *F-measure* para os outros conjuntos de dados. Isso demonstra que o método proposto é versátil e atinge bons resultados na tarefa de agrupamento.

É válido ressaltar que os mesmos parâmetros foram utilizados para os conjuntos *Flowers* e *COREL5K*, mesmo que a diferença de tamanho entre os conjuntos seja significativa. Este fato demonstra que o tamanho da vizinhança recíproca analisada não precisa ser aumentado conforme o número de objetos no conjunto de dados aumenta.

Tabela 7 – Comparação entre o *C-ReckNN* e métodos da literatura para os conjuntos *Flowers* e *COREL5K*.

Conjunto	Desc.	Método	F-Meas.	ARI	NMI	V-Meas.
Flowers	ACC	K-means	17.80	12.50	28.44	28.22
		Agglom.	14.58	07.44	25.19	23.20
		FINCH	10.95	00.31	33.66	20.40
		Aff. Prop.	08.17	06.28	**50.08**	**38.76**
		C-ReckNN	**18.90**	**13.55**	29.12	28.63
	ResNet	K-Means	62.05	59.67	73.75	73.56
		Agglom.	43.80	39.41	66.61	62.35
		FINCH	21.66	13.06	65.30	51.45
		Aff. Prop.	29.73	28.08	**83.35**	65.90
		C-ReckNN	**65.82**	**63.63**	77.27	**76.84**
Corel5k	ACC	K-Means	22.06	20.41	47.39	47.08
		Agglom.	14.62	12.15	42.37	38.95
		FINCH	08.31	04.90	48.56	36.25
		Aff. Prop.	13.35	12.68	**63.82**	**53.59**
		C-ReckNN	**24.69**	**23.20**	49.87	49.31
	ResNet	K-Means	77.35	76.87	89.56	89.03
		Agglom.	47.65	46.25	91.03	90.65
		FINCH	40.98	39.16	90.06	81.31
		Aff. Prop.	32.69	32.17	**93.04**	77.53
		C-ReckNN	**83.00**	**82.66**	91.36	**90.73**

5.3.2 SGCC

A avaliação experimental do método *SGCC* foi realizada em duas etapas. Primeiramente, a Seção 5.3.2.1 apresenta os resultados obtidos durante a exploração das diversas configurações de parâmetros disponíveis. Após a obtenção dos melhores resultados, a Seção 5.3.2.2 apresenta a comparação do método *SGCC* com métodos clássicos e recentes da literatura.

5.3.2.1 Avaliação de configurações do método

O *SGCC* foi inicialmente avaliado em dois experimentos, realizados em sete conjuntos de dados distintos. Foram utilizados quatro conjuntos de imagem, com tamanhos variando de 1.360 a 11.788 elementos: *MPEG-7* com descritor de forma *CFD* [Pedronette; Torres, 2010], *Flowers*, *Corel5K* e *CUB200*, todos utilizando características extraídas pela rede neural *ResNet-152* [He et al., 2015] e três redes de citação: *Cora*, *CiteSeer* e *PubMed*, os quais possuem vetores de características binários que representam a ocorrência de palavras e possuem grafos de citação próprios. Para obter as listas ranqueadas de entrada, a distância Euclidiana foi utilizada para os conjuntos *Flowers*, *Corel5K* e *CUB200*. Para

o conjunto *MPEG-7*, a matriz de distância fornecida pelo *CFD* é utilizada e a distância *Jaccard* é aplicada aos vetores binários dos conjuntos de citação utilizados.

Tabela 8 – Resultados obtidos para o método *SGCC*, utilizando $k = 50$, características originais do conjunto de dados e a Equação 4.20 (média ponderada de pertencimento) como função $nc(o_i)$.

Dataset	Network	t	NMI	V-Measure	ACC
Corel5K	GCN	1	91.34 ± 00.16	91.10 ± 00.15	88.45 ± 00.12
	SGC	1	$\mathbf{91.74 \pm 00.06}$	$\mathbf{91.50 \pm 00.06}$	$\mathbf{88.74 \pm 00.05}$
	APPNP	1	91.64 ± 00.15	90.47 ± 00.15	88.72 ± 00.15
CUB200	GCN	2	68.93 ± 00.42	68.22 ± 00.45	47.70 ± 00.46
	SGC	2	$\mathbf{69.97 \pm 00.03}$	$\mathbf{69.33 \pm 00.03}$	$\mathbf{48.37 \pm 00.04}$
	APPNP	2	69.68 ± 00.13	68.93 ± 00.12	47.77 ± 00.30
Flowers	GCN	2	80.48 ± 00.57	80.18 ± 00.59	82.61 ± 00.61
	SGC	2	$\mathbf{81.27 \pm 00.07}$	$\mathbf{81.01 \pm 00.07}$	$\mathbf{83.49 \pm 00.09}$
	APPNP	2	80.79 ± 00.25	80.51 ± 00.25	82.85 ± 00.23
MPEG-7	GCN	-	-	-	-
	SGC	2	$\mathbf{89.74 \pm 00.34}$	$\mathbf{87.64 \pm 00.35}$	$\mathbf{74.06 \pm 00.77}$
	APPNP	1	16.85 ± 33.70	06.83 ± 13.65	02.02 ± 01.19
Cora	GCN	1	31.32 ± 00.40	30.71 ± 00.40	45.86 ± 00.21
	SGC	1	35.94 ± 00.11	35.49 ± 00.11	$\mathbf{49.65 \pm 00.06}$
	APPNP	1	$\mathbf{37.12 \pm 00.23}$	$\mathbf{36.52 \pm 00.24}$	49.18 ± 00.27
Citeseer	GCN	2	28.80 ± 00.21	28.51 ± 00.21	54.20 ± 00.17
	SGC	2	$\mathbf{30.86 \pm 00.07}$	$\mathbf{30.47 \pm 00.07}$	$\mathbf{55.86 \pm 00.05}$
	APPNP	2	30.55 ± 00.17	30.24 ± 00.17	55.68 ± 00.20
PubMed	GCN	1	18.32 ± 00.08	$\mathbf{18.04 \pm 00.07}$	$\mathbf{56.45 \pm 00.07}$
	SGC	1	$\mathbf{31.19 \pm 00.40}$	17.77 ± 00.49	49.76 ± 00.29
	APPNP	1	18.17 ± 00.08	17.69 ± 00.10	56.27 ± 00.38

Durante o treinamento dos modelos de *GCN*, o grafo utilizado para os experimentos com conjuntos de imagem foi criado com base nas listas ranqueadas obtidas do *LHRR*. Quando avaliados as redes de citação, o grafo original do conjunto de dados foi utilizado.

Os experimentos realizados buscaram comparar os resultados obtidos ao utilizar as características originais, extraídas pelos descritores citados acima, e as características baseadas no hipergrafo, formuladas pelo método proposto. Quatro cenários foram considerados para cada experimento, alternando entre as abordagens da função $nc(o_i)$ e testando k com valor fixo de 50 e alternando no intervalo [15..100]. As Tabelas 8 e 9 utilizam as equações 4.20 e 4.21, respectivamente, e um valor fixo de $k = 50$.

Neste primeiro cenário, o modelo *SGC* alcançou os melhores resultados na maioria dos experimentos. Os valores obtidos pelos três modelos foram próximos para quase todos os conjuntos, variando cerca de 2 pontos em todas as métricas, exceto nos conjuntos *MPEG-7*, que utilizou a matriz de distância obtida pelo descritor como matriz de características para

Tabela 9 – Resultados obtidos para o método *SGCC*, utilizando $k = 50$, características originais do conjunto de dados e a Equação 4.21 (produto interno das hiperarestas) como função $nc(o_i)$

Dataset	Network	t	NMI	V-Measure	ACC
Corel5K	GCN	2	92.19 ± 00.14	91.96 ± 00.14	90.69 ± 00.16
	SGC	2	$\mathbf{92.71 \pm 00.04}$	$\mathbf{92.47 \pm 00.04}$	$\mathbf{91.11 \pm 00.03}$
	APPNP	2	92.46 ± 00.06	92.22 ± 00.06	90.95 ± 00.08
CUB200	GCN	2	72.88 ± 00.17	67.38 ± 00.15	40.48 ± 00.10
	SGC	2	$\mathbf{73.83 \pm 00.02}$	$\mathbf{68.25 \pm 00.02}$	$\mathbf{41.12 \pm 00.02}$
	APPNP	2	73.50 ± 00.12	67.96 ± 00.12	40.84 ± 00.10
Flowers	GCN	2	82.14 ± 00.36	80.82 ± 00.36	81.60 ± 00.24
	SGC	2	$\mathbf{82.84 \pm 00.17}$	$\mathbf{81.56 \pm 00.17}$	$\mathbf{82.07 \pm 00.10}$
	APPNP	2	82.35 ± 00.25	81.03 ± 00.23	81.59 ± 00.12
MPEG-7	GCN	1	07.80 ± 23.39	03.67 ± 11.02	01.86 ± 01.29
	SGC	2	$\mathbf{89.70 \pm 00.30}$	$\mathbf{77.01 \pm 00.33}$	$\mathbf{37.74 \pm 00.43}$
	APPNP	1	40.35 ± 40.41	16.79 ± 17.51	03.51 ± 02.21
Cora	GCN	1	33.12 ± 00.97	19.51 ± 00.33	39.05 ± 00.15
	SGC	2	33.64 ± 00.14	26.32 ± 00.10	38.56 ± 00.08
	APPNP	2	$\mathbf{35.86 \pm 00.27}$	$\mathbf{29.94 \pm 00.18}$	$\mathbf{40.30 \pm 00.30}$
Citeseer	GCN	2	34.65 ± 00.16	33.28 ± 00.15	57.33 ± 00.13
	SGC	2	35.90 ± 00.11	34.37 ± 00.11	58.02 ± 00.11
	APPNP	2	$\mathbf{36.51 \pm 00.16}$	$\mathbf{35.15 \pm 00.15}$	$\mathbf{58.88 \pm 00.20}$
PubMed	GCN	1	20.04 ± 00.06	19.91 ± 00.06	53.37 ± 00.04
	SGC	1	$\mathbf{23.63 \pm 00.14}$	$\mathbf{20.80 \pm 00.13}$	53.32 ± 00.16
	APPNP	1	20.57 ± 00.11	20.32 ± 00.09	$\mathbf{54.08 \pm 00.14}$

o treinamento das redes neurais, e *PubMed*, que apresenta grande dificuldade de separação ao utilizar vetores de características binários e somente conter 3 agrupamentos, mesmo contendo mais de 19.000 elementos.

Também é possível notar que a utilização da equação de *produto interno das hiperarestas* (Tabela 9) para realização das aglomerações iniciais piorou o resultado de acurácia em quase todos os conjuntos. Apesar de ter aumentado alguns resultados de *NMI* e ter obtido resultados melhores em todas as métricas para os conjuntos *Corel5K* e *CiteSeer*. Os resultados de *NMI* são decorrentes da maior permissividade da métrica em relação a alocações de elementos em conjuntos incorretos. Por fim, fica claro que a utilização das matrizes de distância como matrizes de características para o conjunto *MPEG-7* dificulta o aprendizado dos modelos, onde as redes neurais *GCN* e *APPNP* não conseguiram separar o conjunto, obtendo resultados muito inferiores ou nulos.

Em um segundo cenário, utilizando as características originais do conjunto e variando o valor de k no intervalo [15..100], a Tabela 10 apresenta os resultados utilizando a *média ponderada de pertencimento*, enquanto a Tabela 11 lista os valores obtidos com a

Tabela 10 – Resultados obtidos para o método *SGCC*, variando o parâmetro k em um intervalo [10..100], usando as características originais do conjunto de dados e a Equação 4.20 (média ponderada de pertencimento) como função $nc(o_i)$

Dataset	Network	k	t	NMI	V-Measure	ACC
Corel5K	GCN	95	2	91.89 ± 00.13	91.79 ± 00.13	90.86 ± 00.11
	SGC	70	2	$\mathbf{92.62 \pm 00.06}$	$\mathbf{92.44 \pm 00.06}$	90.80 ± 00.04
	APPNP	95	2	92.27 ± 00.12	92.16 ± 00.11	$\mathbf{91.19 \pm 00.12}$
CUB200	GCN	55	2	69.07 ± 00.13	68.21 ± 00.12	47.52 ± 00.16
	SGC	55	2	$\mathbf{69.97 \pm 00.02}$	$\mathbf{69.19 \pm 00.02}$	$\mathbf{48.38 \pm 00.02}$
	APPNP	50	2	69.68 ± 00.13	68.93 ± 00.12	47.77 ± 00.30
Flowers	GCN	45	2	80.98 ± 00.28	80.70 ± 00.31	82.68 ± 00.47
	SGC	50	2	$\mathbf{81.27 \pm 00.07}$	$\mathbf{81.01 \pm 00.07}$	$\mathbf{83.49 \pm 00.09}$
	APPNP	45	2	81.27 ± 00.24	80.99 ± 00.26	82.86 ± 00.33
MPEG-7	GCN	25	1	07.58 ± 22.75	02.32 ± 06.96	01.76 ± 00.99
	SGC	20	2	$\mathbf{96.45 \pm 00.15}$	$\mathbf{96.37 \pm 00.15}$	$\mathbf{94.56 \pm 00.16}$
	APPNP	75	1	32.39 ± 39.81	12.71 ± 20.09	04.82 ± 08.77
Cora	GCN	85	1	39.45 ± 00.34	38.97 ± 00.33	59.22 ± 00.22
	SGC	85	1	$\mathbf{45.02 \pm 00.15}$	$\mathbf{44.81 \pm 00.15}$	$\mathbf{62.96 \pm 00.09}$
	APPNP	65	1	44.58 ± 00.19	44.39 ± 00.18	62.46 ± 00.18
Citeseer	GCN	60	1	32.84 ± 00.24	32.64 ± 00.24	61.11 ± 00.20
	SGC	65	1	$\mathbf{35.50 \pm 00.08}$	$\mathbf{35.34 \pm 00.08}$	$\mathbf{62.95 \pm 00.07}$
	APPNP	65	1	35.42 ± 00.21	35.23 ± 00.21	62.73 ± 00.14
PubMed	GCN	40	1	26.31 ± 00.07	$\mathbf{25.37 \pm 00.07}$	$\mathbf{62.95 \pm 00.11}$
	SGC	45	1	$\mathbf{28.59 \pm 00.08}$	22.41 ± 00.07	52.15 ± 00.04
	APPNP	65	1	27.41 ± 00.39	25.12 ± 00.13	62.07 ± 00.41

utilização do *produto interno das hiperarestas*. Neste experimento, a maioria dos resultados supera os valores obtidos ao fixar o valor de $k = 50$, porém, cerca de metade dos conjuntos continuou alcançando os melhores resultados com valores de k próximos ao valor sugerido no experimento anterior. Isso demonstra a estabilidade do impacto obtido ao variar a vizinhança explorada pelo *LHRR* e pelo grafo de vizinhos recíprocos.

Assim como nos experimentos anteriores, a aplicação do *produto interno das hiperarestas* reduziu a maioria dos resultados de acurácia obtidos. Além disso, modelo *SGC* reportou os melhores resultados para a maioria dos experimentos, enquanto o modelo *APPNP* reportou os melhores resultados para todos os conjuntos de citação reportados na Tabela 11.

As Tabelas 12 e 13 avaliam um terceiro cenário, onde as características obtidas pelo método, conforme descrito na Equação 4.24, são utilizadas em conjunto com as duas abordagens da função $nc(o_i)$. Nesta nova configuração, os melhores resultados foram obtidos pelos modelos *GCN* e *APPNP*, os quais apresentaram maior capacidade de aprendizado com base nas características baseadas no hipergrafo. A rede *SGC*, por outro lado, apresentou dificuldades em obter bons resultados nestes experimentos, chegando

Tabela 11 – Resultados obtidos para o método *SGCC*, variando o parâmetro k em um intervalo [10..100], usando as características originais do conjunto de dados e a Equação 4.21 (produto interno das hiperarestas) como função $nc(o_i)$

Dataset	Network	k	t	NMI	V-Measure	ACC
Corel5K	GCN	50	2	92.19 ± 00.14	91.96 ± 00.14	90.69 ± 00.16
	SGC	50	2	$\mathbf{92.71 \pm 00.04}$	$\mathbf{92.47 \pm 00.04}$	$\mathbf{91.11 \pm 00.03}$
	APPNP	50	2	92.46 ± 00.06	92.22 ± 00.06	90.95 ± 00.08
CUB200	GCN	25	2	69.73 ± 00.26	67.84 ± 00.26	46.24 ± 00.24
	SGC	30	2	$\mathbf{71.60 \pm 00.02}$	$\mathbf{68.98 \pm 00.02}$	$\mathbf{47.22 \pm 00.02}$
	APPNP	30	2	71.60 ± 00.16	68.86 ± 00.14	46.58 ± 00.16
Flowers	GCN	50	2	82.14 ± 00.36	80.82 ± 00.36	81.60 ± 00.24
	SGC	50	2	$\mathbf{82.84 \pm 00.17}$	$\mathbf{81.56 \pm 00.17}$	$\mathbf{82.07 \pm 00.10}$
	APPNP	50	2	82.35 ± 00.25	81.03 ± 00.23	81.59 ± 00.12
MPEG-7	GCN	85	1	45.70 ± 37.46	21.05 ± 18.90	04.84 ± 03.21
	SGC	20	2	$\mathbf{96.38 \pm 00.08}$	$\mathbf{96.30 \pm 00.08}$	$\mathbf{94.39 \pm 00.13}$
	APPNP	55	1	63.23 ± 31.69	29.66 ± 18.13	06.21 ± 04.03
Cora	GCN	95	2	37.90 ± 00.31	36.88 ± 00.28	59.42 ± 00.22
	SGC	85	1	43.12 ± 00.13	40.99 ± 00.13	60.88 ± 00.08
	APPNP	95	1	$\mathbf{43.68 \pm 00.29}$	$\mathbf{42.42 \pm 00.30}$	$\mathbf{62.79 \pm 00.24}$
Citeseer	GCN	55	2	34.69 ± 00.13	34.11 ± 00.13	58.93 ± 00.22
	SGC	55	2	35.76 ± 00.09	35.05 ± 00.09	59.14 ± 00.07
	APPNP	55	2	$\mathbf{36.55 \pm 00.14}$	$\mathbf{35.94 \pm 00.13}$	$\mathbf{60.10 \pm 00.14}$
PubMed	GCN	45	2	33.10 ± 00.16	32.20 ± 00.16	68.77 ± 00.05
	SGC	75	1	31.33 ± 00.80	24.75 ± 00.61	53.86 ± 00.06
	APPNP	45	2	$\mathbf{33.84 \pm 00.05}$	$\mathbf{32.91 \pm 00.05}$	$\mathbf{69.51 \pm 00.02}$

a atingir resultados nulos em um dos cenários avaliados. Este comportamento deve-se provavelmente a características arquiteturais da *SGC*, que se destaca por ser uma rede simplificada.

De maneira geral, todas as métricas alcançaram valores inferiores ao substituir as características originais, porém, é importante ressaltar que a dimensão dos vetores é reduzido drasticamente neste novo cenário. Dois exemplos podem ser destacados nos conjuntos *Flowers*, que diminui seus vetores de 2048 posições para 17, e *CiteSeer*, que tem seus vetores binários de 3703 posições reduzidos para somente 6 dimensões.

Em uma última configuração avaliada, as Tabelas 14 e 15 utilizam as características baseadas no hipergrafo e variam o parâmetro k no intervalo [10..100]. Os melhores resultados encontrados ao variar o parâmetro k são comparáveis com os resultados obtidos pelas características originais do conjunto. Apesar disso, o modelo *SGC* novamente apresentou valores inferiores ao reportado pelos outros modelos.

O melhor valor de k também diminuiu para alguns conjuntos, como a *Flowers* e a *PubMed*, novamente indicando permuta entre eficácia (qualidade) e eficiência (tempo) ao utilizar o conjunto de características de menor dimensionalidade.

Tabela 12 – Resultados obtidos para o método *SGCC*, utilizando $k = 50$, características baseadas no hipergrafo e a Equação 4.20 (média ponderada de pertencimento) como função $nc(o_i)$

Dataset	Network	t	NMI	V-Measure	ACC
Corel5K	GCN	1	89.38 ± 00.12	88.96 ± 00.12	$\mathbf{86.83 \pm 00.14}$
	SGC	1	83.13 ± 00.68	74.28 ± 00.94	54.12 ± 01.42
	APPNP	1	$\mathbf{91.08 \pm 00.21}$	$\mathbf{90.71 \pm 00.25}$	87.54 ± 00.72
CUB200	GCN	2	68.72 ± 00.14	64.40 ± 00.13	$\mathbf{43.55 \pm 00.17}$
	SGC	1	$\mathbf{73.07 \pm 00.31}$	56.22 ± 00.30	19.52 ± 00.32
	APPNP	2	69.85 ± 00.14	$\mathbf{66.21 \pm 00.09}$	42.56 ± 00.22
Flowers	GCN	2	75.98 ± 00.57	74.91 ± 00.45	77.04 ± 00.33
	SGC	1	72.19 ± 00.95	65.22 ± 00.84	55.62 ± 00.97
	APPNP	2	$\mathbf{78.91 \pm 00.50}$	$\mathbf{78.25 \pm 00.33}$	$\mathbf{79.74 \pm 00.96}$
MPEG-7	GCN	2	89.88 ± 00.26	$\mathbf{86.62 \pm 00.30}$	$\mathbf{69.15 \pm 01.06}$
	SGC	2	89.18 ± 00.26	73.63 ± 00.43	30.64 ± 01.03
	APPNP	2	$\mathbf{89.95 \pm 00.19}$	84.61 ± 00.12	59.41 ± 00.44
Cora	GCN	1	34.99 ± 00.59	34.82 ± 00.57	50.05 ± 00.53
	SGC	2	24.99 ± 02.35	19.49 ± 01.95	36.09 ± 02.13
	APPNP	1	$\mathbf{38.40 \pm 00.76}$	$\mathbf{38.09 \pm 00.82}$	$\mathbf{52.25 \pm 01.19}$
Citeseer	GCN	2	32.85 ± 00.30	31.33 ± 00.33	54.09 ± 00.86
	SGC	1	26.79 ± 01.09	24.20 ± 01.24	46.49 ± 02.10
	APPNP	2	$\mathbf{34.71 \pm 00.26}$	$\mathbf{32.36 \pm 00.22}$	$\mathbf{54.77 \pm 00.41}$
PubMed	GCN	1	18.59 ± 00.11	$\mathbf{18.31 \pm 00.09}$	$\mathbf{57.03 \pm 00.09}$
	SGC	1	$\mathbf{21.53 \pm 06.32}$	7.69 ± 07.22	44.28 ± 04.39
	APPNP	1	18.23 ± 00.28	17.97 ± 00.20	56.71 ± 00.32

O parâmetro t, variado no intervalo [1..2] durante estes experimentos, demonstrou melhores resultados utilizando $t = 2$ quando aplicada o *produto interno das hiperarestas* foi aplicado na obtenção dos agrupamentos iniciais.

5.3.2.2 Comparação com métodos da literatura

Em um experimento final, os melhores resultados obtidos em cada conjunto avaliado foram comparados com métodos clássicos e recentes da literatura. Durante esta comparação, foram avaliados os métodos *K-Means* [MacQueen, 1967], *Agglomerative* [Murtagh, 1983], *HDBSCAN* [McInnes; Healy; Astels, 2017], *FINCH* [Sarfraz; Sharma; Stiefelhagen, 2019], *C-ReckNN* [Lopes. et al., 2020], *SDCN* [Bo et al., 2020] e *MinCutPool* [Bianchi; Grattarola; Alippi, 2020].

A Tabela 16 apresenta a comparação dos resultados de *NMI*, *V-Measure* e *ACC* para os conjuntos de imagem avaliados. Os melhores valores estão destacados em negrito e todos os valores em um mesmo intervalo de desvio padrão são destacados, quando necessário.

Tabela 13 – Resultados obtidos para o método *SGCC*, utilizando $k = 50$, características baseadas no hipergrafo e a Equação 4.21 (produto interno das hiperarestas) como função $nc(o_i)$

Dataset	Network	t	NMI	V-Measure	ACC
Corel5K	GCN	1	**89.67 ± 00.14**	**88.57 ± 00.17**	83.65 ± 00.11
	SGC	1	84.48 ± 00.47	75.85 ± 01.15	55.65 ± 01.79
	APPNP	2	89.23 ± 00.10	88.05 ± 00.13	**86.38 ± 00.20**
CUB200	GCN	2	71.39 ± 00.26	63.54 ± 00.18	**37.30 ± 00.19**
	SGC	2	70.60 ± 00.55	51.46 ± 01.07	17.19 ± 00.31
	APPNP	2	**73.40 ± 00.16**	**65.30 ± 00.16**	36.01 ± 00.31
Flowers	GCN	2	76.66 ± 00.25	74.08 ± 00.32	75.34 ± 00.33
	SGC	1	**80.38 ± 03.44**	66.54 ± 03.11	48.44 ± 01.32
	APPNP	2	79.90 ± 00.27	**78.02 ± 00.29**	**78.64 ± 00.86**
MPEG-7	GCN	2	**90.41 ± 00.17**	**77.03 ± 00.17**	**36.56 ± 00.12**
	SGC	2	89.35 ± 00.47	69.47 ± 00.42	23.07 ± 00.44
	APPNP	2	90.24 ± 00.30	76.19 ± 00.24	34.92 ± 00.31
Cora	GCN	1	40.89 ± 01.00	**16.30 ± 00.54**	37.13 ± 00.34
	SGC	-	-	-	-
	APPNP	1	**43.40 ± 00.67**	13.66 ± 00.66	**36.09 ± 00.28**
Citeseer	GCN	2	34.48 ± 00.69	31.28 ± 00.51	53.73 ± 00.50
	SGC	2	23.10 ± 03.30	10.82 ± 02.05	29.98 ± 02.41
	APPNP	2	**36.15 ± 00.24**	**32.00 ± 00.23**	**54.25 ± 00.32**
PubMed	GCN	1	20.85 ± 00.08	20.60 ± 00.06	53.37 ± 00.12
	SGC	1	**24.00 ± 02.28**	20.15 ± 01.79	52.19 ± 01.49
	APPNP	1	21.00 ± 00.12	**20.73 ± 00.11**	**53.44 ± 00.23**

O método *SGCC* obteve os melhores resultados em todos os conjuntos de dados, principalmente durante a utilização do modelo *SGC*. Os métodos clássicos *K-means* e *Agglomerative* se destacaram como melhores resultados dentre os demais comparados em 3 dos 4 conjuntos, seguidos pelos métodos *SDCN* e *MinCutPool*, algoritmos recentes que também utilizam redes neurais baseadas em grafo.

Além disso, todos os modelos de redes neurais baseadas em grafo utilizadas pelo *SGCC*, utilizando configurações diferentes, chegaram a valores muito próximos para as três métricas em todos os conjuntos avaliados, reforçando a versatilidade do método em prover informações suficientes para o treinamento dos diferentes modelos. O *C-ReckNN*, o qual foi comparado nas métricas de *NMI* e *V-Measure* para os conjuntos *Corel5K*, *Flowers* e *MPEG-7*, apresentou resultados comparáveis no conjunto *MPEG-7*, obtendo o maior valor determinístico de *NMI*.

Por fim, a Tabela 17 apresenta a comparação realizada nas três redes de citações avaliadas. O *SGCC* obteve os melhores resultados em todos os conjuntos, para todas as métricas avaliadas. O modelo que se destacou durante a avaliação em redes de citação

Tabela 14 – Resultados obtidos para o método *SGCC*, variando o parâmetro k em um intervalo [10..100], usando as características baseadas no hipergrafo e a Equação 4.20 (média ponderada de pertencimento) como função $nc(o_i)$

Dataset	Network	k	t	NMI	V-Measure	ACC
Corel5K	GCN	95	2	90.87 ± 00.07	90.71 ± 00.07	90.02 ± 00.07
	SGC	60	1	85.80 ± 00.52	78.17 ± 00.73	56.69 ± 01.29
	APPNP	95	2	$\mathbf{91.40 \pm 00.11}$	$\mathbf{91.14 \pm 00.14}$	$\mathbf{90.14 \pm 00.53}$
CUB200	GCN	40	2	67.49 ± 00.19	64.94 ± 00.15	$\mathbf{44.22 \pm 00.13}$
	SGC	30	1	$\mathbf{70.17 \pm 00.26}$	57.47 ± 00.21	23.84 ± 00.30
	APPNP	30	2	69.35 ± 00.08	$\mathbf{66.08 \pm 00.14}$	43.62 ± 00.27
Flowers	GCN	35	1	76.59 ± 00.21	76.32 ± 00.22	79.87 ± 00.23
	SGC	25	1	70.95 ± 01.03	64.35 ± 00.93	57.16 ± 01.05
	APPNP	35	1	$\mathbf{79.68 \pm 00.26}$	$\mathbf{79.29 \pm 00.23}$	$\mathbf{81.78 \pm 00.29}$
MPEG-7	GCN	20	2	96.58 ± 00.09	96.28 ± 00.11	93.50 ± 00.19
	SGC	15	1	92.68 ± 00.35	84.16 ± 00.63	63.26 ± 00.76
	APPNP	20	2	$\mathbf{96.64 \pm 00.15}$	$\mathbf{96.39 \pm 00.16}$	$\mathbf{93.97 \pm 00.46}$
Cora	GCN	95	1	44.17 ± 00.46	43.45 ± 00.47	62.63 ± 00.62
	SGC	90	1	32.42 ± 04.35	23.91 ± 03.03	46.17 ± 03.40
	APPNP	95	1	$\mathbf{50.02 \pm 00.56}$	$\mathbf{47.62 \pm 00.66}$	$\mathbf{65.94 \pm 01.09}$
Citeseer	GCN	65	1	$\mathbf{41.44 \pm 00.28}$	$\mathbf{41.39 \pm 00.29}$	$\mathbf{66.92 \pm 00.26}$
	SGC	60	1	34.76 ± 01.86	32.13 ± 02.31	57.42 ± 02.88
	APPNP	65	1	41.26 ± 00.39	41.08 ± 00.34	66.14 ± 00.25
PubMed	GCN	40	1	$\mathbf{26.19 \pm 00.12}$	$\mathbf{24.94 \pm 00.12}$	$\mathbf{62.30 \pm 00.10}$
	SGC	95	1	14.47 ± 03.33	09.43 ± 02.82	50.08 ± 03.13
	APPNP	40	1	25.95 ± 00.32	24.46 ± 00.18	61.85 ± 00.14

foi o *APPNP*, confirmando os resultados avaliados nas diversas configurações do método proposto.

Os algoritmos de redes neurais baseadas em grafo, *SDCN* e *MinCutPool*, obtiveram os melhores resultados dentre os métodos da literatura avaliados. Como um resultado inesperado, o *K-Means* conseguiu o maior valor de *NMI* para o conjunto *PubMed*, reafirmando a versatilidade deste algoritmo clássico ao utilizar somente os vetores binários para separação do conjunto.

5.4 Avaliação Visual

O método *C-ReckNN* foi avaliado visualmente em dois aspectos: *(i)* O impacto da etapa de *manifold learning* [Pedronette; Gonçalves; Guilherme, 2018] é avaliado por meio da utilização das distâncias fornecidas pela saída ρ_r (descrita na Equação 3.7) em um método hierárquico aglomerativo; *(ii)* O método proposto, *C-ReckNN*, é comparado com outras técnicas para agrupamento de conjuntos de dados clássicos da literatura.

Em um primeiro experimento, um método de agrupamento hierárquico *average-*

Tabela 15 – Resultados obtidos para o método *SGCC*, variando o parâmetro k em um intervalo [10..100], usando as características baseadas no hipergrafo e a Equação 4.21 (produto interno das hiperarestas) como função $nc(o_i)$

Dataset	Network	k	t	NMI	V-Measure	ACC
Corel5K	GCN	100	2	91.68 ± 00.17	90.68 ± 00.16	$\mathbf{87.57 \pm 00.04}$
	SGC	65	1	87.51 ± 00.84	79.15 ± 00.89	56.20 ± 01.12
	APPNP	100	2	$\mathbf{91.89 \pm 00.08}$	$\mathbf{90.86 \pm 00.10}$	87.50 ± 00.49
CUB200	GCN	20	2	69.37 ± 00.18	64.51 ± 00.15	$\mathbf{41.45 \pm 00.18}$
	SGC	65	2	73.98 ± 00.40	55.22 ± 00.72	18.74 ± 00.46
	APPNP	25	2	$\mathbf{72.07 \pm 00.33}$	$\mathbf{65.56 \pm 00.15}$	40.67 ± 00.30
Flowers	GCN	55	2	76.77 ± 00.25	74.47 ± 00.31	75.99 ± 00.35
	SGC	50	2	78.03 ± 03.10	67.20 ± 03.12	54.82 ± 02.30
	APPNP	30	2	$\mathbf{81.95 \pm 00.28}$	$\mathbf{79.63 \pm 00.20}$	$\mathbf{79.63 \pm 00.29}$
MPEG-7	GCN	20	2	96.45 ± 00.14	96.20 ± 00.12	93.44 ± 00.16
	SGC	15	1	92.00 ± 00.27	84.07 ± 00.70	63.31 ± 00.86
	APPNP	20	2	$\mathbf{96.57 \pm 00.08}$	$\mathbf{96.48 \pm 00.08}$	$\mathbf{94.46 \pm 00.12}$
Cora	GCN	80	2	35.46 ± 00.33	33.97 ± 00.39	54.99 ± 00.53
	SGC	25	2	26.60 ± 02.69	19.15 ± 03.25	36.63 ± 03.20
	APPNP	80	2	$\mathbf{38.52 \pm 00.51}$	$\mathbf{36.74 \pm 00.59}$	$\mathbf{56.92 \pm 00.72}$
Citeseer	GCN	55	2	34.90 ± 01.08	32.95 ± 00.79	55.32 ± 00.41
	SGC	60	1	30.37 ± 07.63	16.32 ± 04.94	32.27 ± 03.00
	APPNP	55	2	$\mathbf{36.71 \pm 00.12}$	$\mathbf{34.06 \pm 00.15}$	$\mathbf{54.38 \pm 00.45}$
PubMed	GCN	45	2	28.08 ± 00.65	26.01 ± 00.72	64.64 ± 00.33
	SGC	50	1	24.00 ± 02.28	20.15 ± 01.79	52.19 ± 01.49
	APPNP	45	2	$\mathbf{30.64 \pm 00.89}$	$\mathbf{28.31 \pm 01.35}$	$\mathbf{66.52 \pm 01.30}$

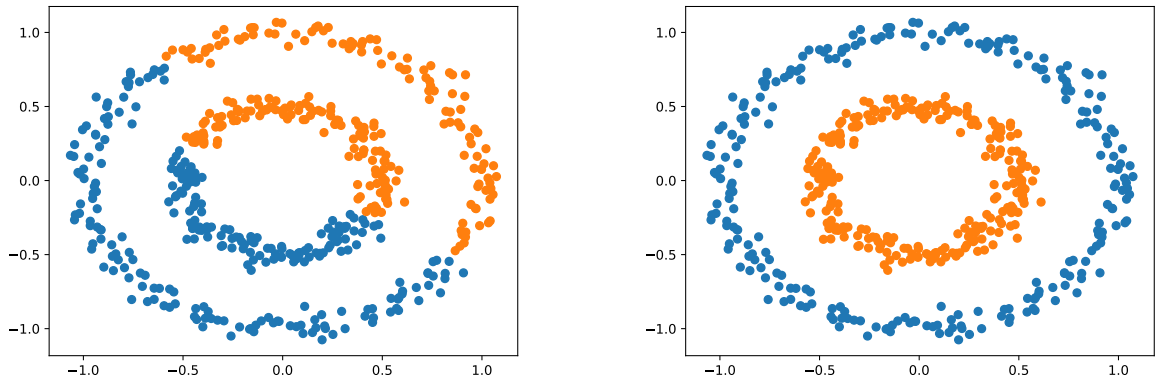
Tabela 16 – Comparação de resultados em conjuntos de imagens entre o método *SGCC*, métodos clássicos e métodos estado-da-arte.

Método	Corel5K			CUB200			Flowers			MPEG-7		
	NMI	VM	ACC	NMI	VM	ACC	NMI	VM	ACC	NMI	VM	ACC
K-Means	89.38 ±00.46	88.71 ±00.42	82.12 ±01.07	67.07 ±00.19	66.45 ±00.14	41.03 ±00.44	73.23 ±00.93	72.90 ±00.99	71.85 ±02.06	79.32 ±00.32	78.65 ±00.37	59.55 ±00.93
Agglomerative	91.03	90.65	86.68	67.06	66.24	42.03	78.06	77.03	72.64	90.43	86.76	59.00
HDBSCAN	75.66	54.91	35.28	49.94	14.89	04.30	38.60	15.98	13.52	90.16	79.32	64.92
FINCH	90.06	81.13	52.32	77.23	25.65	04.57	79.60	66.54	52.20	87.04	83.72	60.64
C-ReckNN	91.36	90.73	-	-	-	-	77.27	76.84	-	**96.99**	**96.76**	-
SDCN	87.43 ±00.36	86.95 ±00.32	81.51 ±00.74	62.62 ±00.21	61.23 ±00.18	31.76 ±00.62	67.02 ±00.99	66.73 ±00.99	36.91 ±00.58	-	-	-
MinCutPool	85.76 ±00.78	77.71 ±17.28	33.96 ±12.38	-	-	-	72.55 ±02.05	72.46 ±02.07	74.54 ±02.82	30.07 ±36.89	06.59 ±08.62	00.02 ±00.69
Abordagens Propostas												
SGCC (GCN)	92.19 ±00.14	91.96 ±00.14	90.69 ±00.16	69.07 ±00.13	68.21 ±00.12	47.52 ±00.16	80.48 ±00.57	80.18 ±00.59	82.61 ±00.61	**96.58 ±00.09**	**96.28 ±00.11**	93.50 ±00.19
SGCC (SGC)	**92.71 ±00.04**	**92.47 ±00.04**	**91.11 ±00.02**	**69.97 ±00.02**	**69.19 ±00.02**	**48.38 ±00.16**	**81.27 ±00.07**	**81.01 ±00.07**	**83.49 ±00.09**	**96.45 ±00.15**	**96.37 ±00.15**	**94.56 ±00.16**
SGCC (APPNP)	91.40 ±00.11	91.14 ±00.14	90.14 ±00.53	69.68 ±00.13	68.93 ±00.12	47.77 ±00.30	81.27 ±00.24	80.99 ±00.26	82.86 ±00.33	**96.57 ±00.08**	**96.48 ±00.08**	**94.46 ±00.12**

Tabela 17 – Comparação de resultados em redes de citação entre o método *SGCC*, métodos clássicos e métodos estado-da-arte.

Método	Entrada	Cora			Citeseer			PubMed		
		NMI	VM	ACC	NMI	VM	ACC	NMI	VM	ACC
K-Means	**X**	16.80 ±04.80	15.60 ±04.57	35.50 ±03.08	18.65 ±04.79	18.19 ±04.81	40.57 ±06.16	**35.46 ±00.07**	31.26 ±00.08	59.51 ±00.01
Agglomerative	**X**	23.39	21.93	37.22	19.76	18.97	42.23	11.75	04.04	42.59
HDBSCAN	**X**	04.84	00.39	29.87	40.01	01.29	21.52	01.38	00.06	39.84
FINCH	**X**	20.38	01.84	30.46	26.78	15.06	32.94	05.39	01.64	40.38
SDCN	**X** & **A**	21.65 ±00.16	21.17 ±00.16	38.49 ±00.18	30.96 ±00.10	30.69 ±00.10	58.09 ±00.10	07.64 ±00.28	00.02 ±00.00	39.94 ±00.00
MinCutPool	**X** & **A**	41.68 ±01.96	40.41 ±01.90	39.43 ±01.82	28.51 ±02.78	28.20 ±02.75	35.01 ±02.34	20.66 ±20.29	20.29 ±01.10	46.84 ±02.76
Metodologias Propostas										
SGCC (GCN)	**X** & **A**	44.17 ±00.46	43.45 ±00.47	62.63 ± − 00.62	**41.44 ±00.28**	**41.39 ±00.29**	**66.92 ±00.26**	33.10 ±00.16	32.20 ±00.16	68.77 ±00.05
SGCC (SGC)	**X** & **A**	45.02 ±00.15	44.81 ±00.15	62.96 ±00.09	35.50 ±00.08	35.34 ±00.08	62.95 ±00.07	31.33 ±00.80	24.75 ±00.61	53.86 ±00.06
SGCC (APPNP)	**X** & **A**	**50.02 ±00.56**	**47.62 ±00.66**	**65.94 ±01.09**	**41.26 ±00.39**	**41.08 ±00.34**	66.14 ±00.25	33.84 ±00.05	**32.91 ±00.05**	**69.51 ±00.02**

linkage foi aplicado à base de dados *Two-Circles*. Para verificar o impacto da distância gerada pela etapa de *manifold learning* do *C-ReckNN*, o mesmo método de agrupamento foi utilizado com base nas distâncias fornecidas por ρ_r. A Figura 13 apresenta os resultados. Diferentemente da execução que se baseou nas distâncias Euclidianas originais da base de dados, o método de agrupamento hierárquico aglomerativo *average-linkage* foi capaz de separar a base corretamente utilizando as novas medidas de distância.



(a) Original (b) Manifold learning

Figura 13 – Avaliação da aplicação da distância obtida na etapa de *manifold learning* aplicada ao agrupamento aglomerativo com ligação *Average-Linkage*

Este método consegue separar esta mesma base ao utilizar o *single-linkage* como tipo de ligação para união dos agrupamentos. Porém, esta ligação não obtém bons resultados em

bases de dados maiores e mais complexas, tendendo a encontrar somente um agrupamento com todos os objetos da base de dados. Desta maneira, as distâncias aprimoradas pela etapa de *manifold learning* melhoram o agrupamento utilizando a ligação *average-linkage*, provendo uma melhor separação em comparação ao método convencional.

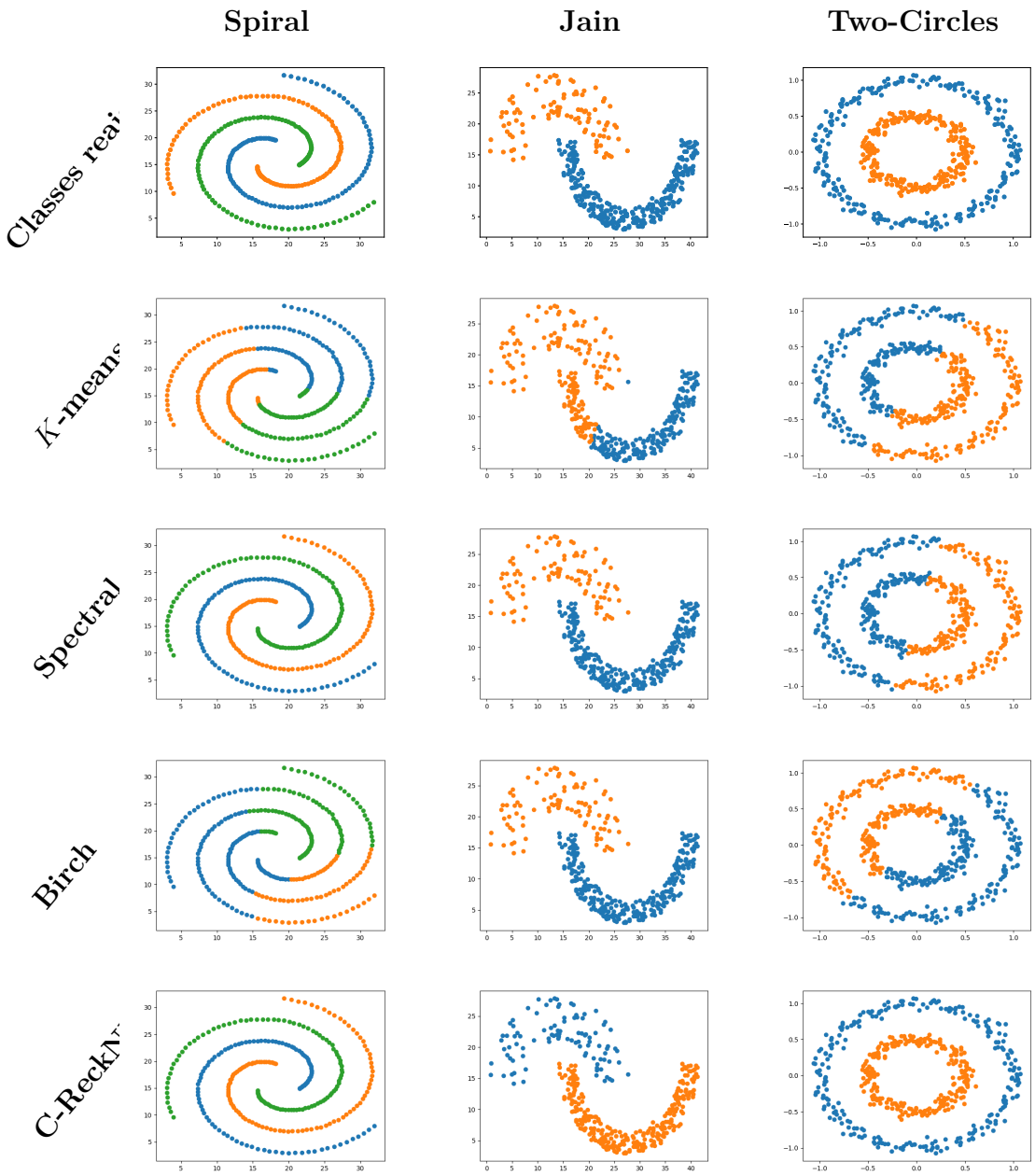


Figura 14 – Análise visual de métodos de agrupamento (linhas) e conjuntos de dados de visualização (colunas).

Em um segundo experimento, a Figura 14 apresenta a execução de três métodos da literatura em comparação com o *C-ReckNN*. Cada objeto das bases de dados é representada por um ponto no gráfico e cada cor representa o agrupamento em que este ponto foi alocado pelo respectivo método. Cada linha da imagem representa um método de agrupamento, enquanto a primeira linha demonstra o resultado esperado para cada uma das bases de

dados.

É importante notar que as cores podem variar, dependendo da ordem em que os métodos separam os agrupamentos, porém a separação da base deve ser a mesma. A metodologia proposta, *C-ReckNN*, foi capaz de separar corretamente todas as bases de dados. O método particional *K-means* [MacQueen, 1967], com a utilização da distância Euclidiana, não separou corretamente nenhuma das bases. Isto ocorre, pois, o método se baseia na definição de centros para seus agrupamentos e no particionamento dos elementos do conjunto ao redor destes centros, obtendo agrupamentos com formato elíptico.

O *Spectral* [Donath; Hoffman, 1973] é um método baseado em grafo que conseguiu separar corretamente as bases *Spiral* e *Jain*, porém não obteve sucesso na separação da base *Two Circles*. *Birch* [Zhang; Ramakrishnan; Livny, 1996] é um método hierárquico aglomerativo e, utilizando a métrica *average-linkage*, não separou corretamente as bases *Spirals* e *Two Circles*.

Os experimentos realizados demonstram que a técnica de *manifold learning* escolhida é eficaz na produção de um conjunto de informações mais confiável sobre a base de dados e que o *C-ReckNN* consegue explorar essas informações para realizar separações não-triviais, inclusive superando métodos clássicos da literatura.

Devido à utilização de uma rede neural baseada em grafos, que necessita de uma quantidade elevada de exemplos para aprender uma representação efetiva para os dados de entrada, o *SGCC* não obteve bons resultados ao separar os conjuntos de dados apresentados na Figura 14.

Desta maneira, visando avaliar a eficácia das representações aprendidas pelo *SGCC*, o método *t-SNE* [Maaten; Hinton, 2008] foi utilizado para reduzir a dimensionalidade das características originais do conjunto, das características baseadas no hipergrafo e das representações finais obtidas pelo modelo de *GCN* antes da aplicação da função *softmax*. A Figura 15 apresenta as visualizações das características dos conjuntos *Flowers*, *Cora* e *CiteSeer*.

As características são dispostas em colunas, enquanto os conjuntos são apresentados em três linhas. É possível notar que as classes, representadas por cores diferentes na visualização, apresentam uma separação maior nas características baseadas no hipergrafo em comparação com as características originais dos conjuntos para todos os exemplos analisados. Além disso, esta separação é aprimorada ao ser processada pela *GCN*.

As características originais do conjunto *Flowers* já permitem que as classes similares sejam aproximadas na representação para duas dimensões, devido a terem sido extraídas por uma rede neural profunda estado-da-arte para descrição de características. Porém, por serem criadas com base na exploração das relações do conjunto, diferentemente da rede neural utilizada com a técnica de transferência de aprendizado para extração das

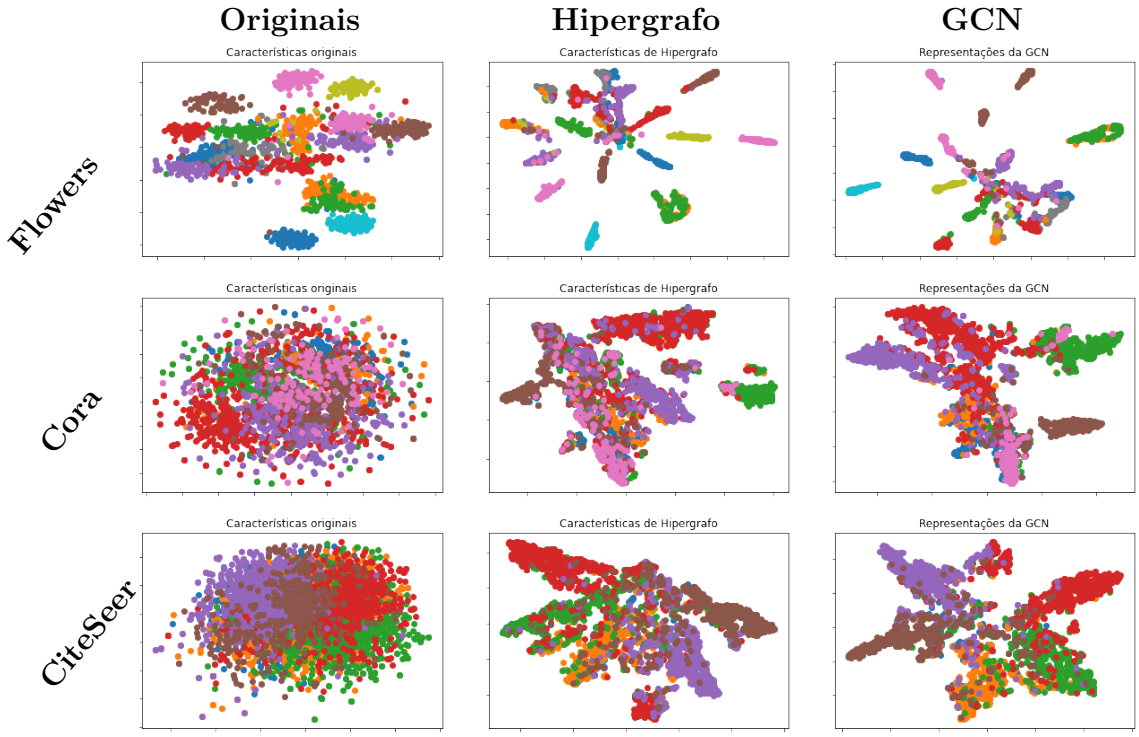


Figura 15 – Análise visual do impacto das características baseadas no hipergrafo e das representações obtidas pela GCN para três dos conjuntos de dados avaliados. As diferentes características estão dispostas nas colunas, enquanto os conjuntos estão distribuídos nas linhas da imagem.

características, as características do hipergrafo permitem que o *t-SNE* crie grupos mais concisos para cada uma das classes, sendo capaz de separar parte das mesmas em áreas isoladas da visualização. O efeito de separação é maior ao analisar as representações aprendidas pela *GCN* para o conjunto, melhorando resultados para classes que já haviam sido separadas pelas características do hipergrafo.

Ao analisar as redes de citação *Cora* e *CiteSeer*, as características originais resultam em uma grande nuvem de pontos envolvendo todas as classes, devido à sua representação binária, a qual não consegue armazenar uma representação discriminativa do conjunto. Neste cenário, as características baseadas no hipergrafo possibilitam a formação de grupos mais definidos para as classes. Mesmo a partir de vetores binários, as relações extraídas do método de *manifold learning* são capazes de unir a maioria dos elementos de cada classe em regiões próximas, formando massas mais definidas que na visualização anterior. Novamente, as representações aprendidas pela *GCN* melhoram a separabilidade do conjunto na visualização e retiram elementos alocados em classes erradas, os quais são visíveis durante a análise das características do hipergrafo.

Por fim, esse experimento demonstra a eficácia do método de *manifold learning* para obtenção de novas relações de similaridade entre os elementos e apresenta indicativos de que estas novas representações, utilizadas pelo *SGCC*, auxiliam na obtenção de agrupamentos

naturais dos conjuntos, como verificado nos três casos analisados.

6 Conclusões

Este capítulo apresenta uma discussão sobre as contribuições apresentadas e alguns direcionamentos para futuras linhas de pesquisa, com base nos conhecimentos adquiridos. A Seção 6.1 faz uma análise do que foi apresentado neste trabalho e a Seção 6.2 explora caminhos para a continuidade da pesquisa.

6.1 Contribuições e Considerações Finais

Este trabalho propôs duas novas abordagens para métodos de agrupamento. Ambas as abordagens apresentam foco na aplicação de técnicas de *manifold learning*, cujo propósito central consiste na análise da estrutura dos conjuntos de dados para obtenção de informações de similaridade mais eficazes. As abordagens propostas diferem dos métodos existentes na literatura ao expandir o uso informações de ranqueamentos e vizinhança, utilizando métodos para refinamento de similaridade, seja como etapa de pré-processamento, ou para extração de informações das estruturas internas destas técnicas, visando a construção de novos algoritmos. Neste cenário, não foi possível observar na literatura métodos estreitamente similares às abordagens propostas. Além disso, um dos métodos propostos explora arquiteturas recentes de redes convolucionais baseadas em grafos.

Conforme discutido, uma avaliação experimental diversificada foi realizada e os resultados obtidos em diferentes cenários foram promissores, se comparados a métodos clássicos e recentes da literatura. O trabalho desenvolvido também gerou publicações e submissões em veículos científicos. Uma contribuição foi publicada em congresso científico internacional realizado no ano de 2020 [Lopes. et al., 2020] e outra encontra-se em processão de submissão para relevante periódico internacional da área [Lopes; Pedronette, 2021]. Além disso, um trabalho conduzido em colaboração investigou o uso de técnicas de *manifold learning* como etapa de pré-processamento para métodos de agrupamento em outros cenários. O artigo que descreve o trabalho desenvolvido foi publicado em periódico internacional [Rozin et al., 2021].

Este trabalho também apresentou um levantamento bibliográfico sobre as técnicas de agrupamento. Foram abordados aspectos técnicos e apresentadas diferentes abordagens para a realização do processo de agrupamento, com base em categorias definidas na literatura. Também foram elencados e discutidos métodos clássicos visando ilustrar as diferentes visões sobre o tema e a evolução das abordagens ao longo do tempo. Para avaliação dos agrupamentos obtidos, foram discutidas as métricas específicas para avaliação de métodos de agrupamento. Por fim, alguns trabalhos recentes foram discutidos, em especial aqueles cuja composição tangencia ou aproxima-se dos temas centrais desta

dissertação. A discussão em torno de trabalhos recentes sobre técnicas de agrupamento destaca a constante evolução presente na área de pesquisa que, apesar de iniciada há décadas, se mantém ativa produzindo contribuições relevantes.

6.2 Trabalhos Futuros

Após a realização deste trabalho e, tendo em vista os resultados positivos obtidos, várias direções podem ser elencadas como próximos passos. Dentre elas, pode-se destacar os itens a seguir:

- **Definição de parâmetros:** Os parâmetros utilizados no *SGCC* podem ser explorados para definição de valores fixos ou para criação de heurísticas que excluam a necessidade de alguns dos valores utilizados atualmente.
- **Expansão da pesquisa com o método LHRR:** O método de *manifold learning* utilizado pelo *SGCC* se mostrou capaz de extrair e sintetizar as características dos conjuntos de dados avaliados, não somente sendo capaz de selecionar representantes para cada agrupamento, como também de criar representações efetivas com dimensões muito inferiores às originais. Estas particularidades podem ser exploradas para outras áreas como *embedding* ou seleção de elementos centrais para métodos como o *K-Means*.
- **Investigação de métodos não-supervisionados de classificação de imagens:** - Conforme citado na Seção 2.6, muitos métodos recentes de agrupamento baseados em redes neurais profundas focam em classificar conjuntos de imagem de maneira não-supervisionada, esta tem se mostrado uma área interessante e promissora para pesquisas futuras.

Referências

Benabdellah, A. C.; Benghabrit, A.; Bouhaddou, I. A survey of clustering algorithms for an industrial context. *Procedia Computer Science*, Elsevier BV, v. 148, p. 291–302, 2019. Citado 4 vezes nas páginas 17, 21, 22 e 23.

Bianchi, F. M.; Grattarola, D.; Alippi, C. Spectral clustering with graph neural networks for graph pooling. In: ACM. *Proceedings of the 37th international conference on Machine Learning (ICML)*. [S.l.], 2020. p. 2729–2738. Citado 3 vezes nas páginas 18, 42 e 78.

Bianchi, F. M.; Grattarola, D.; Livi, L.; Alippi, C. Graph neural networks with convolutional arma filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2021. Citado na página 61.

Bo, D.; Wang, X.; Shi, C.; Zhu, M.; Lu, E.; Cui, P. Structural deep clustering network. In: *Proceedings of The Web Conference (WWW) 2020*. [S.l.: s.n.], 2020. Citado 3 vezes nas páginas 18, 42 e 78.

Cai, H.; Zheng, V. W.; Chang, K. C.-C. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, v. 30, n. 9, p. 1616–1637, 2018. Citado na página 61.

Campello, R. J. G. B.; Kröger, P.; Sander, J.; Zimek, A. Density-based clustering. *WIREs Data Mining and Knowledge Discovery*, n/a, n. n/a, p. e1343, 2019. Citado 2 vezes nas páginas 30 e 31.

Chinchor, N. Muc-4 evaluation metrics. In: *Proceedings of the 4th Conference on Message Understanding*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1992. (MUC4 '92), p. 22–29. ISBN 1-55860-273-9. Citado na página 36.

Darlow, L. N.; Storkey, A. *DHOG: Deep Hierarchical Object Grouping*. 2020. Citado 2 vezes nas páginas 18 e 42.

Dempster, A. P.; Laird, N. M.; Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, v. 39, n. 1, p. 1–22, 1977. Citado na página 30.

Donath, W. E.; Hoffman, A. J. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, v. 17, n. 5, p. 420–425, Sep. 1973. Citado 2 vezes nas páginas 28 e 84.

Donoser, M.; Bischof, H. Diffusion processes for retrieval revisited. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2013. p. 1320–1327. Citado na página 38.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. [S.l.]: AAAI Press, 1996. (KDD'96), p. 226–231. Citado na página 31.

Fränti, P.; Sieranoja, S. K-means properties on six clustering benchmark datasets. *Applied Intelligence*, v. 48, n. 12, p. 4743–4759, Dec 2018. Citado na página 69.

Frey, B. J.; Dueck, D. Clustering by passing messages between data points. *Science*, American Association for the Advancement of Science, v. 315, n. 5814, p. 972–976, 2007. Citado na página 72.

Guha, S.; Rastogi, R.; Shim, K. Cure: an efficient clustering algorithm for large databases. In: ACM. *ACM Sigmod Record*. [S.l.], 1998. v. 27, n. 2, p. 73–84. Citado na página 25.

Guha, S.; Rastogi, R.; Shim, K. Rock: A robust clustering algorithm for categorical attributes. *Information systems*, Elsevier, v. 25, n. 5, p. 345–366, 2000. Citado na página 25.

He, K.; Zhang, X.; Ren, S.; Sun, J. *Deep Residual Learning for Image Recognition*. 2015. Citado 2 vezes nas páginas 72 e 73.

Hinneburg, A.; Keim, D. A. An efficient approach to clustering in large multimedia databases with noise. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. [S.l.]: AAAI Press, 1998. (KDD'98), p. 58–65. Citado na página 32.

Hoi, S. C.; Liu, W.; Chang, S.-F. Graph-based semi-supervised learning: A comprehensive review. *ACM Trans. Multimedia Comput. Commun. Appl.*, Association for Computing Machinery, New York, NY, USA, v. 6, n. 3, ago. 2010. ISSN 1551-6857. Disponível em: <https://doi.org/10.1145/1823746.1823752>. Citado na página 17.

Huang, J.; Kumar, S. R.; Mitra, M.; Zhu, W.-J.; Zabih, R. Image indexing using color correlograms. In: IEEE. *Proceedings of IEEE computer society conference on Computer Vision and Pattern Recognition*. [S.l.], 1997. p. 762–768. Citado na página 72.

Huang, S. G. J.; Zhu, X. Deep semantic clustering by partition confidence maximisation. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2020. Citado 2 vezes nas páginas 18 e 42.

Huang, Y.; Liu, Q.; Zhang, S.; Metaxas, D. N. Image retrieval via probabilistic hypergraph ranking. In: *IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR'10)*. [S.l.: s.n.], 2010. p. 3376–3383. Citado na página 57.

Hubert, L.; Arabie, P. Comparing partitions. *Journal of Classification*, v. 2, n. 1, p. 193–218, Dec 1985. Citado na página 34.

Jain, A. K. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, Elsevier BV, v. 31, n. 8, p. 651–666, jun 2010. Citado 5 vezes nas páginas 17, 21, 25, 26 e 27.

Jain, A. K.; Murty, M. N.; Flynn, P. J. Data clustering: a review. *ACM Computing Surveys*, Association for Computing Machinery (ACM), v. 31, n. 3, p. 264–323, sep 1999. Citado 5 vezes nas páginas 21, 22, 23, 25 e 33.

Jegou, H.; Schmid, C.; Harzallah, H.; Verbeek, J. Accurate image search using the contextual dissimilarity measure. *PAMI*, v. 32, n. 1, p. 2–11, 2010. Citado na página 48.

Kaufman, L.; Rousseeuw, P. J. Finding groups in data: An introduction to cluster analysis–john wiley & sons. *Inc., New York*, 1990. Citado na página 27.

Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A. S. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, v. 53, n. 8, p. 5455–5516, Dec 2020. ISSN 1573-7462. Disponível em: <https://doi.org/10.1007/s10462-020-09825-6>. Citado na página 60.

Kipf, T. N.; Welling, M. Semi-supervised classification with graph convolutional networks. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. Disponível em: <https://openreview.net/forum?id=SJU4ayYgl>. Citado 3 vezes nas páginas 61, 62 e 70.

Klicpera, J.; Bojchevski, A.; Günnemann, S. Combining neural networks with personalized pagerank for classification on graphs. In: *International Conference on Learning Representations*. [s.n.], 2019. Disponível em: <https://openreview.net/forum?id=H1gL-2A9Ym>. Citado 3 vezes nas páginas 61, 62 e 70.

Kohonen, T. The self-organizing map. *Neurocomputing*, v. 21, n. 1, p. 1 – 6, 1998. Citado na página 32.

Kuhn, H. W. The hungarian method for the assignment problem. *Naval research logistics quarterly*, Wiley Online Library, v. 2, n. 1-2, p. 83–97, 1955. Citado na página 37.

Kumar, T.; Vaidyanathan, S.; Ananthapadmanabhan, H.; Parthasarathy, S.; Ravindran, B. *Hypergraph Clustering: A Modularity Maximization Approach*. 2018. Citado na página 30.

Latecki, L. J.; Lakamper, R.; Eckhardt, U. Shape descriptors for non-rigid shapes with a single closed contour. In: *CVPR*. [S.l.: s.n.], 2000. p. 424–429. Citado na página 68.

LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Citado na página 17.

Ling, H.; Yang, X.; Latecki, L. J. Balancing deformability and discriminability for shape matching. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2010. p. 411–424. Citado na página 72.

Liu, G.-H.; Yang, J.-Y. Content-based image retrieval using color difference histogram. *Pattern Recognition*, v. 46, n. 1, p. 188 – 198, 2013. Citado na página 68.

Lopes, L. T.; Pedronette, D. C. G. Self-supervised clustering based on manifold learning and graph convolutional networks. *IEEE Transactions on Multimedia*, 2021. In Submission. Citado na página 87.

Lopes., L. T.; Valem., L. P.; Pedronette., D. C. G.; Guilherme., I. R.; Papa., J. P.; Santana., M. C. S.; Colombo., D. Manifold learning-based clustering approach applied to anomaly detection in surveillance videos. In: INSTICC. *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*. [S.l.]: SciTePress, 2020. p. 404–412. Citado 4 vezes nas páginas 19, 45, 78 e 87.

Maaten, L. van der; Hinton, G. Viualizing data using t-sne. *Journal of Machine Learning Research*, v. 9, p. 2579–2605, 11 2008. Citado na página 84.

MacQueen, J. B. Some methods for classification and analysis of multivariate observations. In: . [S.l.: s.n.], 1967. Citado 6 vezes nas páginas 25, 39, 40, 72, 78 e 84.

Masud, M. A.; Huang, J. Z.; Wei, C.; Wang, J.; Khan, I.; Zhong, M. I-nice: A new approach for identifying the number of clusters and initial cluster centres. *Information Sciences*, Elsevier BV, v. 466, p. 129–151, oct 2018. Citado na página 26.

Maćkiewicz, A.; Ratajczak, W. Principal components analysis (pca). *Computers Geosciences*, v. 19, n. 3, p. 303–342, 1993. ISSN 0098-3004. Disponível em: <https://www.sciencedirect.com/science/article/pii/009830049390090R>. Citado na página 66.

McInnes, L.; Healy, J.; Astels, S. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, v. 2, 03 2017. Citado 2 vezes nas páginas 32 e 78.

Min, E.; Guo, X.; Liu, Q.; Zhang, G.; Cui, J.; Long, J. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, v. 6, p. 39501–39514, 2018. Citado 2 vezes nas páginas 33 e 36.

Murtagh, F. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, Oxford University Press, v. 26, n. 4, p. 354–359, 1983. Citado 4 vezes nas páginas 39, 40, 72 e 78.

Nilsback, M.-E.; Zisserman, A. A visual vocabulary for flower classification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2006. v. 2, p. 1447–1454. Citado na página 68.

Ou, M.; Cui, P.; Pei, J.; Zhang, Z.; Zhu, W. Asymmetric transitivity preserving graph embedding. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 1105–1114. ISBN 9781450342322. Disponível em: <https://doi.org/10.1145/2939672.2939751>. Citado na página 66.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 69.

Pedronette, D.; Latecki, L. J. Rank-based self-training for graph convolutional networks. *Information Processing Management*, v. 58, p. 102443, 03 2021. Citado na página 62.

Pedronette, D.; Torres, R. Unsupervised rank diffusion for content-based image retrieval. *Neurocomputing*, v. 260, 05 2017. Citado na página 38.

Pedronette, D. C. G.; Gonçalves, F. M. F.; Guilherme, I. R. Unsupervised manifold learning through reciprocal knn graph and connected components for image retrieval tasks. *Pattern Recognition*, v. 75, p. 161 – 174, 2018. Distance Metric Learning for Pattern Recognition. Citado 8 vezes nas páginas 19, 38, 39, 45, 46, 49, 50 e 80.

Pedronette, D. C. G.; Torres, R. da S. Shape retrieval using contour features and distance optimization. In: CITESEER. *VISAPP (2)*. [S.l.], 2010. p. 197–202. Citado 2 vezes nas páginas 72 e 73.

Pedronette, D. C. G.; Torres, R. da S. Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition*, v. 46, n. 8, p. 2350 – 2360, 2013. Citado na página 38.

Pedronette, D. C. G.; Valem, L. P.; Almeida, J.; da S. Torres, R. Multimedia retrieval through unsupervised hypergraph-based manifold ranking. *IEEE Transactions on Image Processing*, v. 28, n. 12, p. 5824–5838, 2019. Citado 5 vezes nas páginas 19, 54, 56, 58 e 59.

Pedronette, D. C. G.; Valem, L. P.; Torres, R. da S. A bfs-tree of ranking references for unsupervised manifold learning. *Pattern Recognition*, v. 111, p. 107666, 2021. ISSN 0031-3203. Citado na página 19.

Purkait, P.; Chin, T.; Sadri, A.; Suter, D. Clustering with hypergraphs: The case for large hyperedges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 39, n. 9, p. 1697–1711, 2017. Citado na página 30.

Qin, D.; Gammeter, S.; Bossard, L.; Quack, T.; van Gool, L. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In: *CVPR*. [S.l.: s.n.], 2011. p. 777 –784. Citado 2 vezes nas páginas 38 e 46.

Ros, F.; Guillaume, S. Munec: a mutual neighbor-based clustering algorithm. *Information Sciences*, Elsevier BV, v. 486, p. 148–170, jun 2019. Citado 2 vezes nas páginas 18 e 40.

Ros, F.; Guillaume, S.; El Hajji, M.; Riad, R. Kdmutual: A novel clustering algorithm combining mutual neighboring and hierarchical approaches using a new selection criterion. *Knowledge-Based Systems*, v. 204, p. 106220, 2020. Citado na página 18.

Rosenberg, A.; Hirschberg, J. V-measure: A conditional entropy-based external cluster evaluation measure. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, 2007. p. 410–420. Citado na página 35.

Rozin, B.; Pereira-Ferrero, V. H.; Lopes, L. T.; Guimarães Pedronette, D. C. A rank-based framework through manifold learning for improved clustering tasks. *Information Sciences*, v. 580, p. 202–220, 2021. ISSN 0020-0255. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020025521008884>. Citado na página 87.

Rui Xu; Wunsch, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, v. 16, n. 3, p. 645–678, May 2005. Citado 11 vezes nas páginas 18, 21, 22, 23, 24, 25, 27, 30, 32, 33 e 34.

Sadeghi, M.; Armanfard, N. Deep clustering with self-supervision using pairwise data similarities. *TechRxiv preprint DOI: https://doi.org/10.36227/techrxiv.14852652.v1*, 2021. Citado na página 18.

Sander, J.; Ester, M.; Kriegel, H.-P.; Xu, X. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, v. 2, n. 2, p. 169–194, Jun 1998. Citado na página 32.

Sarfraz, S.; Sharma, V.; Stiefelhagen, R. Efficient parameter-free clustering using first neighbor relations. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2019. Citado 5 vezes nas páginas 18, 24, 40, 72 e 78.

Saxena, A.; Prasad, M.; Gupta, A.; Bharill, N.; Patel, O. P.; Tiwari, A.; Er, M. J.; Ding, W.; Lin, C.-T. A review of clustering techniques and developments. *Neurocomputing*, v. 267, p. 664 – 681, 2017. Citado 9 vezes nas páginas 18, 21, 23, 24, 27, 28, 30, 34 e 53.

Schubert, E.; Sander, J.; Ester, M.; Kriegel, H. P.; Xu, X. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Trans. Database Syst.*, ACM, New York, NY, USA, v. 42, n. 3, p. 19:1–19:21, jul. 2017. Citado 3 vezes nas páginas 32, 39 e 40.

Schölkopf, B.; Platt, J.; Hofmann, T. Learning with hypergraphs: Clustering, classification, and embedding. In: _____. *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*. [S.l.: s.n.], 2007. p. 1601–1608. Citado na página 30.

Sen, P.; Namata, G. M.; Bilgic, M.; Getoor, L.; Gallagher, B.; Eliassi-Rad, T. Collective classification in network data. *AI Magazine*, v. 29, n. 3, p. 93–106, 2008. Citado na página 69.

Shen, X.; Lin, Z.; Brandt, J.; Avidan, S.; Wu, Y. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2012. p. 3013–3020. Citado na página 38.

Strehl, A.; Ghosh, J. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, v. 3, p. 583–617, 01 2002. Citado na página 34.

Strehl, A.; Ghosh, J. Cluster ensembles - a knowledge reuse framework for combining partitionings. *Journal of Machine Learning Research*, v. 3, p. 583–617, 05 2002. Citado na página 34.

Torres, R. da S.; Falcão, A. X. Content-Based Image Retrieval: Theory and Applications. *Revista de Informática Teórica e Aplicada*, v. 13, n. 2, p. 161–185, 2006. Citado 2 vezes nas páginas 37 e 38.

Tsitsulin, A.; Palowitch, J.; Perozzi, B.; Müller, E. *Graph Clustering with Graph Neural Networks*. 2020. Citado na página 18.

Urquhart, R. Graph theoretical clustering based on limited neighbourhood sets. *Pattern Recognition*, v. 15, n. 3, p. 173 – 187, 1982. Citado na página 29.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. In: *International Conference on Learning Representations*. [s.n.], 2018. Disponível em: <https://openreview.net/forum?id=rJXMpikCZ>. Citado na página 61.

Vesanto, J.; Alhoniemi, E. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, v. 11, n. 3, p. 586–600, May 2000. Citado 2 vezes nas páginas 32 e 33.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, Springer, v. 17, n. 4, p. 395–416, 2007. Citado 2 vezes nas páginas 28 e 29.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. *The Caltech-UCSD Birds-200-2011 Dataset*. [S.l.], 2011. Citado na página 68.

Wang, D.; Cui, P.; Zhu, W. Structural deep network embedding. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 1225–1234. ISBN 9781450342322. Disponível em: <https://doi.org/10.1145/2939672.2939753>. Citado na página 66.

Wang, J.; Ma, Z.; Nie, F.; Li, X. Fast self-supervised clustering with anchor graph. *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–14, 2021. Citado 2 vezes nas páginas 18 e 44.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; Weinberger, K. Simplifying graph convolutional networks. In: Chaudhuri, K.; Salakhutdinov, R. (Ed.). *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019. (Proceedings of Machine Learning Research, v. 97), p. 6861–6871. Disponível em: <http://proceedings.mlr.press/v97/wu19e.html>. Citado 3 vezes nas páginas 61, 62 e 70.

Xie, Y.; Xu, Z.; Zhang, J.; Wang, Z.; Ji, S. Self-supervised learning of graph neural networks: A unified review. *arXiv preprint arXiv:2102.10757*, 2021. Citado 2 vezes nas páginas 17 e 43.

Yang, X.; Koknar-Tezel, S.; Latecki, L. J. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2009. p. 357–364. Citado na página 38.

Yang, X.; Prasad, L.; Latecki, L. J. Affinity learning with diffusion on tensor product graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 1, p. 28–38, 2013. Citado na página 38.

Zahn, C. T. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, C-20, n. 1, p. 68–86, Jan 1971. Citado na página 29.

Zhang, J.; Li, C.-G.; You, C.; Qi, X.; Zhang, H.; Guo, J.; Lin, Z. *Self-Supervised Convolutional Subspace Clustering Network*. 2019. Citado 3 vezes nas páginas 18, 43 e 44.

Zhang, T.; Ramakrishnan, R.; Livny, M. Birch: An efficient data clustering method for very large databases. *SIGMOD Rec.*, ACM, New York, NY, USA, v. 25, n. 2, p. 103–114, jun. 1996. Citado 2 vezes nas páginas 25 e 84.

Ünlü, R.; Xanthopoulos, P. Estimating the number of clusters in a dataset via consensus clustering. *Expert Systems with Applications*, Elsevier BV, v. 125, p. 33–39, jul 2019. Citado na página 26.