UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Seleção de grupos a partir de hierarquias: uma modelagem baseada em grafos

Francisco de Assis Rodrigues dos Anjos

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-C²MC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Francisco de Assis Rodrigues dos Anjos

Seleção de grupos a partir de hierarquias: uma modelagem baseada em grafos

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Ricardo José Gabrielli Barreto Campello

USP – São Carlos
Agosto de 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

A599s Anjos, Francisco de Assis Rodrigues dos
 Seleção de grupos a partir de hierarquias: uma
 modelagem baseada em grafos / Francisco de Assis
 Rodrigues dos Anjos; orientador Ricardo José
 Gabrielli Barreto Campello. -- São Carlos, 2018.
 95 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
 em Ciências de Computação e Matemática
 Computacional) -- Instituto de Ciências Matemáticas
 e de Computação, Universidade de São Paulo, 2018.

 1. agrupamento de dados. 2. critérios de
 avaliação. I. Campello, Ricardo José Gabrielli
 Barreto, orient. II. Título.

Francisco de Assis Rodrigues dos Anjos

Clusters selection from hierarchies: a graph-based model

Master dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Ricardo José Gabrielli Barreto Campello

USP – São Carlos
August 2018

À Irene, Jesuina & Tezilda.

AGRADECIMENTOS

Sozinho, nenhum objetivo pode ser alcançado. Por trás de cada realização sempre existe uma infinidade de pessoas envolvidas sem as quais tal feito jamais seria possível. Dentre tantas pessoas que, de uma maneira ou de outra, me ajudaram ao longo de todos esses anos, algumas delas não poderiam deixar de ser mencionadas.

Em primeiro lugar, agradeço ao Prof. Ricardo por: seu exemplo de caráter, comprometimento e profissionalismo; seus valiosos conselhos profissionais e pessoais; e, acima de tudo, sua compreensão nos momentos mais difíceis ao longo dessa caminhada.

Ao Cabelo, Carol, Dri, Jaci, Mari-Mari, Rê, Rô, Sagu, Siri, Vivis e todos os demais membros do Sankofa Capoeira que, mais do que amigos, se tornaram a minha família São Carlense.

Aos colegas do Laboratório de Análise de Padrões em Dados (LAPaD), Antônio, Arthur, Henrique, Jadson, Joelson, Pablo e demais colegas do Biocom.

A todos os outros colegas da USP com os quais compartilhei não só noites de trabalho e discussões acadêmicas mas também risadas e momentos de descontração. Em especial, ao Jezz (Jesimar Arantes), que me recepcionou tão bem nesta cidade até então desconhecida e com o qual travei tantos debates intermináveis sobre absolutamente tudo.

A todos os professores e funcionários do Instituto de Ciências Matemáticas e de Computação, que sempre me trataram de maneira tão cordial e me proporcionaram um ambiente de estudos sem igual.

Finalmente, gostaria de agradecer à CAPES pelo auxílio financeiro ao longo desses anos.

*“Botas... as botas apertadas são uma das maiores venturas da terra,
porque, fazendo doer os pés, dão azo ao prazer de as descalçar.”*
(Machado de Assis)

RESUMO

ANJOS, F. A. R. **Seleção de grupos a partir de hierarquias: uma modelagem baseada em grafos**. 2018. 95 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

A análise de agrupamento de dados é uma tarefa fundamental em mineração de dados e aprendizagem de máquina. Ela tem por objetivo encontrar um conjunto finito de categorias que evidencie as relações entre os objetos (registros, instâncias, observações, exemplos) de um conjunto de dados de interesse. Os algoritmos de agrupamento podem ser divididos em particionais e hierárquicos. Uma das vantagens dos algoritmos hierárquicos é conseguir representar agrupamentos em diferentes níveis de granularidade e ainda serem capazes de produzir partições planas como aquelas produzidas pelos algoritmos particionais, mas para isso é necessário que seja realizado um corte (por exemplo horizontal) sobre o dendrograma ou hierarquia dos grupos. A escolha de como realizar esse corte é um problema clássico que vem sendo investigado há décadas. Mais recentemente, este problema tem ganho especial importância no contexto de algoritmos hierárquicos baseados em densidade, pois somente estratégias mais sofisticadas de corte, em particular cortes não-horizontais denominados *cortes locais* (ao invés de globais) conseguem selecionar grupos de densidades diferentes para compor a solução final. Entre as principais vantagens dos algoritmos baseados em densidade está sua robustez à interferência de dados anômalos, que são detectados e deixados de fora da partição final, rotulados como ruído, além da capacidade de detectar *clusters* de formas arbitrárias. O objetivo deste trabalho foi adaptar uma variante da medida da Modularidade, utilizada amplamente na área de detecção de comunidades em redes complexas, para que esta possa ser aplicada ao problema de corte local de hierarquias de agrupamento. Os resultados obtidos mostraram que essa adaptação da modularidade pode ser uma alternativa competitiva para a medida de estabilidade utilizada originalmente pelo algoritmo estado-da-arte em agrupamento de dados baseado em densidade, *HDBSCAN**.

Palavras-chave: Agrupamento de dados, critérios de avaliação.

ABSTRACT

ANJOS, F. A. R. **Clusters selection from hierarchies: a graph-based model**. 2018. 95 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Cluster Analysis is a fundamental task in Data Mining and Machine Learning. It aims to find a finite set of categories that evidences the relationships between the objects (records, instances, observations, examples) of a data set of interest. Clustering algorithms can be divided into partitional and hierarchical. One of the advantages of hierarchical algorithms is to be able to represent clusters at different levels of granularity while being able to produce flat partitions like those produced by partitional algorithms. To achieve this, it is necessary to perform a cut (for example horizontal) through the dendrogram or cluster tree. How to perform this cut is a classic problem that has been investigated for decades. More recently, this problem has gained special importance in the context of density-based hierarchical algorithms, since only more sophisticated cutting strategies, in particular non-horizontal cuts (instead of global ones) are able to select clusters with different densities to compose the final solution. Among the main advantages of density-based algorithms is their robustness to noise and their capability to detect clusters of arbitrary shape. The objective of this work was to adapt a variant of the Q Modularity measure, widely used in the realm of community detection in complex networks, so that it can be applied to the problem of local cuts through cluster hierarchies. The results show that the proposed measure can be a competitive alternative to the stability measure, originally used by the state-of-the-art density-based clustering algorithm *HDBSCAN**.

Keywords: Cluster analysis, Evaluation methods.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de um conjunto de dados e o seu dendrograma correspondente. O gráfico à esquerda mostra um conjunto de pontos gerados por duas distribuições normais. Os círculos separam essas distribuições e representam um possível agrupamento. No gráfico da direita é mostrado o dendrograma que representa a hierarquia produzida pelo algoritmo <i>Single-Linkage</i> . A linha pontilhada corresponde ao nível na escada em que ocorre a solução correspondente aos círculos A e B.	34
Figura 2 – Exemplo de um conjunto de dados formado por <i>clusters</i> de diferentes densidades.	38
Figura 3 – Exemplo didático de um conjunto de dados.	41
Figura 4 – Árvore de Grupos gerada a partir da hierarquia apresentada na Tabela 3.	43
Figura 5 – Exemplo de Árvore de Grupos: valores abaixo dos nós indicam uma medida de qualidade dos respectivos <i>clusters</i>	44
Figura 6 – Corte global em dendrograma e sua limitação: à esquerda tem-se um conjunto de dados e à direita tem-se um dendrograma de agrupamento desses dados com um exemplo de corte horizontal global que retorna uma solução plana com três <i>clusters</i> , A, B e C. Não há corte global que retorne A, B, D e E simultaneamente.	60
Figura 7 – Cortes locais em diferentes níveis do dendrograma: a partir de cortes locais é possível encontrar uma solução plana composta pelos <i>clusters</i> A, B, D, E da Figura 6.	61
Figura 8 – Resultados ARI para a coleção de dados sintéticos 1 (baixa dimensionalidade): subfiguras à esquerda correspondem a bases com número de <i>clusters</i> menor (4 ou 10); subfiguras à direita correspondem a bases com número de <i>clusters</i> maior (20 ou 40).	79
Figura 9 – Resultados ARI para a coleção de dados sintéticos 2 (alta dimensionalidade): subfiguras à esquerda correspondem a bases com número de <i>clusters</i> menor (4 ou 10); subfiguras à direita correspondem a bases com número de <i>clusters</i> maior (20 ou 40).	82
Figura 10 – Resultados agregados da coleção de dados sintéticos 1. As áreas sombreadas representam os intervalos de confiança das médias (95%).	83
Figura 11 – Resultados agregados da coleção de dados sintéticos 2. As áreas sombreadas representam os intervalos de confiança das médias (95%).	83

Figura 12 – Resultados ARI para a coleção de dados reais. 86

LISTA DE TABELAS

Tabela 1	– Hierarquia simplificada do conjunto de dados da Figura 1 com $m_{\text{CISize}} = 2$. A parte de cima da tabela corresponde aos níveis hierárquicos mais altos. Na primeira linha estão os rótulos dos objetos e na primeira coluna (“Escala”) estão os níveis de dissimilaridade em que um dado objeto passou a fazer parte de um <i>cluster</i> ou que dois <i>clusters</i> se uniram.	35
Tabela 2	– Hierarquia simplificada completa produzida pelo HDBSCAN* com $m_{\text{pts}} = m_{\text{CISize}} = 3$ para o exemplo da Figura 3.	42
Tabela 3	– Hierarquia simplificada compactada gerada pelo HDBSCAN* com $m_{\text{pts}} = m_{\text{CISize}} = 3$ para o exemplo da Figura 3.	42
Tabela 4	– Coleção de dados reais utilizados durante os experimentos.	75
Tabela 5	– Resultados ARI para dados artificiais (Coleção Sintética 1 — Misturas de Gaussianas Multivariadas 2D e 10D): valores em verde (vermelho) indicam que a medida em questão obteve resultado melhor (pior) que a medida concorrente, enquanto valores em preto representam empate. Células destacadas em amarelo indicam o melhor de todos os resultados (dentre as diferentes medidas e valores de m_{pts}) para cada base. Os resultados foram subdivididos para os conjuntos de dados com 4 ou 10 <i>clusters</i> (Caso 1) e 20 ou 40 <i>clusters</i> (Caso 2).	78
Tabela 6	– Resultados ARI para dados artificiais (Coleção Sintética 2 — <i>Clusters</i> Elipsoidais 50D e 100D): valores em verde (vermelho) indicam que a medida em questão obteve resultado melhor (pior) que a medida concorrente, enquanto valores em preto representam empate. Células destacadas em amarelo indicam o melhor de todos os resultados (dentre as diferentes medidas e valores de m_{pts}) para cada base. Os resultados foram subdivididos para os conjuntos de dados com 4 ou 10 <i>clusters</i> (Caso 1) e 20 ou 40 <i>clusters</i> (Caso 2).	81
Tabela 7	– Resultados ARI para dados reais: valores em verde (vermelho) indicam que a medida em questão obteve resultado melhor (pior) que a concorrente; valores em preto representam empate. Destaque em amarelo indica o melhor de todos os resultados (dentre as diferentes medidas e valores de m_{pts}) para cada base; destaque em azul indica que este melhor resultado foi alcançado por ambas as medidas.	84

Tabela 8 – Resultados gerais obtidos ao comparar as medidas Mod. knn e Stability. Para cada coleção de dados, foram realizadas a contagem de vitórias, empates e derrotas da Mod. knn em relação à Stability. Também foi realizado o teste estatístico de Wilcoxon com o fim de verificar diferença estatística entre os resultados obtidos pelas medidas.	85
--	----

LISTA DE ABREVIATURAS E SIGLAS

AL	Average-Linkage
CL	Complete-Linkage
DBSCAN	Density Based Spatial Clustering of Applications with Noise
FOSC	Framework for Optimal Selection of Clusters
HDBSCAN*	Hierarchical DBSCAN*
SL	Single-Linkage
SSE	Sum of Squared Errors
WM	Ward's Method

LISTA DE SÍMBOLOS

$\mathbb{G}(\mathbb{V}, \mathbb{E})$ — um grafo, ou rede complexa, qualquer

$\mathbb{V}$ — conjunto de vértices nós pertencentes a um grafo

$\mathbb{E}$ — conjunto de arestas pertencentes a um grafo

N — ordem de um grafo (número de vértices de um grafo)

ℓ — tamanho um grafo (número de arestas de um grafo)

$\Gamma(v_i)$ — vizinhança do vértice v_i

k_i — grau do vértice v_i

k_i^{in} — grau de entrada do vértice v_i

k_i^{out} — grau de saída do vértice v_i

s_i — força do vértice v_i

$\mathbb{G}'$ — subgrafo de um grafo G

$\mathbb{V}'$ — subconjunto de vértices de um grafo

$\mathbb{E}'$ — subconjunto de arestas de um grafo

K_n — clique de ordem n , i.e. um grafo completo com n vértices

C_i — i -ésimo grupo (cluster, comunidade) de uma dada partição

$\mathbb{A}$ — matriz de adjacência de um grafo

$\mathbb{W}$ — matriz de pesos de um grafo

P_k — distribuição de grau de um grafo

$\delta(,)$ — delta de Kronecker

C_i — coeficiente de agrupamento do vértice i

$d(o_i, o_j)$ — distância euclidiana entre os objetos o_i e o_j

Q — função de modularidade

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Motivação	25
1.2	Definição do Problema	26
1.3	Objetivos Gerais e Hipóteses de Pesquisa	27
1.4	Organização do Trabalho	28
2	AGRUPAMENTO DE DADOS	29
2.1	Partições e Algoritmos Particionais	31
2.2	Hierarquias e Algoritmos Hierárquicos	33
2.3	Algoritmos Baseados em Densidade	36
2.3.1	<i>DBSCAN</i>	36
2.3.2	<i>DBSCAN*</i>	37
2.3.3	<i>HDBSCAN*</i>	39
2.4	FOSC	43
3	REDES COMPLEXAS	47
3.1	Conceitos Básicos	47
3.2	Medidas de Caracterização das Redes Complexas	50
3.3	Modularidade e Estrutura de Comunidades	53
3.3.1	<i>Medidas de Qualidade</i>	53
4	SELEÇÃO DE GRUPOS A PARTIR DE HIERARQUIAS: UMA MO- DELAGEM BASEADA EM GRAFOS	57
4.1	Motivação	58
4.2	Modularidade Q Ponderada como Critério de Qualidade de Agru- pamento em FOSC	64
4.2.1	<i>Modularidade Q Ponderada</i>	65
4.2.2	<i>Grafo de Similaridade</i>	66
4.2.2.1	<i>Resumo do Método Proposto</i>	68
4.2.3	<i>Caso Particular: Agrupamento HDBSCAN*</i>	69
4.2.3.1	<i>Grafo de Alcançabilidade Mútua</i>	69
4.2.3.2	<i>Modelagem de Ruído como Singletons</i>	69
4.2.3.3	<i>Escolha da Vizinhança KNN</i>	70
4.2.4	<i>Análise de Complexidade</i>	71

5	RESULTADOS	73
5.1	Configuração dos Experimentos	73
5.1.1	Bases de Dados	73
5.1.1.1	<i>Conjuntos de Dados Sintéticos</i>	73
5.1.1.2	<i>Conjuntos de Dados Reais</i>	74
5.1.2	Avaliação dos Resultados	75
5.1.3	Algoritmo e Parâmetros	75
5.1.4	Baseline	76
5.1.5	Teste Estatístico	76
5.2	Análise dos Resultados	77
5.2.1	<i>Resultados para Bases Sintéticas</i>	77
5.2.2	<i>Resultados para Bases Reais</i>	82
5.2.3	<i>Teste Estatístico</i>	85
5.3	Sumário	85
6	CONCLUSÕES	87
REFERÊNCIAS		91

INTRODUÇÃO

A análise de agrupamento de dados (*cluster analysis* ou *data clustering*) é um dos problemas conceituais fundamentais nas áreas de aprendizagem de máquina (MITCHELL, 1997; BISHOP, 2006) e mineração de dados (HAN, 2005; TAN; STEINBACH; KUMAR, 2005), caracterizada pela busca de uma categorização que exponha as relações existentes entre os objetos (instâncias, registros, observações, exemplos) de um conjunto de dados de interesse. Normalmente tais conjuntos de dados são representados por vetores de atributos (características) que descrevem todos os objetos, por medidas de (dis)similidade ou pelo inter-relacionamento entre os objetos (JAIN; DUBES, 1988; KAUFMAN; ROUSSEEUW, 2005; AGGARWAL; REDDY, 2013). Em muitas aplicações práticas, a solução deste problema frequentemente constitui a meta final do procedimento de análise, no entanto, em diversos outros casos, a solução de um problema de agrupamento é apenas a primeira etapa no processo de resolução de problemas como a classificação de padrões, a extração de regras ou a categorização de textos. Por sua ampla aplicabilidade nas mais diversas áreas de conhecimento, tais como biologia, economia, medicina, química, sociologia, etc., e por reunir elementos de diferentes disciplinas como estatística, algoritmos, reconhecimento de padrões e inteligência artificial, a análise de agrupamento de dados é considerada como um campo de pesquisa interdisciplinar (HARTIGAN, 1975; JAIN; DUBES, 1988; JAIN; MURTY; FLYNN, 1999).

1.1 Motivação

Ao longo das seis últimas décadas foi criada uma extensa literatura sobre análise de agrupamento de dados, nela uma infinidade de algoritmos com as mais diferentes características foram propostos (XU; WUNSCH II, 2005; JAIN, 2010). De maneira geral, pode-se dividir esses algoritmos em duas categorias principais de acordo com o tipo de particionamento que realizam sobre os dados, a saber, algoritmos hierárquicos e algorit-

mos particionais. Os algoritmos particionais ainda podem ser subdivididos em algoritmos particionais rígidos, em que cada objeto do conjunto de dados deve ser atribuído a exatamente um grupo, e os algoritmos particionais com sobreposição, em que um objeto pode pertencer a mais de um grupo ao mesmo tempo. Já os algoritmos hierárquicos produzem soluções de agrupamento com diferentes níveis de particionamento que podem ser pensados como uma sequência aninhada de agrupamentos particionais rígidos, nesse caso cada nível seria formado por uma partição rígida do conjunto de dados com uma quantidade distinta de subconjuntos mutualmente disjuntos, denominados de grupos (ou *clusters*). Os algoritmos hierárquicos são particularmente interessantes em diversas áreas como biologia, economia, sociologia etc., por serem capazes de representar estruturas hierárquicas naturalmente existentes nessas áreas, além disso eles também são capazes de produzir partições planas (não hierárquicas) como aquelas geradas pelos algoritmos particionais, bastando para isso cortar (por exemplo horizontalmente) o *dendrograma* ou a árvore de grupos (*cluster tree*) gerada por um algoritmo hierárquico.

A escolha de como cortar horizontalmente uma hierarquia é um problema clássico que já vem sendo investigado há décadas (MILLIGAN; COOPER, 1985; JAIN; DUBES, 1988). Apesar de amplamente investigado, esse ainda é um problema em aberto em que cada um dos algoritmos utiliza um critério próprio para decidir quais *clusters* devem formar a solução final. Dada a natureza não supervisionada do problema de agrupamento de dados, critérios de qualidade de *clusters* ou de agrupamentos são sujeitos à subjetividade do problema em si. Isso dificulta sensivelmente a escolha de um algoritmo em particular e também a escolha da configuração mais adequada para esse algoritmo em aplicações práticas. Isso também faz com que seja altamente complexo avaliar a qualidade da solução obtida por um algoritmo ou por uma configuração em particular adotada pelo analista. Tais questões estão inter-relacionadas e se referem respectivamente aos problemas de seleção de modelos e avaliação (ou validação) de resultados em aprendizado não supervisionado (JAIN; DUBES, 1988).

Dentro do paradigma de agrupamento baseado em densidade (KRIEGEL *et al.*, 2011), onde o aninhamento hierárquico de grupos e subgrupos é caracterizado por diferenças de densidade e está diretamente associado à definição de Hartigan (1975) de árvores de contorno de densidade (*density-contour trees*), os problemas de avaliação e seleção de modelos têm sido pouquíssimo explorados até então.

1.2 Definição do Problema

A fim de resolver o problema de avaliação de agrupamentos de forma objetiva, é necessário um critério ou índice quantitativo (denominado estatística ou critério de validação). Existem três tipos diferentes de critérios de validação: (i) critérios externos medem

o grau de compatibilidade entre uma estrutura de agrupamento obtida (a ser avaliada) e uma estrutura de referência (*ground truth*), fornecida previamente pelo analista. Uma vez que uma estrutura de referência não está disponível em aplicações práticas de análise de agrupamento, devido à natureza não supervisionada do problema, critérios externos são utilizados apenas para fins de investigação em experimentos controlados, envolvendo conjuntos de dados para o qual uma solução de agrupamento esperada já é conhecida, tal como quando se compara o desempenho de diferentes algoritmos usando dados sintéticos gerados a partir de distribuições de probabilidade arbitrárias (MEILĀ, 2007; VINH; EPPS; BAILEY, 2010); (ii) critérios internos medem a qualidade de uma estrutura de agrupamento obtida utilizando apenas os dados e a estrutura a ser avaliada. Como nenhuma informação externa é usada, critérios internos têm um apelo mais prático do que os externos; (iii) critérios relativos são aqueles capazes de comparar duas estruturas de agrupamento e apontar qual é a melhor em termos relativos. A maioria dos critérios externos atendem a essa exigência, mas o termo “critério de validação relativo” na literatura geralmente se refere a critérios internos que também são relativos (VENDRAMIN; CAMPELLO; HRUSCHKA, 2010).

Recentemente foi proposto um arcabouço ou *framework* para extração de agrupamentos planos a partir de hierarquias, denominado *Framework for Optimal Selection of Clusters (FOSC)* (CAMPELLO *et al.*, 2013), que consegue, a partir de um critério relativo de qualidade de *clusters* que satisfaça determinadas propriedades matemáticas desejáveis, extrair a partição ótima de qualquer hierarquia em relação a esse critério, possivelmente através de cortes não-horizontais (i.e., cortes locais ao invés de globais) através da hierarquia. No entanto, como critérios diferentes levam a soluções ótimas distintas, ainda se faz necessário investigar qual ou quais medidas de qualidade (critérios relativos) tendem a produzir melhores soluções na prática e em quais cenários, para que tais critérios possam ser considerados primeiro por um analista nos respectivos cenários. Isso mostra como o problema de avaliação, apesar de ter sido investigado extensivamente ao longo das últimas décadas, ainda constitui um dos tópicos mais desafiadores de pesquisa em análise de agrupamento de dados.

1.3 Objetivos Gerais e Hipóteses de Pesquisa

O principal objetivo deste trabalho é analisar teoricamente e experimentalmente a adaptação do critério da *modularidade* (NEWMAN; GIRVAN, 2004), amplamente utilizado na área de detecção de comunidades em redes complexas (área também denominada de agrupamento em grafos), como função objetivo a ser otimizada no processo de extração de *clusters* do *framework* FOSC (CAMPELLO *et al.*, 2013). A **hipótese de pesquisa** é: em problemas de agrupamento de dados tradicionais, descritos por vetores de atributos, é possível modelar o problema utilizando conceitos da teoria dos grafos e aplicar FOSC com

o critério adaptado da modularidade para extrair soluções de agrupamento superiores, em determinados cenários, àquelas obtidas com o critério originalmente utilizado por FOSC (estabilidade).

1.4 Organização do Trabalho

Esse documento está organizado da seguinte maneira: no Capítulo 2 são revisados os principais algoritmos de agrupamento de dados e o *framework* de extração ótima de *clusters* de hierarquias. O Capítulo 3 apresenta uma breve revisão dos principais conceitos básicos relacionados à teoria de grafos, de redes complexas. Já no Capítulo 4 são descritas as contribuições propostas por este trabalho para corte de hierarquias. No Capítulo 5 são descritos a metodologia, as bases de dados utilizadas para realização dos experimentos e os resultados obtidos. Por último, no Capítulo 6 são apresentadas as conclusões e os trabalhos futuros.

AGRUPAMENTO DE DADOS

A análise de agrupamento de dados (*cluster analysis*), também chamada apenas de agrupamento de dados (*data clustering*), é um dos problemas conceituais fundamentais tanto na área de Aprendizagem de Máquina (*Machine Learning*) (MITCHELL, 1997; BISHOP, 2006) como na área de Mineração de Dados (*Data Mining*) (HAN, 2005; TAN; STEINBACH; KUMAR, 2005). É caracterizada pela busca de uma categorização que evidencie as relações existentes entre os objetos (instâncias, registros, observações, exemplos) de um conjunto de dados de interesse. Comumente tais conjuntos de dados são representados por vetores de atributos (*feature vectors*) que descrevem todos os objetos a partir de suas características, pela quantificação de suas semelhanças a partir de medidas de (dis)similaridade ou por grafos que modelam o inter-relacionamento entre os objetos (JAIN; DUBES, 1988; KAUFMAN; ROUSSEEUW, 2005; AGGARWAL; REDDY, 2013).

Em muitas aplicações práticas, a solução deste problema frequentemente constitui a meta final do procedimento de análise, no entanto, em diversos outros casos, a solução de um problema de agrupamento é apenas a primeira etapa no processo de resolução de problemas como a classificação de padrões, a extração de regras ou a categorização de textos. Por sua ampla aplicabilidade nas mais variadas áreas de conhecimento, tais como biologia, economia, medicina, química, sociologia, etc., e por reunir elementos de diferentes disciplinas como estatística, algoritmos, reconhecimento de padrões e inteligência artificial, a análise de agrupamento de dados é considerada como um campo de pesquisa interdisciplinar (HARTIGAN, 1975; JAIN; DUBES, 1988).

Apesar de não existir uma definição exata e globalmente aceita de grupo (*cluster*), ao longo das últimas décadas, diversos métodos de agrupamento foram criados e utilizados para encontrar e analisar padrões de grupos em dados oriundos de diversas áreas (JAIN; MURTY; FLYNN, 1999; XU; WUNSCH II, 2005). Como as categorias às quais os objetos podem ser atribuídos não são conhecidas a priori, o procedimento de agrupamento de dados é considerado como um tipo de aprendizado de máquina não supervisionado. Seus

métodos podem ser divididos em dois tipos principais, os métodos particionais e os métodos hierárquicos. Os algoritmos (ou métodos) de agrupamento particionais produzem como saída um único particionamento dos dados. Já as soluções produzidas pelos métodos hierárquicos têm o formato de uma árvore que descreve o relacionamento entre grupos aninhados. Esse tipo de solução é especialmente interessante em uma série de áreas de aplicação por: (i) existir de fato uma hierarquia no conjunto de dados investigados; (ii) em algumas áreas, como na biologia, os especialistas podem preferir uma representação hierárquica por serem mais familiarizados com esse tipo de representação; (iii) ser inerente ao domínio da aplicação uma representação com diferentes níveis como na categorização de documentos; (iv) serem ferramentas úteis para visualização e análise de soluções de agrupamento em diferentes níveis de granularidade (JAIN; DUBES, 1988; EVERITT; LANDAU; LEESE, 2001).

Basicamente, o problema de particionamento pode ser definido como a busca por k grupos aos quais os N objetos do conjunto de dados devem ser atribuídos de modo que os objetos pertencentes a cada grupo sejam mais similares ou inter-relacionados entre si do que aos outros grupos. Os algoritmos particionais ainda podem ser subdivididos de acordo com o tipo de partição que produzem: algoritmos particionais exclusivos (rígidos, sem sobreposição) atribuem cada objeto a apenas um grupo, enquanto algoritmos particionais não exclusivos (*soft*, com sobreposição) atribuem um valor de pertencimento de cada objeto a cada grupo. Já os algoritmos hierárquicos podem, de acordo com a forma como operam, ser subdivididos em aglomerativos e divisivos. Os algoritmos aglomerativos iniciam o seu funcionamento atribuindo à cada objeto um grupo diferente e, a cada passo, escolhem dois (ou mais) grupos mais similares para se unirem até que todos os objetos passem a pertencer a um único grupo. No caso dos algoritmos divisivos, o processo é inverso. A partir de um único grupo que contém todos os objetos, a cada passo é escolhido um grupo para ser dividido em dois (ou mais) subgrupos até que mais nenhum grupo possa ser dividido. Em ambos os casos, podem ser definidas heurísticas que interrompam o processo antes de sua conclusão caso considerem o restante dos grupos ainda por serem criados como desnecessários (KAUFMAN; ROUSSEEUW, 2005; XU; WUNSCH II, 2005).

Nos casos em que também se deseja uma partição plana em que cada objeto é atribuído a apenas um *cluster*, esta pode ser extraída da hierarquia a partir da escolha (manual ou automática) de um de seus níveis. A escolha de como cortar horizontalmente uma hierarquia é um problema clássico que já vem sendo investigado há décadas (MILLIGAN; COOPER, 1985; JAIN; DUBES, 1988). Apesar de ser amplamente utilizado, cortes globais (i.e. horizontais) possuem limitações severas que impedem a escolha de soluções compostas por grupos identificados em diferentes níveis de abstração. Esse problema afeta especialmente os algoritmos hierárquicos baseados em densidade, no qual o aninhamento hierárquico de grupos e subgrupos é caracterizado por diferenças de densidade e está diretamente associado à definição de Hartigan (1975) de árvores de contorno de densidade

(*density-contour trees*). Nesse paradigma, um corte horizontal representa a escolha de um único limiar (*threshold*) global de densidade (TAN; STEINBACH; KUMAR, 2005; KRIEGEL *et al.*, 2011), o que também representa uma das maiores deficiências de muitos algoritmos baseados em densidade não-hierárquicos, como o DBSCAN (ESTER *et al.*, 1996) e o DENCLUE (HINNEBURG; KEIM *et al.*, 1998).

2.1 Partições e Algoritmos Particionais

Como dito, o agrupamento de dados particional busca categorizar os objetos pertencentes a um conjunto de dados de interesse em subconjuntos disjuntos. Por exemplo, pode-se supor que, em um banco de dados formado pelos registros de compras realizadas por N clientes diferentes, existam k grupos de clientes que possuam padrões de compra similares de tal maneira que todos os N clientes possam ser atribuídos a um desses k grupos. Formalmente, seja $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ o conjunto de dados contendo N objetos $\mathbf{x}_i$, $i = (1, 2, \dots, n)$. Cada objeto $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]$ é composto por n variáveis distintas.

Uma partição $\mathbf{P} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_k\}$ desse conjunto de dados, formada por k grupos, é uma coleção de subconjuntos de objetos $\mathbf{C}_j$ ($j = 1, \dots, k$), tal que:

- $\mathbf{C}_1 \cup \mathbf{C}_2 \cup \dots \cup \mathbf{C}_k = \mathbf{X}$;
- $\mathbf{C}_j \neq \emptyset$ ($\forall j$);
- $\mathbf{C}_i \cap \mathbf{C}_j = \emptyset$ ($\forall i \neq j$).

Em outras palavras: (i) cada objeto deve ser atribuído a um grupo; (ii) cada grupo possui ao menos um objeto; (iii) um objeto não pode ser atribuído a mais de um grupo ao mesmo tempo.

Assumindo que k seja conhecido, o número de possíveis formas de se agrupar N objetos em k *clusters* é dado por:

$$\frac{1}{k!} \sum_{i=0}^k (-1)^i \binom{k}{i} (k-i)^N \quad (2.1)$$

Como exemplo, para $N = 100$ e $k = 5$ existem aproximadamente $56,6 \times 10^{67}$ possíveis formas de se particionar $\mathbf{X}$. Dado que em geral k é desconhecido, o problema é ainda maior, sendo impossível enumerar e avaliar todas as partições. Por esse motivo, ao longo das últimas décadas, foram propostos diferentes algoritmos que buscam produzir boas partições. Entre eles, o algoritmo *k-Means* é provavelmente o mais conhecido de todos os algoritmos de agrupamento de dados (JAIN, 2010).

A ideia por trás do algoritmo *k-Means* já era discutida desde a década de 50 (apesar da publicação mais citada ser datada de 1967 (MACQUEEN, 1967)). Na sua forma original, o algoritmo assume que todas as n variáveis que descrevem o conjunto de dados são reais, i.e., cada objeto $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]$ é um vetor numérico que pode ser entendido como um ponto em um espaço n -dimensional.

Assumindo que existam k grupos em $\mathbf{X}$, o algoritmo *k-Means* cria k protótipos e tenta ajustá-los de modo que cada um deles represente, da melhor forma possível, um dos k grupos presentes nos dados. Isto quer dizer que o protótipo de um *cluster* deve ser o ponto mais similar aos objetos pertencentes a este. Essa similaridade é definida como a soma das distâncias (euclidianas) quadradas entre tal ponto e os objetos pertencentes ao *cluster*. Assim, o objetivo do algoritmo *k-Means* é selecionar k pontos (protótipos) $\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_k$ tal que a soma das distâncias quadradas (Sum of Squared Errors (SSE)) desses pontos aos objetos de seus respectivos *clusters* seja minimizada. Para resolver esse problema, o *k-Means* utiliza uma abordagem iterativa composta de duas etapas principais: (i) a atribuição dos objetos a um dos protótipos; e (ii) a atualização destes protótipos. Em termos gerais, o algoritmo é definido pelos seguintes passos:

1. Escolhe-se k protótipos aleatoriamente;
2. Atribui-se cada objeto ao protótipo mais próximo;
3. Recalcula-se os protótipos de cada *cluster* como a média multivariada dos objetos daquele *cluster*;
4. Repete-se os passos 2 e 3 até que mais nenhum objeto mude de *cluster* (ou até que um dado critério de parada seja atingido).

Em cada iteração, é garantido que o algoritmo não aumenta a SSE de maneira que, eventualmente convirja para uma solução com valor mínimo de SSE. Apesar de sempre convergir atingindo um mínimo, não existe garantia que o valor atingido seja o mínimo global. Assim, os resultados alcançados podem variar significativamente, dependendo da escolha dos protótipos iniciais. Além disso, o *k-Means* possui outros dois problemas críticos: (i) exige que o número de grupos k seja fornecido pelo usuário, o que na prática normalmente não é conhecido; e (ii) o uso de protótipos induzem grupos volumétricos globulares, sendo geralmente incapazes de detectar grupos com formas arbitrárias. Os algoritmos hierárquicos, discutidos na sequência, não possuem a limitação (i) e, alguns deles, também não possuem a limitação (ii), apesar de sofrerem de outros problemas.

2.2 Hierarquias e Algoritmos Hierárquicos

Ao contrário dos algoritmos de agrupamento de dados particionais, em que se produz uma única solução plana (*flat*, i.e., não-hierárquica), os algoritmos de agrupamento de dados hierárquicos produzem uma coleção composta por diversas soluções com variados números de grupos. Nessa coleção, o número de grupos existentes em cada partição variam de $k = 1$ (um único grupo contendo todos os objetos) até $k = N$ (N grupos compostos, cada um, por apenas um único objeto). Além disso, existe uma relação entre as partições que a compõem de modo que a partição $\mathbf{P}_k$, composta por k diferentes grupos, está contida na partição $\mathbf{P}_{k-1}$, composta por $k - 1$ grupos. Uma partição $\mathbf{P}_k$ qualquer é considerada contida em outra partição $\mathbf{P}_l$ se, qualquer grupo $\mathbf{C}_i$ pertencente a $\mathbf{P}_k$ é um subconjunto de algum grupo $\mathbf{C}_j$ pertencente a $\mathbf{P}_l$. Por exemplo, seja $\mathbf{P}_4 = (\{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6\}, \{\mathbf{x}_7, \mathbf{x}_8\}, \{\mathbf{x}_5, \mathbf{x}_9\})$ e $\mathbf{P}_3 = (\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_5, \mathbf{x}_9\}, \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6\}, \{\mathbf{x}_7, \mathbf{x}_8\})$, então $\mathbf{P}_4 \subset \mathbf{P}_3$. Nesse caso, as partições são ditas aninhadas e tal coleção recebe o nome de hierarquia.

Os algoritmos de agrupamento de dados hierárquicos podem construir essas hierarquias de maneira aglomerativa (começando de $\mathbf{P}_N$ e unindo, dois a dois, os *clusters* mais similares) ou de maneira divisiva (começando de $\mathbf{P}_1$ e escolhendo um *cluster* para ser dividido). Na prática, os algoritmos aglomerativos são mais conhecidos e utilizados por serem mais simples e intuitivos. A seguir, estão descritos os principais passos de um algoritmo hierárquico aglomerativo:

1. Inicialmente considera-se cada objeto como sendo um *cluster* individual (i.e., um *singleton*);
2. Une-se o par de *clusters* mais similar, de acordo com uma dada medida de (dis)similaridade entre *clusters*;
3. Repete-se o passo 2 até restar somente um único *cluster* contendo todos os objetos do conjunto de dados.

Note que no segundo passo, como dois *clusters* são unidos, o número total de *clusters* é reduzido em uma unidade, i.e., a partição $\mathbf{P}_k$ que possui k *clusters* é transformada na partição $\mathbf{P}_{k-1}$ que contém $k - 1$ *clusters*. Logo, o algoritmo produz N partições com $k = N, N - 1, \dots, 1$ *clusters*, respectivamente.

Para que a descrição de um algoritmo aglomerativo genérico, como exposto o acima, esteja completa, é necessário especificar como é calculada a (dis)similaridade entre os grupos. Basicamente é essa escolha que diferencia os diferentes tipos de algoritmos aglomerativos. Dentre eles, os mais utilizados na prática são:

Single-Linkage (SL): No algoritmo SL a distância entre dois grupos de objetos, C_i e C_j , é definida pela menor de todas as distâncias entre um objeto qualquer pertencente a C_i e outro objeto pertencente a C_j .

Complete-Linkage (CL): Ao contrário do SL, que utiliza a menor distância entre dois grupos, o algoritmo CL utiliza a maior distância entre dois grupos como medida de dissimilaridade.

Average-Linkage (AL): Como o nome já indica, o algoritmo AL se baseia na média das distâncias entre todos os objetos de dois grupos.

Ward's Method (WM): A medida usada pelo WM é um pouco mais complicada do que as descritas anteriormente. No método de Ward a distância entre dois *clusters* é dada pelo aumento da SSE que seria observado se esses *clusters* fossem unidos e representados por um único centroide calculado a partir de todos os objetos dos dois *clusters* em questão conjuntamente.

Os algoritmos hierárquicos descritos acima possuem diferentes pontos positivos e negativos: o SL é o único algoritmo capaz de identificar *clusters* de formato arbitrário, os outros favorecem *clusters* de formato volumétrico, como os *clusters* globulares favorecidos pelo *k-Means*. No entanto, o SL é muito sensível a ruído, especialmente a “ruídos de fundo” que podem criar pontes artificiais entre os grupos, causando assim a união indesejada desses grupos. O CL previne esse efeito mas é sensível à presença de *outliers*, que distorcem drasticamente a maior distância entre grupos (medida utilizada pelo CL) causando a quebra dos *clusters* muito grandes. O AL e o WM são em geral mais estáveis e confiáveis, porém eles não conseguem detectar *clusters* de formatos arbitrários.

Figura 1 – Exemplo de um conjunto de dados e o seu dendrograma correspondente. O gráfico à esquerda mostra um conjunto de pontos gerados por duas distribuições normais. Os círculos separam essas distribuições e representam um possível agrupamento. No gráfico da direita é mostrado o dendrograma que representa a hierarquia produzida pelo algoritmo *Single-Linkage*. A linha pontilhada corresponde ao nível na escala em que ocorre a solução correspondente aos círculos A e B.

Fonte: Elaborada pelo autor.

Independentemente do algoritmo hierárquico utilizado, a hierarquia produzida pode ser visualizada por meio de um dendrograma, como mostrado na [Figura 1](#), à direita. O conjunto de dados, mostrado à esquerda, foi agrupado utilizando-se o algoritmo SL. Um dendrograma pode ser entendido como “árvore invertida” em que os objetos são colocados na parte de baixo (folhas), os grupos são representados por linhas verticais e suas uniões são representadas por linhas horizontais no nível da escala correspondente à união entre eles.

Algumas implementações dos algoritmos hierárquicos permitem que o usuário especifique, por meio de um parâmetro m_{CISize} , qual o tamanho mínimo dos *clusters* que devem ser considerados relevantes. Assim, qualquer *cluster* que seja menor do que m_{CISize} é automaticamente descartado como um componente espúrio ou *ruído* ([CAMPELLO et al., 2013](#)). A [Tabela 1](#) mostra a hierarquia produzida pelo Single-Linkage para $m_{\text{CISize}} = 2$. Cada linha da [Tabela 1](#) armazena o rótulo do *cluster* que contém o objeto da coluna correspondente.

Tabela 1 – Hierarquia simplificada do conjunto de dados da [Figura 1](#) com $m_{\text{CISize}} = 2$. A parte de cima da tabela corresponde aos níveis hierárquicos mais altos. Na primeira linha estão os rótulos dos objetos e na primeira coluna (“Escala”) estão os níveis de dissimilaridade em que um dado objeto passou a fazer parte de um *cluster* ou que dois *clusters* se uniram.

Escala	A7	A1	A6	A2	A4	A3	A5	B3	B6	B5	B1	B4	B2	B7
2.4785	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1.1304	2	2	2	2	2	2	2	3	3	3	3	3	3	3
1.0911	2	2	2	2	2	2	2	5	5	4	4	4	4	4
1.0588	0	2	2	2	2	2	2	5	5	4	4	4	4	4
1.0585	0	6	6	7	7	7	7	5	5	4	4	4	4	4
0.9522	0	6	6	7	7	7	7	5	5	8	8	8	9	9
0.9402	0	6	6	7	7	7	7	0	0	8	8	8	9	9
0.8060	0	6	6	0	7	7	7	0	0	8	8	8	9	9
0.6158	0	6	6	0	7	7	7	0	0	8	8	8	0	0
0.4870	0	6	6	0	7	7	7	0	0	0	8	8	0	0
0.3934	0	6	6	0	0	7	7	0	0	0	8	8	0	0
0.3235	0	6	6	0	0	0	0	0	0	0	8	8	0	0
0.1608	0	0	0	0	0	0	0	0	0	0	8	8	0	0
0.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborada pelo autor.

Em uma hierarquia tradicional, todos os objetos são rotulados, mesmo quando correspondem a *singletons* (i.e., um *cluster* composto de um único objeto). Além disso, em uma interpretação “de cima para baixo” (visão *top-down*) das hierarquias tradicionais, um único objeto se desconectando dos demais objetos de um *cluster* já se caracteriza como uma divisão em dois *clusters*. Já nas hierarquias produzidas com $m_{\text{CISize}} > 1$, como

aquela da [Tabela 1](#), todos os componentes menores do que m_{ClSize} são descartados e seus objetos recebem rótulo 0, indicando que eles não pertencem a nenhum *cluster* naquele nível da hierarquia. Esses componentes não são interpretados como *clusters* e, portanto, o surgimento deles conforme se percorre a hierarquia de cima para baixo não são vistos como uma divisão do *cluster* original a que esses pertenciam, logo o *cluster* original continua existindo com o mesmo rótulo (vide por exemplo o *cluster* 2 na [Tabela 1](#), que mantém o mesmo rótulo independente da presença ou não do objeto A7).

Independente do uso ou não de m_{ClSize} e da simplificação correspondente da hierarquia, para se obter uma partição plana a partir de um dendrograma ou hierarquia simplificada, é necessário escolher um nível da escala e então realizar um corte horizontal (como indicado pela linha pontilhada da [Figura 1](#)). No entanto, essa metodologia possui limitações, como a impossibilidade de se selecionar *clusters* em diferentes níveis hierárquicos. Esses problemas serão discutidos mais adiante na [Seção 2.4](#).

2.3 Algoritmos Baseados em Densidade

Os algoritmos de agrupamento de dados baseados em densidade surgiram das tentativas de se produzir agrupamentos robustos a ruídos que também conseguissem detectar grupos de formas arbitrárias e determinar automaticamente o número de grupos k presentes nos dados. Esses algoritmos entendem *clusters* como sendo regiões com alta densidade (alta concentração) de objetos separadas por regiões com baixa densidade de objetos.

2.3.1 DBSCAN

O Density Based Spatial Clustering of Applications with Noise ([ESTER et al., 1996](#)), ou apenas DBSCAN, é sem dúvida o mais conhecido algoritmo de agrupamento de dados baseado em densidade. Sua ideia é muito simples. Essencialmente, para cada objeto $\mathbf{x}_i$, é determinado quantos objetos se situam a uma distância $d(\cdot, \cdot)$ inferior a ϵ , i.e., quantos objetos estão em sua ϵ -vizinhança. Em um espaço de duas dimensões (um plano), quando a distância euclidiana é usada, a ϵ -vizinhança de um ponto $\mathbf{x}_i$ é constituída por todos os pontos (incluindo o próprio $\mathbf{x}_i$) situados dentro de um círculo de centro $\mathbf{x}_i$ e raio ϵ . Caso existam pelo menos m_{pts} objetos pertencentes à ϵ -vizinhança de $\mathbf{x}_i$, então $\mathbf{x}_i$ é considerado um *core point*.

Portanto, um *core point* é um objeto que é considerado denso em relação a um limiar (*threshold*) de densidade, determinando a partir dos parâmetros ϵ e m_{pts} . Todo *core point* é parte de um *cluster*. Além disso, dois *core points* presentes na ϵ -vizinhança um do outro são ditos conectados e, portanto, fazem parte do mesmo *cluster*. Assim, no paradigma de agrupamento baseado em densidade, um *cluster* é definido como um conjunto máximo (*maximal set*) de objetos conectados. Os objetos que não são considerados *core*

points (i.e., não são densos) mas, ainda assim, pertencem à ε -vizinhança de um *core point*, são inseridos no *cluster* correspondente. Tais objetos são chamados de *border points* por, em geral, estarem situados nas fronteiras (extremidades) de um *cluster*. O restante dos objetos, que não são nem *core points* nem *border points*, são declarados como sendo ruído (*noise*) e tratados como *outliers*, sendo deixados de fora de qualquer um dos *clusters*.

Na próxima subseção é definido formalmente o algoritmo DBSCAN*, uma versão simplificada do DBSCAN, que é particularmente importante no escopo desta dissertação por servir de base para o algoritmo HDBSCAN*, descrito na [Subseção 2.3.3](#) e posteriormente utilizado nos [Capítulo 4](#) e [Capítulo 5](#).

2.3.2 DBSCAN*

[Campello, Moulavi e Sander \(2013\)](#) propõem o DBSCAN* como uma versão simplificada do DBSCAN, definida apenas em termos de *core* e *noise objects* que, assim como o DBSCAN, necessita apenas de uma medida de dissimilaridade entre pares de objetos, $d(\mathbf{x}_i, \mathbf{x}_j)$, e um limiar de densidade, determinado por ε e m_{pts} . Comumente, assume-se que tal medida $d(\cdot, \cdot)$ como sendo uma distância, porém não é necessário supor que seja uma métrica.

Definição 2.3.1 (*Core e Noise*). Um objeto $\mathbf{x}_i$ é chamado de *core* em relação a ε e m_{pts} se sua ε -vizinhança contém pelo menos m_{pts} objetos, i.e., se $|N_\varepsilon(\mathbf{x}_i)| \geq m_{pts}$, em que $|N_\varepsilon(\mathbf{x}_i)| = \{\mathbf{x}_j \in \mathbf{X} \mid d(\mathbf{x}_i, \mathbf{x}_j) \leq \varepsilon\}$. Caso contrário, esse objeto é denominado de *noise*.

Definição 2.3.2 (ε -alcançável). Dois objetos *core* $\mathbf{x}_i$ e $\mathbf{x}_j$ são ε -alcançáveis em relação a ε e m_{pts} se $\mathbf{x}_i \in N_\varepsilon(\mathbf{x}_j)$ e $\mathbf{x}_j \in N_\varepsilon(\mathbf{x}_i)$.

Definição 2.3.3 (Densamente Conectados). Dois objetos *core* $\mathbf{x}_i$ e $\mathbf{x}_j$ são densamente conectados em relação a ε e m_{pts} se eles são diretamente ou transitivamente ε -alcançáveis.

Definição 2.3.4 (*Cluster*). Um *cluster* $\mathbf{C}$ com relação a ε e m_{pts} é um subconjunto máximo não-vazio de $\mathbf{X}$ tal que cada par de objetos em $\mathbf{C}$ é densamente conectado.

Como no DBSCAN, dois parâmetros são necessários para definir se um dado objeto $\mathbf{x}_i$ é considerado denso, a saber, o número mínimo de objetos m_{pts} internos a uma hipersfera centralizada no objeto $\mathbf{x}_i$, e o raio ε dessa hipersfera. Diferente de DBSCAN, entretanto, os *clusters* são formados apenas pelos objetos que satisfaçam esse limiar mínimo de densidade (objetos *core*). Dois objetos estão no mesmo *cluster* se e somente se eles forem densos e capazes de alcançar um ao outro, diretamente ou por meio de uma sequência de pares de objetos contidos na ε -vizinhança um do outro.

O DBSCAN* não é apenas mais simples do que o DBSCAN, como também mais rigoroso do ponto de vista estatístico, por estar em conformidade com o princípio teórico de

grupos de contorno de densidade (*density-contour clusters*) definidos por [Hartigan \(1975\)](#). O algoritmo original do DBSCAN flexibiliza esse princípio ao permitir que alguns objetos abaixo do limiar de densidade (os objetos *border*) sejam incorporados aos *clusters*. No DBSCAN, um objeto *border* é, especificamente, um objeto não-*core* que se situa dentro da ϵ -vizinhança de um objeto *core*. Assim, os grupos encontrados pelo DBSCAN* são exatamente os mesmos identificados pelo DBSCAN, desde que sejam desconsiderados os *borders*.

Existem argumentos teóricos que discutem a importância do parâmetro m_{pts} , considerando-o como algo não crítico visto que pequenas variações no seu valor não causam mudanças significativas no agrupamento obtido. Em outras palavras, são esperados resultados semelhantes para diferentes valores de m_{pts} dentro de um dado intervalo. A razão é que o m_{pts} pode ser interpretado como um fator de suavização de um estimador de densidade não-paramétrico ([CAMPELLO et al., 2015](#)). No entanto, o mesmo não pode ser dito sobre o parâmetro ϵ , posto que pequenas mudanças em seu valor geralmente podem causar grandes mudanças nos agrupamentos encontrados. Por esse motivo, tanto o DBSCAN como o DBSCAN* são extremamente sensíveis à escolha do parâmetro ϵ .

Apesar de existirem heurísticas que tentam estimar bons valores para ϵ , essas heurísticas são limitadas, visto que elas dependem de uma série de suposições que dificilmente são observadas na prática. Além disso, tanto o DBSCAN como o DBSCAN* são incapazes de encontrar soluções formadas por grupos com diferenças acentuadas de densidade. Por exemplo, em conjuntos de dados como o da [Figura 2](#) (à direita), pode-se argumentar que a solução ideal é composta pelos grupos **A**, **B**, **C**, entretanto, caso seja escolhido um limiar de densidade capaz de detectar os *clusters* **B** e **C**, o *cluster* **A** praticamente desaparece, tendo seus objetos descartados como ruído. Por outro lado, se é escolhido um limiar adequado para a descoberta de **A**, então **B** e **C** são unidos a ele.

Figura 2 – Exemplo de um conjunto de dados formado por *clusters* de diferentes densidades.

Fonte: Elaborada pelo autor.

Além de serem incapazes de encontrar soluções com diferenças acentuadas de densidade, algoritmos tais como DBSCAN e DBSCAN* são particionais, ou seja, eles geram uma solução plana de agrupamento. Algoritmos particionais descartam completamente a estrutura hierárquica que pode estar presente no conjunto de dados. Na [Figura 2](#) à esquerda, pode-se perceber que existe uma estrutura hierárquica entre os *clusters*, em que os *clusters* menos densos são compostos por outros *subclusters* mais densos, relações que não podem ser capturadas ou descritas por uma solução de agrupamento plana. Essas limitações levaram à criação dos algoritmos de agrupamento hierárquicos baseados em densidade, como HDBSCAN*, que não necessita do parâmetro ϵ e também é capaz identificar grupos com diferentes densidades.

2.3.3 HDBSCAN*

Proposto por [Campello, Moulavi e Sander \(2013\)](#), o Hierarchical DBSCAN* (ou apenas HDBSCAN*) é considerado como estado da arte dos algoritmos de agrupamento hierárquicos baseados em densidade. Em [Campello et al. \(2015\)](#), ele é estendido para outros domínios, como o agrupamento semi-supervisionado, se transformando em um arcabouço (*framework*) capaz de atuar em: (i) agrupamento de dados de maneira não-supervisionada (o domínio original); (ii) agrupamento de dados semi-supervisionado, em que restrições *should-link* e *should-not-link* são fornecidas; (iii) detecção de *outliers*; e (iv) visualização de dados.

Dado um valor qualquer para m_{pts} , o HDBSCAN* é capaz de calcular de maneira eficiente (possivelmente quase tão rápido quanto uma única execução do DBSCAN) uma hierarquia de partições para todos os possíveis valores de $\epsilon \in [0, \infty)$. Cada nível hierárquico (cada partição) está associado a um valor de ϵ que, se utilizado pelo DBSCAN*, produziria exatamente a mesma solução. Caso uma solução DBSCAN seja desejada, basta para isso que a hierarquia seja cortada no nível ϵ correspondente e adicionar os objetos *border* (que pode ser feito em tempo linear).

O HDBSCAN* também é capaz de criar uma hierarquia pós-processada denominada de *hierarquia simplificada* que descreve uma árvore de *clusters* (*cluster tree*) contendo apenas os *clusters* mais importantes. Essa hierarquia simplificada pode ser representada de forma compactada, para isso sendo necessário armazenar apenas os níveis hierárquicos em que ocorreu a divisão real de um *cluster*, sendo descartados todos os níveis em que um *cluster* apenas “encolheu” (*shrink*) pela perda de objetos que deixaram de ser considerados densos se tornando ruído (ou pela perda de componentes espúrios, caso o uso opcional de um tamanho mínimo de *cluster*, m_{CISize} , seja adotado).

Além dos conceitos definidos na [Subseção 2.3.2](#), o HDBSCAN* utiliza dois conceitos adicionais, a distância *core* e a distância de alcançabilidade mútua, definidos como:

Definição 2.3.5 (Distância *Core*). A distância *core* $d_{\text{core}}(\mathbf{x}_i)$ de um objeto $\mathbf{x}_i \in \mathbf{X}$ em relação a m_{pts} é igual a distância de $\mathbf{x}_i$ ao seu m_{pts} -ésimo vizinho mais próximo (incluindo ele mesmo).

Definição 2.3.6 (Distância de Alcançabilidade Mútua). A distância de alcançabilidade mútua entre dois objetos $\mathbf{x}_i$ e $\mathbf{x}_j$ pertencentes a $\mathbf{X}$, em relação a m_{pts} , é definida como $d_{\text{mreach}}(\mathbf{x}_i, \mathbf{x}_j) = \max\{d_{\text{core}}(\mathbf{x}_i), d_{\text{core}}(\mathbf{x}_j), d(\mathbf{x}_i, \mathbf{x}_j)\}$.

A distância de alcançabilidade mútua $d_{\text{mreach}}(\mathbf{x}_i, \mathbf{x}_j)$ é o menor valor de raio ε para o qual o par de objetos $\mathbf{x}_i$ e $\mathbf{x}_j$ são considerados *core* e ε -alcançáveis um do outro (Definição 2.3.2). Assim, para um dado m_{pts} , ε estabelece um limiar de densidade $\frac{1}{\varepsilon}$, inversamente proporcional ao seu valor. Portanto, a distância de alcançabilidade mútua é inversamente proporcional ao maior limiar de densidade para o qual $\mathbf{x}_i$ e $\mathbf{x}_j$ são densamente conectados (Definição 2.3.3).

O HDBSCAN* fornece como resultado uma hierarquia completa composta por todas as possíveis soluções que poderiam ser encontradas pelo DBSCAN*, dado um valor qualquer de m_{pts} e um intervalo infinito de valores para $\varepsilon \in [0, \infty)$. Para isso, utiliza-se (apenas conceitualmente, pois na prática não é necessário armazená-lo) o seguinte grafo¹ de distâncias transformadas (CAMPELLO; MOULAVI; SANDER, 2013; CAMPELLO *et al.*, 2015):

Definição 2.3.7 (Grafo de Alcançabilidade Mútua). O grafo de alcançabilidade mútua $\mathbb{G}_{m_{\text{pts}}}$ é um grafo ponderado completo, no qual cada vértice v_i representa um objeto $\mathbf{x}_i \in \mathbf{X}$ e o peso de cada aresta $\{v_i, v_j\}$ é igual à distância de alcançabilidade mútua entre $\mathbf{x}_i$ e $\mathbf{x}_j$ em relação a m_{pts} .

Seja $\mathbb{G}_{m_{\text{pts}}, \varepsilon} \subseteq \mathbb{G}_{m_{\text{pts}}}$ o subgrafo obtido pela remoção de todas as arestas que possuam peso maior do que um dado valor de ε . Pelos argumentos apresentados anteriormente, percebe-se que os *clusters* identificados pelo DBSCAN*, em relação a m_{pts} e ε , correspondem às componentes conexas de $\mathbb{G}_{m_{\text{pts}}, \varepsilon}$ e os objetos rotulados como ruído (*noise*) aos vértices isolados pela remoção dessas arestas. Essa relação permite que todos os agrupamentos que o DBSCAN* é capaz de encontrar ($\varepsilon \in [0, \infty)$) sejam gerados de maneira hierárquica pela remoção das arestas de $\mathbb{G}_{m_{\text{pts}}}$ em ordem decrescente de peso. É interessante notar que essa é essencialmente a definição baseada em grafo do algoritmo de agrupamento hierárquico Single-Linkage (JAIN; DUBES, 1988) e, portanto, existe uma relação conceitual entre os algoritmos DBSCAN* e SL sobre o espaço transformado das distâncias de alcançabilidade mútua. O agrupamento obtido pelo DBSCAN* em relação a um m_{pts} e algum valor de ε é idêntico a primeiramente executar o algoritmo SL sobre os

¹ Grafos, subgrafos, vértices, arestas e outros assuntos relacionados são explicados no Capítulo 3.

espaço transformado das distâncias de alcançabilidade mútua (em relação a m_{pts}) e, então, cortar o dendrograma resultante no nível ϵ e tratar todos os *singletons* com $d_{\text{core}} > \epsilon$ como ruído. Isso mostra que é possível criar uma versão hierárquica do DBSCAN* pela aplicação de um algoritmo que computa uma hierarquia SL sobre o espaço transformado.

Uma das formas mais eficientes de se calcular uma hierarquia SL é pela utilização de um algoritmo divisivo que atua removendo as arestas, em ordem decrescente de peso, de uma árvore geradora mínima (*Minimum Spanning Tree* — MST) (GOWER; ROSS, 1969; JAIN; DUBES, 1988). No presente contexto, trata-se de uma MST no espaço transformado das distâncias de alcançabilidade mútua, denominada aqui MST_r . A fim de representar o nível da hierarquia a partir do qual um objeto se torna um ruído ($d_{\text{core}} > \epsilon$) e acima do qual ele pode ser parte de um *cluster*, ou mesmo um *cluster* por se só (i.e., um *singleton*), o HDBSCAN* amplia essa MST_r com a adição de laços (*loops*), com peso igual à distância *core* (d_{core}) do respectivo objeto representado pelo vértice. Assim, de maneira geral, o HDBSCAN*: (i) computa a MST_r usando algum método computacionalmente eficiente (por exemplo, o algoritmo de Prim); (ii) adiciona à MST_r auto-arestas contendo a distância *core* de cada vértice; e (iii) remove as arestas em ordem decrescente (em caso de empate, todas são removidas simultaneamente) e rotula as componentes conexas como *clusters*.

Figura 3 – Exemplo didático de um conjunto de dados.

Fonte: Elaborada pelo autor.

O HDBSCAN* também permite que, opcionalmente, seja informado um tamanho mínimo m_{ClSize} para os *clusters* produzidos (comumente utiliza-se $m_{\text{ClSize}} = m_{\text{pts}}$). Qualquer componente que seja menor do que o parâmetro m_{ClSize} não é interpretado como um *cluster* e é automaticamente descartado da hierarquia como ruído. Na Tabela 2 é apresentada a hierarquia produzida pelo HDBSCAN* para o conjunto de dados da Figura 3.

Tabela 2 – Hierarquia simplificada completa produzida pelo HDBSCAN* com $m_{\text{pts}} = m_{\text{ClSize}} = 3$ para o exemplo da Figura 3.

ϵ	$\mathbf{x}_3$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_8$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$	$\mathbf{x}_5$	$\mathbf{x}_4$	$\mathbf{x}_6$	$\mathbf{x}_7$
3.2650	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3.1828	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3
3.1623	0	2	2	2	2	2	2	2	2	2	2	3	3	3	3
2.0809	0	0	2	2	2	2	2	2	2	2	2	3	3	3	3
0.8544	0	0	0	4	4	4	4	4	5	5	5	3	3	3	3
0.8246	0	0	0	0	4	4	4	4	5	5	5	3	3	3	3
0.6403	0	0	0	0	0	4	4	4	5	5	5	3	3	3	3
0.6402	0	0	0	0	0	4	4	4	5	5	5	0	3	3	3
0.6325	0	0	0	0	0	4	4	4	0	0	0	0	3	3	3
0.6083	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborada pelo autor.

Hierarquias completas como aquela da Tabela 2 podem ser representadas de forma compactada em que apenas os pares consecutivos de níveis hierárquicos em que ocorrem uma divisão real de um *cluster* em dois ou mais *clusters* são armazenados, i.e., são descartados todos os níveis em que apenas objetos deixam de ser densos ou componentes espúrios surgem e passam a ser tratados como ruído, sem que ocorra a divisão de um *cluster* em novos *clusters* nesse nível. A Tabela 3 representa a hierarquia compactada do exemplo da Figura 3.

Tabela 3 – Hierarquia simplificada compactada gerada pelo HDBSCAN* com $m_{\text{pts}} = m_{\text{ClSize}} = 3$ para o exemplo da Figura 3.

ϵ	$\mathbf{x}_3$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_8$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$	$\mathbf{x}_5$	$\mathbf{x}_4$	$\mathbf{x}_6$	$\mathbf{x}_7$
3.2650	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3.1828	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3
2.0809	0	0	2	2	2	2	2	2	2	2	2	3	3	3	3
0.8544	0	0	0	4	4	4	4	4	5	5	5	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborada pelo autor.

Ao final, além de uma hierarquia completa ou compactada, o HDBSCAN* gera também uma árvore de grupos, como mostrado na Figura 4, formada apenas pelos *clusters* mais relevantes e, opcionalmente como pós-processamento, pode extrair dela uma partição plana a partir de uma medida de qualidade de *clusters* e de agrupamentos. Na próxima Seção, a estratégia de extração de *clusters* utilizada pelo HDBSCAN*, denominada FOSC, é discutida.

Figura 4 – Árvore de Grupos gerada a partir da hierarquia apresentada na Tabela 3.

Fonte: Elaborada pelo autor.

2.4 FOSC

Nesta seção a extração de uma partição plana a partir de cortes locais sobre as ramificações de uma árvore de grupos genérica é formulada como um problema de otimização. Em seguida, é apresentado o algoritmo FOSC (CAMPELLO *et al.*, 2013) capaz de resolver tal problema a partir de uma função objetivo que agrega sobre os valores de qualidades medidos por uma função de qualidade local dos grupos que compõem a solução. Diferente da maioria dos algoritmos não hierárquicos, que tentam produzir diretamente uma solução plana a partir dos dados, esse método consegue determinar não só a solução ótima com relação ao critério que é otimizado como também o número de *clusters* como um subproduto ao invés de um parâmetro de entrada explícito ou implícito. Tal método possui complexidade linear sendo assim escalável em relação ao número de grupos.

Seja $\{C_1, \dots, C_k\}$ o conjunto contendo todos os *clusters* de uma hierarquia de $\mathbf{X}$ da qual é necessário extrair uma partição plana. Suponha que tal hierarquia seja representada por uma árvore de grupos (*cluster tree*) T , como na Figura 5, em que cada nó representa um *cluster*. Seja C_1 a raiz da árvore que representa o *cluster* com todos os objetos inclusos, i.e., $C_1 = \mathbf{X}$. Para simplificar a explicação suponha, sem perda de generalidade, se tratar de uma árvore binária (os casos em que T não é binária são facilmente generalizáveis) em que cada nó representa o *cluster* C_i e seus nós filhos à esquerda e à direita são representados por C_{i_l} e C_{i_r} , respectivamente.

A partir de uma sequência de valores numéricos associados aos nós de T , que quantificam uma certa propriedade de interesse relacionada aos *clusters*, o problema de extração é dado pela escolha de uma coleção de nós de T que: (a) maximizem a soma dos valores associados; (b) de modo que os *clusters* representados pelos nós escolhidos formem uma partição válida do conjunto de dados $\mathbf{X}$. Dada uma função ou critério de qualidade Q de agrupamento que seja *aditiva* e *local* (conforme as definições a seguir), é possível determinar os valores de interesse/qualidade associados aos *clusters* individuais como as componentes correspondentes de Q .

Figura 5 – Exemplo de Árvore de Grupos: valores abaixo dos nós indicam uma medida de qualidade dos respectivos *clusters*.

Fonte: Elaborada pelo autor.

Definição 2.4.1 (Aditividade). Uma função de qualidade de agrupamento Q é dita aditiva se e somente se puder ser decomposta como a soma de componentes individuais, em que cada componente seja associado a um único *cluster* do agrupamento em questão.

Definição 2.4.2 (Localidade). Uma função de qualidade de agrupamento Q é dita local se e somente se a avaliação de qualquer de seus componentes, associado a um determinado *cluster*, não necessite de nenhuma informação a priori a respeito dos demais componentes, ou seja, pode ser calculada de forma independente de quais são os outros *clusters* que compõe o agrupamento em questão.

Ao decorrer do texto, são discutidas algumas funções de qualidade que possuem as propriedades definidas acima, mas por ora, basta supor que para cada *cluster* da árvore de grupos seja conhecido o valor numérico atribuído a ele, como mostrado na [Figura 5](#). Nesse exemplo, caso e.g. o *cluster* C_6 seja escolhido, os *clusters* C_8 , C_9 , C_3 e C_1 deixam de poder compor a solução, pois a escolha simultânea de C_6 e um desses *clusters* violaria a propriedade de que cada objeto não pode pertencer a mais de um *cluster* em uma partição válida. Entre as soluções que satisfazem as restrições citadas, pode-se verificar que C_2 , C_7 , C_8 e C_9 formam a solução ótima do problema, visto que maximizam os valores exibidos pela figura. Para encontrar tal solução de maneira eficiente, é interessante notar que a seleção dos *clusters* em uma subárvore é um subproblema idêntico ao original. Logo, pode-se adotar uma estratégia de programação dinâmica para, incrementalmente, resolver os subproblemas começando a partir dos nós folha e acumulando as soluções intermediárias enquanto se sobe pela árvore.

Começando dos nós folha no nível mais baixo da árvore de grupos, pode-se perceber que a soma dos valores associados a C_8 e C_9 (4.6) é maior do que o valor associado a C_6

(2.91), portanto $\mathbf{C}_6$ deve ser descartado para que $\mathbf{C}_8$ e $\mathbf{C}_9$ sejam escolhidos. Em seguida, o valor de $\mathbf{C}_3$ (6.75) é comparado à de $\mathbf{C}_7$, $\mathbf{C}_8$ e $\mathbf{C}_9$ (11.68), como o seu valor é menor, $\mathbf{C}_3$ é descartado em favor dos outros *clusters*. Agora compara-se $\mathbf{C}_2$ (5.34) ao valor acumulado de seus filhos $\mathbf{C}_4$ e $\mathbf{C}_5$ (4.39), por ser maior, $\mathbf{C}_2$ é escolhido como a solução de sua subárvore e seus filhos são descartados, restando apenas os *clusters* $\mathbf{C}_2$, $\mathbf{C}_7$, $\mathbf{C}_8$ e $\mathbf{C}_9$, que compõem a solução final (visto que $\mathbf{C}_1 = \mathbf{X}$ comumente não é tido como uma solução viável).

Ao realizar a seleção de *clusters* para compor uma partição plana, os *clusters* aninhados devem ser mutualmente exclusivos. Essa condição pode ser formulada como um conjunto de restrições para o problema de maximização da função de qualidade Q , responsável por avaliar a qualidade do agrupamento como uma composição dos componentes $Q(\mathbf{C}_i)$ referentes à qualidade de cada *cluster* $\mathbf{C}_i$ candidato a fazer parte da solução. Assim, esse problema de otimização pode ser escrito como:

$$\begin{aligned} \max_{\delta_2, \dots, \delta_k} \quad & \sum_{i=2}^k \delta_i Q(\mathbf{C}_i) \\ \text{s. a.} \quad & \delta_i \in \{0, 1\}, i = 1, \dots, k. \\ & \sum_{j \in I_h} \delta_j = 1, \forall h \text{ tal que } \mathbf{C}_h \text{ é um } \textit{cluster} \text{ folha.} \end{aligned} \tag{2.2}$$

em que k é o número total de *clusters* na árvore/hierarquia em questão, δ_i ($i = 2, \dots, k$) é um indicador que denota se o *cluster* $\mathbf{C}_i$ é escolhido para fazer parte da solução ($\delta_i = 1$) ou não ($\delta_i = 0$), e I_h é o conjunto de índices de todos os *clusters* pertencentes ao caminho entre $\mathbf{C}_h$ e $\mathbf{C}_1$, onde $\mathbf{C}_h$ é um nó folha. Essas restrições garantem que exatamente um *cluster* é selecionado em cada caminho que se inicia em um nó folha e sobe até a raiz da árvore, garantindo assim que cada objeto seja atribuído a apenas um *cluster*.

As árvores de grupo são produzidas por diferentes tipos de algoritmos que, por sua vez, se baseiam em diferentes tipos de paradigmas. Assim, uma medida de qualidade adequada depende do tipo de hierarquia a ser avaliada. Para dendrogramas ou hierarquias simplificadas (dado um parâmetro m_{ClSize}) produzidas por algoritmos hierárquicos tradicionais, bem como para aquelas produzidas por algoritmos hierárquicos baseados em densidade, [Campello et al. \(2013\)](#) propõem a *estabilidade S* que, assim como o conceito de tempo de vida de um grupo (*cluster lifetime*) ([JAIN; DUBES, 1988](#)), se baseia na premissa de que grupos mais relevantes persistem por mais tempo ao longo da hierarquia. O tempo de vida de um *cluster* em um dendrograma tradicional é definido como a diferença entre o nível na escala em que ele surge e o nível em que ele se divide. No entanto, a definição tradicional de tempo de vida de um *cluster* não leva em consideração a possibilidade que objetos possam deixar um *cluster* entre o seu surgimento e desaparecimento, o que não ocorre em dendrogramas tradicionais mas pode ocorrer em hierarquias simplificadas e hierarquias baseadas em densidade devido à modelagem de ruído e componentes espúrios. Em outras palavras, tal fenômeno pode ser causado pelo exclusão de *clusters* menores do

que um dado tamanho mínimo ($minClSize$) e/ou por objetos deixarem de ser considerados densos (caso a hierarquia tenha sido produzida por um algoritmo baseado em densidade). A estabilidade $S(\mathbf{C}_i)$ de um *cluster* $\mathbf{C}_i$ leva em conta os (possivelmente diferentes) tempos de vida de cada objeto em $\mathbf{C}_i$, sendo dada por:

$$S(\mathbf{C}_i) = \sum_{\mathbf{x}_j \in \mathbf{C}_i} lifetime(\mathbf{x}_j) \quad (2.3)$$

em que $lifetime(\mathbf{x}_j)$ é o tempo de vida em que o objeto $\mathbf{x}_j$ pertenceu a $\mathbf{C}_i$. A estabilidade total S de uma solução de agrupamento $\mathbf{P}$ com *clusters* selecionados a partir da árvore/hierarquia em questão é igual à soma das estabilidades individuais de cada *cluster*, isto é:

$$S = \sum_{\mathbf{C}_i \in \mathbf{P}} S(\mathbf{C}_i) \quad (2.4)$$

Para hierarquias tradicionais representadas como dendrogramas, onde não há modelagem de ruído ou componentes espúrios, a estabilidade de um *cluster* é igual ao seu tempo de vida multiplicado por sua cardinalidade (número de objetos). Já para hierarquias como as da [Tabela 1](#) e [Tabela 2](#), a estabilidade de um *cluster* é igual a soma dos tempos de vida de seus membros. Por exemplo, na [Tabela 2](#), de baixo para cima, o *cluster* $\mathbf{C}_4$ surge no nível 0.6083 (formado pelos objetos $\mathbf{x}_8$, $\mathbf{x}_{11}$ e $\mathbf{x}_{12}$) e desaparece ao se unir com o *cluster* $\mathbf{C}_5$, dando origem a $\mathbf{C}_2$ no nível 2.0809. Durante esse intervalo, outros dois objetos ($\mathbf{x}_9$ e $\mathbf{x}_{10}$) se unem a ele, nos níveis 0.8544 e 0.8246, respectivamente. Assim, sua estabilidade é dada por $S(\mathbf{C}_4) = (2.0809 - 0.6083) * 3 + (2.0809 - 0.8246) + (2.0809 - 0.8544) = 6.9006$.

Quando aplicada sobre hierarquias baseadas em densidade, a estabilidade está relacionada com o conceito estatístico de excesso de massa (*excess of mass*). Assim, ao invés de se utilizar ε diretamente, pode-se alternativamente utilizar $\frac{1}{\varepsilon}$ como uma estimativa aproximada de densidade, caso em que a estabilidade de um *cluster* passa a ser equivalente à sua massa relativa ([CAMPELLO et al., 2015](#)). Nesse caso, a massa relativa do *cluster* $\mathbf{C}_4$ no exemplo anterior seria dada por $S(\mathbf{C}_4) = (1/0.6083 - 1/2.0809) * 3 + (1/0.8246 - 1/2.0809) + (1/0.8544 - 1/2.0809) = 4.912092$.

REDES COMPLEXAS

Em 1736 o matemático Leonhard Euler demonstrou que era impossível atravessar todas as sete pontes da cidade de Königsberg passando por cada uma delas apenas uma vez. Ele percebeu que os elementos mais importantes para o problema eram as extremidades das pontes e a ligação entre essas extremidades, então ele resolve representar o problema de maneira abstrata utilizando para isso vértices e arestas. Surge assim a teoria dos grafos, tema inicial deste capítulo.

3.1 Conceitos Básicos

Desde os tempos de Euler até os dias atuais, diversos problemas complexos de interesse prático ou teórico podem ser representados utilizando-se de grafos. Um grafo, denotado por $\mathbb{G}(\mathbb{V}_{\mathbb{G}}, \mathbb{E}_{\mathbb{G}})$, é composto por um conjunto de vértices $\mathbb{V}_{\mathbb{G}} = \{v_1, v_2, \dots, v_N\}$ e um conjunto de arestas $\mathbb{E}_{\mathbb{G}} \subset \mathbb{V}_{\mathbb{G}} \times \mathbb{V}_{\mathbb{G}}$; em notação mais simplificada: $\mathbb{V} = \mathbb{V}_{\mathbb{G}}$, $\mathbb{E} = \mathbb{E}_{\mathbb{G}}$ e $\mathbb{G} = (\mathbb{V}, \mathbb{E})$.

A *ordem* de um grafo $\mathbb{G}$, representada por N , é dada pelo número de vértices que o compõem (i.e., a cardinalidade de $\mathbb{V}$, $|\mathbb{V}|$), enquanto o seu *tamanho*, ℓ , é dado pela quantidade de arestas existentes em $\mathbb{E}$ (i.e., $|\mathbb{E}|$). Além disso, $\mathbb{G}$ é dito *direcionado* se o seu conjunto de arestas $\mathbb{E}$ for formado por pares ordenados, i.e. $(v_i, v_j) \neq (v_j, v_i)$; caso contrário, $\mathbb{G}$ é um *grafo não direcionado*. Dada uma função $f: \mathbb{E} \mapsto \mathfrak{R}$ que atribui para cada aresta de $\mathbb{G}$ um valor real, tem-se que $\mathbb{G} = (\mathbb{V}, \mathbb{E}, f: \mathbb{E} \mapsto \mathfrak{R})$ é um *grafo ponderado*. Por último, se as definições anteriores forem generalizadas de maneira a permitir a ocorrência de laços (uma aresta conectando um vértice a si mesmo) e arestas múltiplas (duas ou mais arestas conectando o mesmo par de vértices), então $\mathbb{G}$ se torna um *multigrafo*.

Um grafo $\mathbb{G}' = (\mathbb{V}', \mathbb{E}')$ é um *subgrafo* de $\mathbb{G}$ se e somente se $\mathbb{V}' \subseteq \mathbb{V}$ e $\mathbb{E}' \subseteq \mathbb{E}$. Caso o grafo $\mathbb{G}'$ contenha todas as arestas de $\mathbb{G}$ que conectam pares de vértices de $\mathbb{V}'$, então

$\mathbb{G}'$ é um *subgrafo induzido* por $\mathbb{V}'$. Se o número de arestas ℓ de um grafo $\mathbb{G}$ qualquer se aproxima da (ou cresce proporcionalmente à) quantidade máxima de arestas, $N\frac{(N-1)}{2}$ para um grafo não direcionado e $N(N-1)$ para um grafo direcionado, então $\mathbb{G}$ é um *grafo denso*; caso contrário, $\mathbb{G}$ é um *grafo esparso*.

Computacionalmente, um grafo $\mathbb{G}$ pode ser representado por uma *matriz de adjacência*, uma *matriz de incidência* ou uma *lista de adjacência*. Cada representação traz consigo pontos positivos e negativos de modo que a representação mais adequada depende diretamente do objetivo da representação. Seja $\mathbb{A} = [a_{ij}]_{N \times N}$, em que $a_{ij} \in \{0, 1\}$, a matriz de adjacência do grafo $\mathbb{G}$. Cada elemento a_{ij} é igual a 1 se e somente se o vértice v_i for adjacente ao vértice v_j , caso contrário $a_{ij} = 0$. No caso de grafos direcionados, é possível que $a_{ij} \neq a_{ji}$, i.e., a relação de adjacência não é necessariamente recíproca, ou seja, a matriz de adjacência não é necessariamente simétrica. Para os grafos ponderados é definida uma segunda matriz $\mathbb{W} = [w_{ij}]_{N \times N}$, em que $w_{ij} \in \mathfrak{R}$ representa os pesos das arestas. Na maioria das vezes pode-se descartar a matriz de adjacência $\mathbb{A}$ e atuar apenas com a matriz $\mathbb{W}$, com isso cada elemento w_{ij} , além de armazenar o peso da aresta (v_i, v_j) , também representa sua existência ($w_{ij} \neq 0$) ou ausência ($w_{ij} = 0$). A matriz de incidência de um grafo qualquer é definida como $\mathbb{B} = [b_{ij}]_{N \times \ell}$, tal que a posição b_{ij} indica quantas das duas extremidades da j -ésima aresta incidem sobre o vértice v_i (usualmente 0 ou 1, mas possivelmente 2 no caso de um laço em um multigrafo). Já em uma lista de adjacência, cada vértice v_i possui uma lista com todos os vértices adjacentes a ele. As listas de adjacência são estruturas de dados mais eficientes do que as demais quando usadas para representar grafos esparsos.

Seja $\mathbb{E} = \mathbb{V} \times \mathbb{V}$, então, $\mathbb{G}$ é um *grafo completo* e recebe o nome de *clique*. Para representar cliques de diferentes ordens utiliza-se a notação K_n , em que n é a quantidade de vértices que compõem o clique.

Vizinhança de um vértice: Dado um vértice qualquer pertencente a $\mathbb{V}$, sua vizinhança é definida como o conjunto de todos os vértices adjacentes a ele.

$$\Gamma(v_i) = \{v_j | (v_i, v_j) \in \mathbb{E}\} \quad (3.1)$$

Grau de um vértice: O grau de um vértice v_i , denotado por k_i , é equivalente à cardinalidade de sua vizinhança $|\Gamma(v_i)|$, ou ainda, em notação matricial utilizando a matriz de adjacências $\mathbb{A} = [a_{ij}]_{N \times N}$:

$$k_i = \sum_j a_{ij} \quad (3.2)$$

Para os vértices pertencentes a grafos direcionados é possível definir graus de entrada e de saída conforme se segue.

Grau de entrada de um vértice: É dado pelo número de conexões que chegam até o vértice v_j :

$$k_j^{in} = \sum_i a_{ij} \quad (3.3)$$

Grau de saída de um vértice: Representa o número de conexões que saem do vértice v_i .

$$k_i^{out} = \sum_j a_{ij} \quad (3.4)$$

Apesar da [Equação 3.4](#) ser aparentemente equivalente à [Equação 3.2](#), é importante notar que a matriz de adjacência $\mathbb{A}$ de um grafo direcionado não é simétrica, como exposto anteriormente, de modo que, para um grafo direcionado, o grau de um vértice v_i é definido como $k_i = k_i^{in} + k_i^{out}$.

Também pode ser definido um tipo especial de grau para vértices de grafos ponderados que leve em consideração os pesos das arestas de $\mathbb{G}$.

Grau de força de um vértice: A força s_i de um vértice v_i é definida como a somatória dos pesos de todas as arestas incidentes sobre o mesmo. Para um grafo não direcionado tem-se:

$$s_i = \sum_j w_{ij} \quad (3.5)$$

Caminho entre dois vértices: Um caminho entre dois vértices quaisquer v_i e v_j é definido como uma sequência alternada de vértices e arestas

$$path(v_i, v_j) = (v_i, e_i, v_{i+1}, e_{i+1}, \dots, e_{j-2}, v_{j-1}, e_{j-1}, v_j) \quad (3.6)$$

em que cada aresta e_k pertencente ao caminho, incide sobre os vértices v_k e v_{k+1} que, respectivamente, a antecede e a sucede dentro da sequência. O caminho é dito *simples* se nenhum dos vértices se repetir ao longo da sequência. Quando um caminho começa e termina no mesmo vértice ele recebe o nome de *ciclo*. Um grafo que não possui ciclos é dito *acíclico*.

Caminho mínimo entre dois vértices: Em um grafo não ponderado, é o caminho com o menor número de arestas entre os dois vértices. Caso o grafo seja ponderado, é definido como o caminho cuja a soma dos pesos das arestas seja mínima.

Componente conexa: Uma componente conexa de $\mathbb{G}$ é um subgrafo $\mathbb{G}'$ induzido por $\mathbb{V}'$, em que

$$\forall v_i, v_j \in \mathbb{V}' \implies \exists path(v_i, v_j) \subset \mathbb{G}' \quad (3.7)$$

Árvore: Uma árvore é um tipo especial de grafo formado por apenas uma componente conexa e que não possui ciclos.

3.2 Medidas de Caracterização das Redes Complexas

As redes complexas podem ser consideradas como uma classe de grafo que apresenta diversas características interessantes e incomuns à maioria dos outros grafos. A busca pelo entendimento dessas características é essencial pois ajuda compreender tanto os fenômenos relacionados ao surgimento dessas redes quanto a compreender os processos dinâmicos que ocorrem sobre sua topologia. Algumas das propriedades dos grafos descritas anteriormente também são usadas na caracterização das redes complexas, porém, por motivos de espaço, não serão repetidas aqui. Por ser uma área de conhecimento que se iniciou fora da Teoria dos Grafos e se desenvolveu em paralelo à esta, possui seus próprios termos técnicos, que muitas vezes diferem daqueles utilizados na Matemática e na Ciência da Computação. Por isso, no restante deste Capítulo, será mantida a notação adotada até aqui, porém será dada preferência às terminologias (nós, conexões, redes, comunidades, etc.) vindas da área de Redes Complexas. A seguir são descritas algumas das medidas fundamentais para a caracterização e análise dessas redes.

Grau médio: É o valor médio dos graus dos nós (vértices) da rede:

$$\langle k \rangle = \frac{1}{N} \sum_i k_i \quad (3.8)$$

Na prática esse valor também pode ser obtido pela razão entre a *ordem* e o *tamanho* da rede multiplicado por dois.

$$\langle k \rangle = 2 \frac{\ell}{N} \quad (3.9)$$

O valor dois, presente na fórmula, corresponde ao fato de cada aresta conectar dois nós e por tanto, possuir duas extremidades.

Entropia: A entropia mede, de maneira global, o grau de “desordem” da rede, i.e. a heterogeneidade da rede.

$$h = - \sum_k p(k) \log[p(k)] \quad (3.10)$$

em que $p(k)$ é a probabilidade de um dos nós da rede possuir grau igual a k e é calculada a partir da divisão do número de nós com grau k pelo número de nós N .

Enquanto a entropia permite analisar globalmente a heterogeneidade da rede, também é interessante observar se existem correlações entre os nós conectados. A primeira vista, pode parecer razoável imaginar que um dado nó possua a mesma probabilidade de se conectar a qualquer outro, sem apresentar assim qualquer correlação. Porém, na prática, observa-se que em diversos casos reais, nós que compartilham determinadas características possuem chances maiores de apresentar uma conexão entre si. Com esse intuito, pode-se

utilizar a probabilidade conjunta $P(k, k')$ de ocorrer uma conexão entre dois nós de grau k e k' , ou ainda, a probabilidade condicional $P(k'|k)$ definida como:

$$P(k'|k) = \frac{\langle k \rangle P(k, k')}{k P(k')} \quad (3.11)$$

Note que $\sum_{k'} P(k'|k) = 1$ e caso a rede seja não dirigida (não direcionada), $P(k'|k) = P(k|k')$. Contudo, na prática, é difícil analisar diretamente tais probabilidades e comumente opta-se pela análise de correlação sobre medidas mais simples como o grau médio dos vizinhos mais próximos, descrito a seguir.

Grau médio dos vizinhos mais próximos: Mensura o grau médio dos nós adjacentes aos nós de grau k .

$$k_{nn}(k) = \sum_{k'} k' P(k'|k) \quad (3.12)$$

Quando não exista correlação entre os nós, $k_{nn}(k)$ é independente de k . Já se tal correlação existir, $k_{nn}(k)$ é uma função crescente quando nós com alto grau tendem a se conectar entre si, ou uma função decrescente quando nós com alto grau tendem a se conectar com nós de grau baixo.

Assortatividade: Redes em que nós com graus semelhantes tendem a apresentar conexão entre si são ditas *assortativas*, enquanto redes que apresentam o comportamento oposto são chamadas de *disassortativas*. As redes sociais são exemplos de redes que costumam ser assortativas, enquanto redes biológicas são exemplos de redes que tendem a ser disassortativas (NEWMAN, 2002). A partir do coeficiente de correlação de Pearson, a medida de assortatividade é definida como:

$$r = \frac{\frac{1}{\ell} \sum_{i,j} k_i k_j - \left[\frac{1}{\ell} \sum_{i,j} \frac{1}{2} (k_i + k_j) \right]^2}{\frac{1}{\ell} \sum_{i,j} k_i k_j \frac{1}{2} (k_i^2 + k_j^2) - \left[\frac{1}{\ell} \sum_{i,j} \frac{1}{2} (k_i + k_j) \right]^2} \quad (3.13)$$

Na fórmula acima, k_i e k_j são os graus dos nós da aresta $\{v_i, v_j\}$. Caso $r > 0$ então a rede em questão é assortativa, se $r < 0$ então a rede é disassortativa e caso $r = 0$ então a rede não apresenta correlação. É interessante notar que vários modelos de redes complexas como os de Erdős-Rényi (ERDÖS; RÉNYI, 1959), Watts-Strogatz (WATTS; STROGATZ, 1998) e Barabási-Albert (BARABÁSI; ALBERT, 1999) não apresentam correlação.

Transitividade: Também chamada de coeficiente de agrupamento global, mede a presença de triângulos (ciclos de tamanho três) em uma rede revelando a probabilidade de dois vizinhos de um nó estarem ou não conectados.

$$C = \frac{3N_{\Delta}}{N_3} \quad (3.14)$$

em que $3N_\Delta$ é o número de triângulos presentes na rede e N_3 o número de triplas conectadas.

Coefficiente de agrupamento local: Assim como a transitividade, mede o grau de aglomeração, porém de forma local. Sendo definido como a razão entre o número de conexões existentes entre dois nós quaisquer pertencentes à vizinhança de v_i e o número total de conexões possíveis.

$$cc_i = \frac{2e_i}{k_i(k_i - 1)} \quad (3.15)$$

em que e_i é o número de conexões entre os nós vizinhos de v_i e $\frac{1}{2}k_i(k_i - 1)$ é o número de conexões possíveis. É interessante notar que tanto coeficiente de agrupamento global quanto o coeficiente de agrupamento local estão limitados ao intervalo $[0, 1]$.

Grau de intermediação: O grau de intermediação (*betweenness centrality*) quantifica a importância de um nó baseando-se na quantidade de caminhos mínimos que passam por este, é definida como:

$$B_i = \sum_{jk} \frac{\sigma_{jk}(i)}{\sigma_{jk}} \quad (3.16)$$

em que σ_{jk} é o número de caminhos mínimos entre os nós v_j e v_k , e $\sigma_{jk}(i)$ é o número de caminhos mínimos entre v_j e v_k que passam por v_i .

Grau de centralidade: O grau de centralidade (*closeness centrality*) é definido como a distância média entre um nó qualquer v_i e todos os outros nós da rede.

$$Cl_i = \frac{N}{\sum_{j \neq i} d_{ij}} \quad (3.17)$$

em que d_{ij} é o comprimento do caminho mínimo entre os nós v_i e v_j e N é o número total de nós da rede.

Como exposto nessa seção, existem diversas medidas utilizadas para caracterizar uma rede complexa. Algumas das características únicas que as diferenciam de outros grafos comuns são: a distribuição dos graus dos seus nós segue uma lei de potência (*power law distribution*) (BARABÁSI; ALBERT, 1999; CLAUSET; SHALIZI; NEWMAN, 2009); a maioria dos caminhos mínimos possuem poucos nós intermediários (*small-world phenomenon*) (WATTS; STROGATZ, 1998); o diâmetro das redes diminuem (*shrinking diameter*) e a proporção de conexões em relação ao número de nós aumenta (*densification*) conforme novos nós são inseridos na rede (LESKOVEC; KLEINBERG; FALOUTSOS, 2005).

3.3 Modularidade e Estrutura de Comunidades

Apesar de úteis, as medidas definidas na seção anterior levam em consideração apenas a rede como um todo (macro-escala) ou cada um de seus elementos individualmente (micro-escala), deixando assim de considerar os diferentes grupos (*clusters*) presentes na maioria dessas redes. A compreensão de tais grupos pode levar ao melhor entendimento da organização interna das redes (NEWMAN, 2003; RAVASZ; BARABÁSI, 2003). Esse entendimento é fundamental para a compreensão de fenômenos associados a elas, como os processos dinâmicos que ocorrem sobre suas topologias. Por esse motivo, a investigação e o desenvolvimento de algoritmos capazes de encontrar tais grupos é um problema de fundamental importância. Para isso, é necessário antes definir formalmente o que são grupos, contudo, tal definição é subjetiva e comumente apenas alguns conceitos gerais do que constitui um grupo são aceitos. O principal deles diz respeito ao fato da estrutura interna das redes complexas possuir subgrafos altamente conexos quando comparados ao restante da rede. Nesse contexto, um grupo seria um subgrafo da rede que possui mais arestas internas do que arestas que ligam seus nós aos outros nós da rede. Tais grupos são comumente chamados de *comunidades* em razão dos diversos trabalhos publicados na literatura em áreas como sociologia, que investigam o tema há décadas e, em especial, em razão das aplicações em análise de redes sociais na web, que têm se tornado muito populares nos últimos anos (FORTUNATO, 2010).

3.3.1 Medidas de Qualidade

Assim como ocorre com o problema de agrupamento de dados, exposto no Capítulo 2, o problema de detecção de comunidades em redes complexas também é NP-hard, sendo inviável enumerar e avaliar cada uma das partições possíveis. Por esse motivo, faz-se necessário o uso de heurísticas capazes de encontrar boas soluções aproximadas (vide Equação 2.1) e de alguma medida de qualidade capaz de diferenciar entre soluções boas e ruins.

Um exemplo de função de qualidade é a *performance*, definida como a razão entre a quantidade de pares de nós adjacentes pertencentes à mesma comunidade mais a quantidade de pares de nós não adjacentes pertencentes à comunidades diferentes divididos pela quantidade de conexões possíveis.

$$Q_{perf}(P) = \frac{|(v_i, v_j) \in \mathbb{E}, C_i = C_j| + |(v_i, v_j) \notin \mathbb{E}, C_i \neq C_j|}{N(N-1)/2} \quad (3.18)$$

Nessa fórmula, N é o número total de nós, $\mathbb{E}$ é o conjunto de conexões existentes na rede, C_i e C_j são as comunidades do nó v_i e do nó v_j , respectivamente, e (v_i, v_j) representa a conexão entre os nós v_i e v_j . Essa equação é válida apenas para redes não ponderadas, mas sua generalização para redes que possuem peso é simples. Os valores que $Q_{perf}(P)$ pode

assumir estão contidos no intervalo $[0, 1]$. Valores altos indicam que os grupos que formam a partição são internamente densos e externamente esparsos. Essas são características desejáveis para uma solução, logo, quanto mais alto o valor de $Q_{perf}(P)$, melhor a solução. No entanto, como as redes complexas tendem a serem esparsas, o segundo termo da soma, $|(v_i, v_j) \notin \mathbb{E}, C_i \neq C_j|$, pode dominar a equação, gerando valores altos mesmo quando as partições são ruins.

Outro exemplo de função de qualidade é a *coverage*, que é definida como a razão entre o número de conexões internas às comunidades e o número total de conexões presentes na rede.

$$Q_{cov}(P) = \frac{|(v_i, v_j) \in \mathbb{E}, C_i = C_j|}{|(v_i, v_j) \in \mathbb{E}|} \quad (3.19)$$

Assim como o *performance*, a *coverage* assume valores no intervalo $[0, 1]$, valores baixos indicam soluções ruins enquanto valores altos indicam soluções boas. Como essa medida de qualidade considera apenas a separação entre as comunidades, ela acaba atribuindo valores altos para partições contra-intuitivas; por exemplo, uma partição formada por duas comunidades em que uma delas possui apenas um único nó ligado por uma única aresta à outra comunidade, que possui todos os demais nós (e, conseqüentemente, todas as outras conexões), terá valor máximo.

A medida de qualidade mais utilizada na detecção de comunidades em redes complexas é a *modularidade* Q (NEWMAN; GIRVAN, 2004). Ela se baseia no fato de grafos aleatórios não possuírem estruturas de grupos (i.e. subgrafos altamente conexos), então a existência de comunidades pode ser testada a partir da diferença entre o número de arestas de cada subgrafo, que representa uma possível comunidade, e o valor esperado de conexões desse subgrafo se os nós da rede se conectassem aleatoriamente. Especificamente, a modularidade Q pode ser definida como:

$$Q = \frac{1}{2\ell} \sum_{ij} (a_{ij} - P_{ij}) \delta(C_i, C_j) \quad (3.20)$$

em que ℓ é o número de conexões existentes na rede, a_{ij} é o elemento (i, j) da matriz de adjacência da rede ($\mathbb{A}$), e $\delta(C_i, C_j)$ é uma função indicadora igual a 1 caso $C_i = C_j$ ou 0 caso contrário. O termo $\frac{1}{2}$ elimina a duplicidade na contagem das conexões e P_{ij} é a esperança matemática do número de conexões entre os nós v_i e v_j se os nós da rede se conectam segundo um modelo aleatório, denominado *modelo nulo*. Desta forma, Q é claramente interpretada como uma média das diferenças entre o número observado e o número esperado de conexões entre nós de um mesmo grupo (subgrafo candidato a comunidade) segundo o modelo nulo adotado.

A princípio, qualquer modelo nulo pode ser utilizado para determinar P_{ij} . No

entanto, alguns modelos de grafos aleatórios não são bons representantes para redes complexas na prática, por não possuírem várias das características usualmente presentes em redes complexas reais, tais como as características descritas na [Seção 3.2](#). Por isso, normalmente utiliza-se o modelo de configuração (*configuration model*), que gera um grafo aleatório com a mesma sequência de graus. Nesse modelo, atribui-se k_i pontas (*stubs*) a cada nó v_i , e uma conexão é gerada sorteando-se aleatoriamente duas pontas quaisquer para se conectarem. A probabilidade de um nó v_i ter uma de suas pontas escolhidas é $\frac{k_i}{2\ell}$, assim, a probabilidade p_{ij} de dois nós se conectarem ao menos uma vez (dado que o modelo permite laços e arestas múltiplas) é $\frac{k_i k_j}{4\ell^2}$. Logo, o número esperado de conexões é $P_{ij} = 2\ell p_{ij} = \frac{k_i k_j}{2\ell}$, que substituindo na [Equação 3.20](#) resulta em:

$$Q = \frac{1}{2\ell} \sum_{ij} \left(a_{ij} - \frac{k_i k_j}{2\ell} \right) \delta(C_i, C_j) \quad (3.21)$$

Agregando os elementos da [Equação 3.21](#) a partir dos grupos a que cada nó pertence, obtém-se:

$$Q = \sum_i \left[\frac{e_i}{\ell} - \left(\frac{d_i}{2\ell} \right)^2 \right] \quad (3.22)$$

em que e_i é o número de conexões internas ao grupo C_i e d_i é a soma dos graus dos nós de C_i .

SELEÇÃO DE GRUPOS A PARTIR DE HIERARQUIAS: UMA MODELAGEM BASEADA EM GRAFOS

Em análise de agrupamento de dados, um problema clássico que tem sido estudado há décadas por estatísticos e, mais recentemente também por cientistas da computação e cientistas de dados, é a extração de soluções de agrupamento planas (i.e. partições) a partir de uma hierarquia de agrupamento. Como principal contribuição desta dissertação, propõe-se uma abordagem totalmente original para este problema, que é inovadora em dois aspectos distintos. Primeiro, ao contrário das abordagens tradicionais encontradas na literatura, que se limitam a buscar uma solução dentre o conjunto de partições representadas pelos diferentes níveis hierárquicos da hierarquia de agrupamento em questão, a solução aqui proposta utiliza uma técnica proposta recentemente como um arcabouço genérico para extração de soluções possivelmente compostas por sub-soluções obtidas via cortes locais (não-horizontais) em diferentes níveis hierárquicos, soluções essas que nunca poderiam ser encontradas pelos métodos tradicionais (que operam realizando cortes horizontais globais a todo um único nível hierárquico). Tal arcabouço, entretanto, demanda uma medida de qualidade para cada grupo, que deve satisfazer determinadas propriedades teóricas. O segundo aspecto inovador da principal proposta desta dissertação é uma medida de qualidade de grupos que satisfaz tais propriedades ao modelar o problema utilizando resultados da teoria de grafos e redes complexas, preliminarmente discutidos no Capítulo 3.

4.1 Motivação

No Capítulo 2 discutiu-se os dois principais paradigmas de agrupamento de dados, a saber, agrupamento particional e agrupamento hierárquico. Em agrupamento hierárquico uma solução de agrupamento é representada por uma árvore que descreve relações hierárquicas entre *clusters* e *sub-clusters* aninhados (JAIN; DUBES, 1988; EVERITT; LANDAU; LEESE, 2001). Este paradigma de agrupamento é interessante por razões variadas, a principal delas sendo o fato de que variados fenômenos reais podem ser descritos por distribuições de probabilidade multi-modais tais que modos dominantes geram *clusters* compostos de múltiplos sub-modos que se manifestam amostralmente como *sub-clusters*. Além disso, modelos hierárquicos são ferramentas poderosas de visualização de dados multi-dimensionais, tais como na forma de um dendrograma ou árvore simplificada de grupos, conforme já discutido no Capítulo 2.

Por serem capazes de descrever relações hierárquicas de grupos e sub-grupos em diferentes níveis de granularidade, por exemplo regiões e sub-regiões densas do espaço de dados segundo diferentes limiares de densidade, modelos hierárquicos de agrupamento provêm descrições mais ricas dos dados do que aquelas proporcionadas por modelos de agrupamento plano, que buscam por partições (ao invés de hierarquias) em que cada objeto da base de dados recebe um único rótulo de grupo (possivelmente nulo, representando “ruído”). Apesar disso, são comuns as aplicações em que o analista também deseja (ou apenas deseja) uma solução plana de agrupamento, além de (ou ao invés de) uma hierarquia. Por exemplo, enquanto biólogos, historiadores ou arqueólogos podem estar genuinamente interessados em relações hierárquicas entre espécies, documentos, ou artefatos, analistas de mercado podem apenas estar interessados em um conjunto bem definido de nichos disjuntos de consumidores para posterior análise, interpretação e tomada de decisão. Soluções planas de agrupamento também são comumente utilizadas como etapa de pré-processamento em certas tarefas de reconhecimento de padrões, tais como segmentação de imagens para posterior classificação de objetos.

É importante notar que, mesmo quando se deseja uma solução de agrupamento plana, é usualmente vantajoso primeiramente obter uma representação hierárquica dos dados para somente posteriormente extrair a solução plana desejada a partir da hierarquia. As razões são múltiplas. A principal delas é que uma hierarquia descreve os dados em diferentes níveis de granularidade ou densidade, que podem ser vistos como diferentes graus de especificidade ou generalidade, o que proporciona meios para a exploração de uma miríade de possíveis soluções candidatas a partir de uma visão global das estruturas de grupos presentes nos dados. Conforme será discutido neste capítulo, isso permite avaliar e quantificar diferentes aspectos de cada *cluster* candidato, não como uma fotografia estática daquele *cluster* em um nível de granularidade/especificidade/densidade particular, mas de acordo com o seu comportamento e relação a outros *clusters* dos dados ao

longo de múltiplos níveis hierárquicos. No caso de algoritmos baseados em densidade hierárquicos, como HDBSCAN* (discutido no Capítulo 2, Seção 2.3.3) (CAMPELLO *et al.*, 2015), isso proporciona uma oportunidade única de detectar *clusters* a partir de diferentes níveis de densidade, o que não é possível com algoritmos baseados em densidade do tipo particional, tais como, por exemplo, DBSCAN (ESTER *et al.*, 1996), DBSCAN* (CAMPELLO; MOULAVI; SANDER, 2013) e DENCLU (HINNEBURG; KEIM *et al.*, 1998). Outra vantagem da extração de soluções planas a partir de hierarquias é que, ao contrário de muitos algoritmos particionais como *k-Means* (MACQUEEN, 1967), *Expectation Maximization* (EM) (BISHOP, 2006) e tantos outros, não é preciso que o usuário defina *a priori* a quantidade de grupos.

Tradicionalmente, a abordagem para extração de soluções planas a partir de hierarquias de agrupamento tem sido por décadas o denominado “corte horizontal” global do dendrograma, que é nada mais do que a seleção de um dos níveis da hierarquia, correspondente a uma determinada quantidade de *clusters*. Em casos em que é aceitável a participação direta de um analista no processo, esta escolha pode ser feita manualmente em bases de dados com estruturas de grupos bem definidas, caso em que o dendrograma exhibe claramente os *clusters* mais prominentes, viabilizando a realização manual do corte. Em muitos casos práticos, entretanto, deseja-se realizar o corte de forma automatizada, ou porque a aplicação em questão não presume a participação humana, ou seja, deve ser completamente automática, ou porque o usuário não possui a experiência e conhecimento necessário para interpretar o dendrograma e realizar a escolha manualmente, ou ainda porque a estrutura de grupos não é clara o suficiente para que a escolha possa ser feita com confiança meramente com base na interpretação visual, mesmo que por um analista experiente, pois a interpretação visual está sujeita a fatores subjetivos.

Para que a escolha não esteja sujeita a fatores subjetivos e possa ser feita de forma automática, é preciso quantificar de forma objetiva os critérios que se deseja considerar ao realizar um corte, o que é tradicionalmente feito através de alguma estatística aplicada aos diferentes níveis hierárquicos de um dendrograma. A escolha do nível hierárquico mais apropriado segundo alguma estatística e o corte horizontal global de dendrogramas naquele nível é um problema clássico em análise de agrupamentos que tem sido investigado há décadas (JAIN; DUBES, 1988), especialmente pelos estatísticos (MILLIGAN; COOPER, 1985). Apesar de sua ampla utilização tanto na literatura como na prática de agrupamento de dados, esta abordagem possui uma limitação severa que é a impossibilidade de proporcionar soluções compostas com *clusters* descritos em diferentes níveis hierárquicos, ou seja, em diferentes níveis de abstração ou granularidade (BOUDAILLIER; HÉBRAIL, 1997; BOUDAILLIER; HÉBRAIL, 1998). O exemplo na Figura 6 ilustra tal limitação.

Figura 6 – Corte global em dendrograma e sua limitação: à esquerda tem-se um conjunto de dados e à direita tem-se um dendrograma de agrupamento desses dados com um exemplo de corte horizontal global que retorna uma solução plana com três *clusters*, A, B e C. Não há corte global que retorne A, B, D e E simultaneamente.

Fonte: Elaborada pelo autor.

Na Figura 6 à esquerda tem-se um conjunto de dados e à direita tem-se um dendrograma de agrupamento desses dados em *clusters* e *sub-clusters*, bem como um exemplo de corte horizontal global (linha pontilhada) que retorna uma solução plana com três *clusters*, A, B e C. Não há corte global que retorne simultaneamente os *clusters* A, B, D e E numa mesma solução. Note que não há razão para crer que esta solução também não possa ou deva ser vista como uma solução candidata válida a se considerar.

Para contornar tal limitação, os autores em (CAMPELLO *et al.*, 2013) propuseram o método FOSC (*Framework for Optimal Selection of Clusters*), discutido anteriormente no Capítulo 2, Seção 2.4. Conforme já discutido em detalhes naquela seção, dada uma árvore de *clusters* e uma função objetivo que agrega valores de qualidade individual de cada *cluster* candidato a ser inserido na solução, o método opera um algoritmo de programação dinâmica simples, efetivo e extremamente eficiente que percorre a árvore de baixo para cima comparando localmente a qualidade de cada nó (*cluster*) não folha da árvore com seus nós filhos (*sub-clusters*) e propagando as decisões acima na árvore para compor uma solução global e globalmente ótima em tempo linear.

No caso do dendrograma da Figura 6, cada nó (junção) do dendrograma representa um *cluster* candidato e o dendrograma representa a árvore de *clusters* em questão. Para bases de dados com muitos objetos, a quantidade de *clusters* candidatos em um dendrograma pode ser elevada. Como alternativa, pode-se simplificar o dendrograma em uma árvore simplificada de *clusters*, por exemplo com a especificação de um tamanho mínimo dos *clusters* que devem ser considerados relevantes. Conforme já discutido no Capítulo 2, o uso de um parâmetro m_{CISize} permite especificar um tamanho mínimo tal que qualquer *cluster* menor do que m_{CISize} seja automaticamente descartado como um componente es-

púrio ou *ruído* (CAMPELLO *et al.*, 2013). Aplicando este conceito no dendrograma da Figura 6, por exemplo com $m_{\text{ClSize}} = 5$, é simples observar que apenas os *clusters* A, B, D, E, $C = D \cup E$ e $F = A \cup B$ seriam considerados como candidatos válidos, logo a árvore de grupos simplificada poderia ser representada como $\mathbf{X}(C(D,E),F(A,B))$, onde $\mathbf{X}$ representa o conjunto de dados completo (raiz da árvore, i.e., $F \cup C$). Dada uma medida de qualidade individual para os grupos, caso o valor agregado desta medida para os grupos D e E seja maior quando esses grupos estão separados do que o valor correspondente à união desses grupos, FOSC fará a seleção de D e E em detrimento de $C = D \cup E$. Similarmente, caso o valor agregado desta medida para os grupos A e B seja maior quando esses grupos estão separados do que o valor correspondente à união desses grupos, FOSC fará a seleção de A e B em detrimento de $F = A \cup B$. Se a união de D, E, A e B ($\mathbf{X}$ nesse exemplo em particular) fosse um candidato válido porém com valor de qualidade inferior ao valor de qualidade agregado de D, E, A e B como grupos separados, FOSC manteria D, E, A e B como solução final. Essa solução seria correspondente a fazer um corte local no ramo esquerdo dendrograma de forma a separar E e D como *clusters* individuais, e outro corte mais acima, no ramo direito do dendrograma, de forma a separar A e B como *clusters* individuais. A Figura 7 apresenta os cortes locais realizados pelo FOSC. Somente com o uso de tais cortes locais, em diferentes níveis para diferentes ramos da hierarquia, todas as soluções planas válidas (i.e., que não violam as propriedades de partição) presentes em uma árvore de grupos podem ser exploradas.

Figura 7 – Cortes locais em diferentes níveis do dendrograma: a partir de cortes locais é possível encontrar uma solução plana composta pelos *clusters* A, B, D, E da Figura 6.

Fonte: Elaborada pelo autor.

Note que a seleção dos *clusters* realizada por FOSC depende da função objetivo Q que quantifica a qualidade das soluções de agrupamento planas candidatas e que se

deseja maximizar. Conforme foi discutido no Capítulo 2, Seção 2.4, a função objetivo precisa ser tal que possa ser decomposta em valores numéricos individuais $Q(\mathbf{C}_i)$ que quantificam a qualidade de cada *cluster* $\mathbf{C}_i$. Neste caso, FOSC irá: (a) maximizar a soma dos valores associados; (b) de modo que os *clusters* escolhidos formem uma partição válida do conjunto de dados $\mathbf{X}$. Mas para que a solução obtida seja garantidamente um ótimo global da função objetivo em questão, é preciso que (CAMPELLO *et al.*, 2013): (i) ela seja *aditiva* (Definição 2.4.1), o que significa que pode ser escrita como a soma de componentes individuais referentes a cada *cluster* presente na partição $\mathbf{P}$ sendo avaliada, i.e., $Q(\mathbf{P}) = \sum_{\mathbf{C}_i \in \mathbf{P}} Q(\mathbf{C}_i)$; e (ii) ela seja *local* (Definição 2.4.2), que significa que os valores de qualidade individuais $Q(\mathbf{C}_i)$ possam ser calculados localmente apenas com informações referentes ao *cluster* $\mathbf{C}_i$, sem que se precise saber quais são os demais *clusters* que irão compor o agrupamento $\mathbf{P}$ em questão (o que impediria a solução em tempo linear e tornaria o problema combinatório).

Como o conceito de cortes locais e o método FOSC (que permite a implementação prática deste conceito de forma eficiente) são muito recentes, especialmente quando comparados a décadas de pesquisa em estatísticas para cortes horizontais globais, até o momento são quase inexistentes medidas de qualidade que satisfazem as propriedades acima e que tenham sido propostas. Basicamente, a única medida já consolidada na literatura, no sentido que tem sido investigada em detalhes e provada experimentalmente ser efetiva, é a medida de Estabilidade proposta no próprio artigo original em que se introduziu FOSC (Equações 2.3 e 2.4), bem como a especialização desta medida para o caso de hierarquias baseadas em densidade, também denominada *Relative Excess of Mass*, que é utilizada pelo HDBSCAN* (discutida ao final da Seção 2.4).¹

É importante notar que, mesmo num contexto de aprendizado supervisionado, como classificação de padrões, recomenda-se o uso de diferentes medidas de avaliação da qualidade de um modelo, tais como acurácia, precisão, *recall*, F-Measure, e área sob a curva ROC, porque diferentes medidas refletem diferentes aspectos de um modelo ou solução (TAN; STEINBACH; KUMAR, 2005). Utilizar uma única medida pode levar a resultados indesejados, como por exemplo a escolha de um péssimo classificador que só possui alta acurácia porque prioriza a classe majoritária em um problema envolvendo classes altamente desbalanceadas (TAN; STEINBACH; KUMAR, 2005).

No contexto não-supervisionado, que é o caso de agrupamento de dados, o problema é ainda mais crítico, pois não se dispõe de rótulos para a validação dos modelos ou resultados obtidos, ou seja, não se dispõe de um *ground truth*. De fato, conforme colocado na monografia clássica de JAIN; DUBES:

¹ Nessa dissertação, por simplicidade de notação, chamaremos a medida utilizada por HDBSCAN* simplesmente de *Estabilidade*, que é o nome mais geral independente da natureza da hierarquia em questão.

“The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.” (JAIN; DUBES, 1988).

Medidas de avaliação da qualidade de agrupamentos ou de grupos no contexto não supervisionado, que utilizam apenas os dados e a informação dos grupos para quantificar a qualidade desses, são denominadas de *critérios internos de validação* (ou de validade, ou ainda de avaliação) de agrupamentos (JAIN; DUBES, 1988). Quando esses critérios podem ser utilizados para comparar quantitativamente a qualidade de duas soluções de agrupamento ou de dois grupos de forma relativa, esses critérios também são denominados de *critérios relativos de validação* (VENDRAMIN; CAMPELLO; HRUSCHKA, 2010).

Como a própria definição de *cluster* admite diferentes interpretações e portanto certa subjetividade, diferentes critérios de validação existem que refletem diferentes aspectos de uma solução ou grupo candidato (VENDRAMIN; CAMPELLO; HRUSCHKA, 2010). Mais ainda do que no contexto supervisionado (classificação) mencionado anteriormente, é bem conhecido na literatura de agrupamento de dados que não se deve confiar exclusivamente em um critério de validação único, mas ao invés utilizar mais de um critério num contexto de análise exploratória (BEZDEK; PAL, 1998; JASKOWIAK *et al.*, 2016). Neste contexto, certos estudos da literatura vão além de simplesmente propor um novo critério a ser considerado junto a outros existentes, mas têm sido capazes de identificar cenários nos quais determinados critérios são mais ou menos apropriados (VENDRAMIN; CAMPELLO; HRUSCHKA, 2010), o que pode guiar o usuário ou analista na escolha de critérios a serem utilizados em uma aplicação em particular.

Note que a medida de Estabilidade utilizada por FOSC é nada mais que um critério de validade interno e relativo de agrupamento que satisfaz as propriedades de aditividade e localidade necessárias para a aplicação do algoritmo. Infelizmente, praticamente todos os critérios de validação interna e relativa bem conhecidos na literatura de agrupamento se baseiam em relações de compactação e separabilidade entre os *clusters* e, muito embora muitos deles satisfaçam a propriedade de aditividade — e.g. o famoso critério da Silhueta (KAUFMAN; ROUSSEEUW, 2005; VENDRAMIN; CAMPELLO; HRUSCHKA, 2010) e o recente critério DBCV para agrupamentos baseados em densidade (MOULAVI *et al.*, 2014), eles não satisfazem a propriedade de localidade. Isso impede a utilização desses critérios em FOSC.

O desenvolvimento de um novo critério de avaliação de grupos e agrupamentos, que satisfaz as propriedades de aditividade e localidade requeridas por FOSC, é proposto na próxima seção como principal contribuição deste trabalho. No Capítulo 5 será realizada e analisada uma ampla gama de experimentos no sentido de buscar caracterizar os cenários em que a nova medida proposta tende a se comportar de forma mais (ou menos) efetiva

que a medida de Estabilidade, de forma que ambas possam estar à disposição do analista como medidas complementares (mais do que alternativa uma à outra).

4.2 Modularidade Q Ponderada como Critério de Qualidade de Agrupamento em FOSC

Historicamente, um grande número de conceitos teóricos e algoritmos em análise de agrupamento de dados se baseiam direta ou indiretamente, explícita ou implicitamente, em teoria de grafos. Exemplos clássicos são os algoritmos de agrupamento hierárquicos Single-Linkage e Complete-Linkage descritos no Capítulo 2, que podem ser interpretados respectivamente como buscas sucessivas por componentes conexos e cliques em grafos e, no caso do primeiro, pode na verdade ser implementado algorítmicamente via métodos de construção de árvores geradoras mínimas (MST) (JAIN; DUBES, 1988). Conforme discutido na Seção 2.3.3, o próprio algoritmo HDBSCAN* é baseado no cálculo da árvore geradora mínima, *MST_r*, sobre o grafo de alcançabilidade mútua dos objetos de dados (Definição 2.3.7). Não apenas a teoria dos grafos clássica, mas também conceitos modernos da área de redes complexas, como detecção de comunidades em redes sociais (vide Capítulo 3), também têm sido utilizados em análise de agrupamento de dados (ARRUDA; COSTA; RODRIGUES, 2012).

E assim como grafos e redes complexas podem ser utilizados para modelar problemas em agrupamento de dados, técnicas originalmente da área de agrupamento de dados também podem ser adaptadas para modelar e resolver problemas em grafos, tais como a detecção de comunidades em redes complexas. No caso de avaliação/validação de grupos e agrupamentos, a conexão entre essas áreas é em ambas as direções. Por exemplo, critérios de validação de agrupamento de dados têm sido adaptados e estendidos para a avaliação da qualidade de grupos e comunidades em redes complexas (RABBANY *et al.*, 2013), bem como modelos baseados em grafos têm sido utilizados para a validação de agrupamento de dados (MOULAVI *et al.*, 2014).

A observação de que existe uma forte sinergia entre as áreas de agrupamento de dados e teoria dos grafos, em particular detecção de comunidades em redes complexas, motivou a principal hipótese de pesquisa desta dissertação, apresentada no Capítulo 1 e reformulada aqui:

HIPÓTESE DE PESQUISA: *Em problemas de agrupamento de dados tradicionais, descritos por vetores de atributos (ou matrizes de distância), é possível adaptar o critério da Modularidade Q de forma que este possa ser utilizado como medida de qualidade de grupos no método FOSC para extrair soluções de agrupamento superiores, em determinados cenários, àquelas obtidas com o critério da Estabilidade.*

Conforme será discutido mais adiante neste texto, a modelagem do problema de seleção de agrupamentos FOSC utilizando grafos envolverá um grafo ponderado. Logo, o critério que será utilizado para a investigação da hipótese de pesquisa acima será uma versão ponderada da Modularidade Q, que é originalmente definida para grafos não ponderados, conforme visto no Capítulo 3. Esta versão é apresentada a seguir.

4.2.1 Modularidade Q Ponderada

A medida da Modularidade Q vista no Capítulo 3, Equação 3.22, considera que o grafo em questão é não ponderado, ou seja, que as arestas não possuem pesos, o que é equivalente a dizer que todas as arestas possuem um mesmo peso unitário. Para grafos ponderados, pode-se utilizar uma generalização daquela equação que leva em conta os diferentes pesos das arestas. Na literatura de detecção de comunidades, a seguinte versão da Modularidade Q Ponderada é comumente utilizada (FENG *et al.*, 2007):²

$$Q_s(\mathbf{P}) = \sum_{i=1}^k \left[\frac{IS(\mathbf{C}_i)}{TS} - \left(\frac{DS(\mathbf{C}_i)}{TS} \right)^2 \right] \quad (4.1)$$

em que $\mathbf{P} = \{\mathbf{C}_1, \dots, \mathbf{C}_k\}$ é uma partição dos vértices do grafo em subconjuntos disjuntos $\mathbf{C}_i$, denominados comunidades ou *clusters*, k é o número total de *clusters*, TS é a soma dos pesos de todas as arestas do grafo, $IS(\mathbf{C}_i)$ é a soma dos pesos de todas as arestas internas ao *cluster* $\mathbf{C}_i$ (ou seja, arestas cujos ambos os vértices finais pertencem a $\mathbf{C}_i$) e $DS(\mathbf{C}_i)$ é a soma dos pesos de todas as arestas que possuem ao menos um vértice em $\mathbf{C}_i$ (i.e., a soma dos graus de força dos vértices do *cluster* $\mathbf{C}_i$ — vide definição do *grau de força* de um vértice no Capítulo 3, Equação 3.5).

Proposição 4.1. A Modularidade Q Ponderada da Equação 4.1 satisfaz às propriedades de aditividade e localidade das Definições 2.4.1 e 2.4.2, respectivamente, que são requeridas pelo arcabouço FOSC.

Demonstração. Note que a Equação 4.1 pode ser rescrita como $Q_s(\mathbf{P}) = \sum_{i=1}^k Q(\mathbf{C}_i)$, onde:

$$Q(\mathbf{C}_i) = \left[\frac{IS(\mathbf{C}_i)}{TS} - \left(\frac{DS(\mathbf{C}_i)}{TS} \right)^2 \right] \quad (4.2)$$

Logo, $Q_s(\mathbf{P})$ é claramente aditiva com relação aos componentes $Q(\mathbf{C}_i)$ e, para completar a prova, precisamos portanto mostrar que: (a) cada componente $Q(\mathbf{C}_i)$ está associado apenas ao *cluster* $\mathbf{C}_i$ e (b) o cálculo deste componente não depende de quais são os demais *clusters* da partição.

² Essa é a versão utilizada, por exemplo, nos bem conhecidos algoritmos de detecção de comunidades SCAN (XU *et al.*, 2007) e gSkeletonClu (SUN *et al.*, 2010; HUANG *et al.*, 2013).

Note que, para um dado grafo, o termo TS é uma constante (soma das arestas do grafo), logo não depende da partição $\mathbf{P}$ sendo avaliada ou qualquer de seus componentes em particular. O termo $IS(\mathbf{C}_i)$ é a soma de todas as arestas internas ao *cluster* $\mathbf{C}_i$, logo refere-se apenas a $\mathbf{C}_i$ e o seu cálculo envolve apenas vértices e arestas internas a $\mathbf{C}_i$. O termo $DS(\mathbf{C}_i)$ é a soma de todas as arestas que possuem ao menos um vértice em $\mathbf{C}_i$, logo este termo também é determinado única e exclusivamente por quem são os vértices em $\mathbf{C}_i$. Embora $DS(\mathbf{C}_i)$ envolva arestas conectando $\mathbf{C}_i$ a vértices que não fazem parte de $\mathbf{C}_i$ e, portanto, necessariamente pertencem a outros *clusters*, a informação de quais são esses outros *clusters* não é necessária para o cálculo deste termo. Logo, $IS(\mathbf{C}_i)$, $DS(\mathbf{C}_i)$ e, consequentemente, $Q(\mathbf{C}_i)$, podem ser calculados *localmente* ao *cluster* $\mathbf{C}_i$, sem a necessidade de saber quais são os demais *clusters* presentes na partição.

Em resumo, $Q_s(\mathbf{P})$ na Equação 4.1 é dada por uma soma de componentes $Q(\mathbf{C}_i)$, cada um dos quais pode ser computado individualmente e localmente a cada um dos k *clusters* $\mathbf{C}_i$ da partição $\mathbf{P}$ sendo avaliada. $\square$

A Proposição 4.1 é importante porque, dada uma árvore de *clusters* obtida a partir de algum tipo de algoritmo hierárquico (e.g. HDBSCAN*), a partir da qual se quer extrair uma solução plana de agrupamento através do método FOSC, então: caso seja possível representar os dados de maneira apropriada através de um grafo, pode-se avaliar individualmente a qualidade $Q(\mathbf{C}_i)$ de cada um dos *clusters* candidatos da hierarquia como vértices desse grafo e aplicar o algoritmo de extração FOSC sobre esses valores de qualidade para selecionar o conjunto de *clusters* que representam cortes locais na hierarquia tal que a partição plana correspondente maximiza globalmente a Modularidade Ponderada da Equação 4.1. Para isso, entretanto, é preciso modelar os objetos da base de dados através de um grafo que represente as relações de similaridade entre os objetos de forma apropriada à aplicação da equação da modularidade ponderada.

4.2.2 Grafo de Similaridade

Suponha inicialmente que se tem um grafo não ponderado e que se deseja calcular a similaridade entre dois vértices u e v deste grafo com base na topologia do grafo. Uma forma clássica de calcular tal similaridade é a denominada *similaridade estrutural* (FENG *et al.*, 2007):

$$S(u, v) = \frac{|\Gamma_u \cap \Gamma_v|}{\sqrt{|\Gamma_u| |\Gamma_v|}} \quad (4.3)$$

em que $\Gamma_{(\cdot)}$ é a vizinhança do vértice, ou seja, o conjunto dos seus vértices adjacentes. Evidentemente, se u e v possuem exatamente os mesmos vértices adjacentes (incluindo ou não um ao outro), então $|\Gamma_u| = |\Gamma_v|$ e $S(u, v) = 1$. Por outro lado, se u e v não possuem nenhum vértice adjacente em comum, então $|\Gamma_u \cap \Gamma_v| = 0$ e $S(u, v) = 0$.

Note que, em detecção de comunidades, essa medida é tipicamente utilizada para atribuir pesos às arestas de um grafo de forma que esses pesos reflitam a similaridade estrutural entre vértices adjacentes desse grafo, ou seja, a medida adiciona pesos a arestas existentes do grafo, sem alterar a sua topologia (XU *et al.*, 2007).

Em (SUN *et al.*, 2010; HUANG *et al.*, 2013), os autores adaptam essa similaridade para levar em conta os pesos originais das arestas em um grafo ponderado, como segue:

$$\sigma(u, v) = \frac{\sum_{x \in \Gamma_u \cap \Gamma_v} w(u, x) w(v, x)}{\sqrt{\sum_{x \in \Gamma_u} w^2(u, x)} \sqrt{\sum_{x \in \Gamma_v} w^2(v, x)}} \quad (4.4)$$

em que $w(\cdot, \cdot)$ é o peso da respectiva aresta. Esses pesos indicam a força da conexão entre os pares de vértices correspondentes. Numa rede social, por exemplo, quanto maior o valor de $w(u, v)$ mais forte é a relação entre os indivíduos u e v . Note que nesta equação, vértices x conectados fortemente a ambos u e v terão mais peso no cálculo da similaridade do que vértices x que, embora sejam compartilhados pelas vizinhanças de u e v , possuem uma ou ambas as conexões $w(x, u)$ e/ou $w(x, v)$ fraca(s). Evidentemente, no caso particular de grafos não ponderados (arestas com pesos unitários), a Equação 4.4 recai na Equação 4.3.

Analogamente à similaridade estrutural da Equação 4.3, a similaridade da Equação 4.4 também é tipicamente utilizada apenas para ajustar os pesos das arestas de um grafo ponderado, sem alterar sua topologia (SUN *et al.*, 2010; HUANG *et al.*, 2013). De fato, em redes complexas reais não é improvável que vértices compartilhem um ou mais vizinhos em comum, logo, se uma aresta fosse inserida no grafo para cada par de vértices (u, v) tal que $\sigma(u, v)$ é não nulo, um número muito grande de arestas tenderiam a ser inseridas, distorcendo significativamente a estrutura original do grafo.

Aqui neste trabalho, propõe-se adaptar o conceito acima, originalmente oriundo da área de detecção de comunidades em redes sociais, para o contexto de agrupamento de dados. Em particular, a ideia central é construir um grafo de similaridade entre os objetos da base de dados tal que se possa aplicar a transformação 4.4 e, então, usar o grafo transformado resultante (G_σ) para o cálculo dos componentes de modularidade na Equação 4.2, tal que FOSC possa ser aplicado.

Para tal, parte-se de um grafo completo de dissimilaridades entre os pares de objetos da base de dados, G_d . Tal grafo completo pode corresponder simplesmente à matriz de distância aos pares utilizada para realizar o agrupamento dos dados. Em se tratando de distâncias, i.e. dissimilaridade, é preciso antes de mais nada transformar os valores em similaridade, pois os termos $w(\cdot, \cdot)$ em 4.4 se tratam de similaridades. Isso pode ser feito fazendo $s = 1 - (d/d_{max})$, onde d é um valor de dissimilaridade original, d_{max} é o maior dentre todos os valores de dissimilaridade (maior valor da matriz de distância, ou equivalentemente, maior aresta do grafo G_d), e s é o valor de similaridade resultante. Como se trata de uma transformação linear, tal conversão de distâncias em similaridades

apenas modifica a escala, mas não distorce os valores originais em termos relativos.

O grafo completo de similaridade resultante (G_s), entretanto, não é apropriado para aplicação da Modularidade Q, pois esta medida analisa as relações entre as arestas intra e entre *clusters*, o que perde o sentido se todos os pares de vértices são adjacentes. Mesmo no caso da modularidade ponderada, que leva em conta os pesos das arestas e, portanto, é capaz de dar menos importância a arestas conectando objetos distantes no espaço dos dados, cujos pesos são pequenos no grafo de similaridades, a grande quantidade de tais arestas com pesos individualmente pequenos no grafo completo, quando em conjunto, podem ter efeito significativo e indesejado no cálculo da modularidade.

Para evitar este problema, propõe-se neste trabalho utilizar um subgrafo gerador do grafo completo de similaridade, $G_{KNN} \subset G_s$, tal que apenas as arestas na vizinhança KNN de cada vértice de G_s é mantida. Em outras palavras, as K maiores arestas incidentes a cada um dos vértices do grafo são mantidas, enquanto aquelas que não fazem parte do conjunto das K maiores arestas incidentes a nenhum vértice são removidas.

A escolha de K será discutida mais adiante neste capítulo. Por hora, é importante observar que a transformação 4.4 será aplicada ao grafo G_{KNN} apenas para ajustar os pesos das arestas deste grafo, mantendo a sua topologia. O grafo resultante, G_σ , é o grafo que será utilizado como base para o cálculo dos termos da Modularidade Q para os *clusters* candidatos no método FOSC.

4.2.2.1 Resumo do Método Proposto

Em resumo, os passos para a construção do grafo de similaridade e cálculo dos termos da modularidade Q referentes aos *clusters* candidatos para uso em FOSC são:

1. Considere a matriz de distâncias entre os objetos da base de dados que foi utilizada para realizar o agrupamento hierárquico dos dados a partir do qual se deseja extrair uma solução plana utilizando o método FOSC e interprete essa matriz como um grafo completo de distâncias, G_d . Tome o maior valor dessas distâncias, d_{max} , e transforme todas distâncias d da matriz em similaridade fazendo $s = 1 - (d/d_{max})$. Interprete a matriz de similaridade resultante como um grafo completo de similaridades, G_s .
2. Obtenha um subgrafo gerador de G_s , G_{KNN} , removendo as arestas que não estão dentre as K maiores arestas incidentes a nenhum dos vértices de G_s , ou equivalentemente, mantendo apenas as arestas que estejam dentre as K maiores incidentes a ao menos um vértice.
3. Aplique a transformação 4.4 para ajustar os pesos das arestas de G_{KNN} , e denomine o grafo ajustado resultante de G_σ .

4. Para cada *cluster* candidato $\mathbf{C}_i$ na árvore de *clusters*, calcule o valor da componente correspondente da modularidade, $Q(\mathbf{C}_i)$, utilizando a Equação 4.2 sobre o grafo G_σ .
5. Aplique FOSC sobre a hierarquia de grupos utilizando $Q(\mathbf{C}_i)$ como valor de qualidade de cada grupo $\mathbf{C}_i$.

4.2.3 Caso Particular: Agrupamento HDBSCAN*

No caso de algoritmos baseados em densidade, em especial HDBSCAN*, algumas considerações devem ser feitas, conforme discutido abaixo nesta seção.

4.2.3.1 Grafo de Alcançabilidade Mútua

Algoritmos como DBSCAN e algoritmos relacionados, incluindo HDBSCAN*, operam no espaço transformado de distâncias de alcançabilidade. HDBSCAN* em particular, que constitui o foco principal deste trabalho por se tratar de um algoritmo estado-da-arte hierárquico e, portanto, passível de utilização com FOSC, opera sobre o espaço transformado das distâncias de alcançabilidade mútua (*mutual reachability distances* — Definição 2.3.6). Embora nenhum grafo precise ser explicitamente construído, operar em tal espaço transformado é conceitualmente equivalente a operar sobre o grafo completo de distâncias de alcançabilidade mútua ($G_{m_{pts}}$ — Definição 2.3.7). No caso de HDBSCAN*, são essas distâncias que são utilizadas no Passo 1 do algoritmo da seção anterior (Seção 4.2.2.1), ou seja, G_d no Passo 1 corresponde a $G_{m_{pts}}$ no caso de HDBSCAN*.

4.2.3.2 Modelagem de Ruído como Singletons

Operar no espaço transformado de distâncias de alcançabilidade, ao invés do espaço original dos dados (e.g. Euclideano), torna os respectivos algoritmos, incluindo HDBSCAN*, capazes de detectar objetos como sendo ruído, ao mesmo tempo que são robustos a este ruído. Nesse caso, duas questões relevantes são: (a) como tratar objetos rotulados como ruído durante o cálculo da modularidade Q e extração de *clusters* via FOSC? e (b) o critério da modularidade como aqui proposto, para avaliação da qualidade de grupos e agrupamentos, não seria tendencioso a favorecer soluções que rotulam maiores quantidades de ruído, no caso extremo induzindo a solução óbvia de agrupamento onde cada objeto é deixado isolado de todos os demais como um grupo por si só?

A resposta para ambas as perguntas acima é simples. No caso de objetos rotulados como ruído, esses não podem ser simplesmente ignorados no cálculo da modularidade, caso contrário não se penalizaria soluções de agrupamento triviais em que apenas uma pequena fração de objetos é rotulada como grupos que parecem artificialmente compactos e bem separados simplesmente porque a maioria dos objetos da base de dados ao redor deles foi rotulada como ruído. Como solução para este problema, deve-se ao invés interpretar um

ruído como um objeto que não foi agrupado com nenhum outro, ou seja, que foi deixado isolado pelo algoritmo, o que equivale a considerar objetos ruído como *singletons* (i.e. um *cluster* com um único objeto).

Logo, numa hierarquia de agrupamento baseada em densidade, como aquelas produzidas por HDBSCAN*, se um *cluster* $\mathbf{C}_i$ se divide (na visão *top-bottom* da hierarquia) em dois *sub-clusters* $\mathbf{C}_{il}$ e $\mathbf{C}_{ir}$ de tal forma que nem todos os objetos em $\mathbf{C}_i$ estão em $\mathbf{C}_{il}$ ou $\mathbf{C}_{ir}$, porque alguns dos objetos de $\mathbf{C}_i$ se tornam ruído (e não fazem parte portanto de nenhum dos *sub-clusters* resultantes da divisão), cada um desses objetos é visto como um *cluster* por si só. Logo, digamos que 3 objetos em $\mathbf{C}_i$ não fazem parte de nenhum dos *sub-clusters* $\mathbf{C}_{il}$ e $\mathbf{C}_{ir}$ de $\mathbf{C}_i$, então na árvore de grupos correspondente, que será utilizada por FOSC, o nó pai $\mathbf{C}_i$ não deve possuir apenas 2 nós filhos $\mathbf{C}_{il}$ e $\mathbf{C}_{ir}$, mas sim 5 nós filhos (i.e., $\mathbf{C}_{il}$, $\mathbf{C}_{ir}$ e 3 outros nós, cada um correspondente a um dos três *singletons*).

É interessante observar que esta abordagem naturalmente desencoraja a rotulação indiscriminada de objetos como ruído, evitando a tendência indesejada conjecturada na questão (b) formulada anteriormente. De fato, observe que, na Equação 4.2, para um grupo $\mathbf{C}_i$ com um único objeto, ou seja, um *singleton*, o termo $IS(\mathbf{C}_i)$ é nulo, enquanto $DS(\mathbf{C}_i)$ não é. Logo, como $DS(\mathbf{C}_i)$ é positivo e está sendo subtraído na equação, um *singleton* pode somente decrementar o valor da modularidade.

4.2.3.3 Escolha da Vizinhança KNN

No algoritmo da Seção 4.2.2.1, existe um parâmetro K que determina a vizinhança a ser considerada para a construção do subgrafo esparso de similaridade a partir do grafo denso original (grafo completo). A aplicação do método proposto em hierarquias de agrupamento geradas por HDBSCAN* é particularmente conveniente porque este algoritmo já estabelece a noção de vizinhança através do seu parâmetro m_{pts} . É difícil argumentar em favor de se utilizar um tamanho de vizinhança m_{pts} durante o agrupamento dos dados e outro tamanho de vizinhança K diferente para a construção do grafo de similaridade que será utilizado como base para avaliar a qualidade dos *clusters* resultantes. Logo, para manter consistente a noção de vizinhança no agrupamento e na avaliação dos grupos resultantes, faz-se $K = m_{pts}$.

Note que não é necessário envolver qualquer custo computacional adicional para encontrar as $K = m_{pts}$ arestas relevantes de cada um dos vértices, pois essa informação pode ser coletada durante a própria execução de HDBSCAN*, que precisa visitar todas as arestas durante a execução do método Prim de construção da árvore geradora mínima no espaço transformado de distâncias de alcançabilidade mútua (*MSTr*).

4.2.4 Análise de Complexidade

Para compreender a complexidade computacional do HDBSCAN* com as melhorias apresentadas neste trabalho, observa-se que o algoritmo possui as seguintes fases: (i) obtenção da hierarquia de grupos e do grafo gerador; (ii) cálculo da modularidade e (iii) processo de extração dos *clusters*.

A complexidade computacional para obter uma hierarquia de grupos é dada, no HDBSCAN*, por $O(N^2)$ (CAMPELLO *et al.*, 2015). É importante salientar, como discutido na seção anterior, que o grafo transformado resultante, G_σ , é construído ao mesmo tempo em que é construída a MST_r do algoritmo HDBSCAN*. A obtenção do grafo não modifica o custo computacional da construção da hierarquia de grupos.

O processo de cálculo da modularidade também não possui complexidade maior que a da geração da hierarquia de grupos, isto é, sua complexidade, de maneira conservadora, é de $O(N^2)$, conforme discutido a seguir. Os componentes TS (soma total dos pesos das arestas) e $DS(\mathbf{C}_i)$ (soma de todas as arestas que possuem uma extremidade em $\mathbf{C}_i$), em que $\mathbf{C}_i$ é um *singleton*, podem ser obtidos durante a construção de G_σ , sem nenhum custo adicional. Para obter o valor de $DS(\mathbf{C}_i)$ nos casos em que $\mathbf{C}_i$ não é um *singleton*, basta somar o $DS(\mathbf{C}_i)$ de seus *clusters* filhos, processando os nós da árvore de grupos de baixo para cima a partir das folhas, com uma complexidade total de ordem linear com o número de *clusters* na árvore. No pior caso, a árvore será binária e cada nó folha conterá apenas um único objeto, caso em que o número de *clusters* na árvore será $2N - 1$, onde N é o número de objetos. Logo, no pior caso, o cálculo dos termos $DS(\mathbf{C}_i)$ para todos os *clusters* será $O(N)$.

Na presente análise, supõe-se que seja utilizada uma lista de adjacências para armazenar o grafo G_σ , em que o acesso a um vértice é realizado em tempo constante (i.e. $O(1)$). Por construção, cada vértice deste grafo possui $K = m_{\text{pts}}$ vizinhos. Dado que o parâmetro m_{pts} é uma constante pequena, ou seja, $m_{\text{pts}} \ll N$, o grafo G_σ é esparso e, portanto, as arestas e vértices adjacentes a qualquer vértice podem ser visitados em tempo constante neste grafo. Dado um conjunto de *clusters* elegíveis $\{\mathbf{C}_2, \dots, \mathbf{C}_k\}$, utiliza-se como estrutura de dados auxiliar uma matriz de dimensões $N \times k$ que possibilita determinar, em tempo constante $O(1)$, se um dado vértice v pertence a um *cluster* $\mathbf{C}_i$ qualquer. Desta forma, o componente $IS(\mathbf{C}_i)$ de um *cluster* $\mathbf{C}_i$ pode ser calculado em tempo $O(m_{\text{pts}} \times |\mathbf{C}_i|)$, pois é preciso apenas visitar cada um dos objetos (vértices) pertencentes ao *cluster* e verificar quais dos seus vértices adjacentes em G_σ também pertence ou não ao mesmo *cluster*. Assim, para um nível qualquer da hierarquia com h *clusters*, tem-se um custo de $O(m_{\text{pts}} \times (|\mathbf{C}_1| + |\mathbf{C}_2| + \dots + |\mathbf{C}_h|))$, que é igual a $O(m_{\text{pts}} \times N)$ para qualquer nível da hierarquia porque cada nível da hierarquia deve necessariamente envolver todos os objetos da base de dados. No pior caso, essa hierarquia possui um número de níveis igual a N , o que resulta numa complexidade de $O(m_{\text{pts}} \times N^2)$. Como $m_{\text{pts}} \ll N$, a complexidade total

será de ordem $O(N^2)$.

Vale ressaltar que, normalmente, a hierarquia compacta do HDBSCAN* é muito menor do que o tamanho do conjunto de dados. Além disso, caso um mesmo *cluster* $\mathbf{C}_i$ apareça em mais de um nível da hierarquia, o componente $IS(\mathbf{C}_i)$ é calculado, na verdade, apenas uma vez, o que reduz expressivamente o número de operações necessárias.

Uma suposição alternativa para o cálculo da complexidade do $IS(\mathbf{C}_i)$ ocorre caso seja fornecido ao FOSC a árvore de *clusters*. Neste cenário, considerando o conjunto de *clusters* elegíveis $\{\mathbf{C}_2, \dots, \mathbf{C}_k\}$, o número de operações para o cálculo de IS será dado pela somatória das cardinalidades de cada um desses *clusters*, resultando na complexidade de ordem $O(t)$, em que $t = m_{pts} \times \sum_{i=2}^k |\mathbf{C}_i|$. Como m_{pts} é uma constante e, na prática, árvores simplificadas de grupos tendem a ser tal que $t \ll N^2$, a complexidade computacional do algoritmo é limitada superiormente por $O(N^2)$ de forma conservativa também neste cenário.

RESULTADOS

Neste capítulo são discutidos os resultados experimentais do método proposto tanto em bases de dados sintéticas como em bases de dados reais.

5.1 Configuração dos Experimentos

5.1.1 Bases de Dados

5.1.1.1 Conjuntos de Dados Sintéticos

Para a realização de experimentos controlados, onde se conhece a função de densidade de probabilidade que deu origem aos dados amostrados e, portanto, a distribuição real dos *clusters* que se procura encontrar, nós utilizaremos a bem conhecida coleção *benchmark* de conjuntos de dados para agrupamento disponibilizada pelo grupo da Dra. Julia Handl da Universidade de Manchester (HANDL; KNOWLES, n.d.).¹ Esta coleção consiste de 160 conjuntos de dados sintéticos gerados a partir de geradores de dados capazes de produzir *clusters* segundo distribuições Normais multivariadas (HANDL; KNOWLES, 2007).

Os conjuntos de dados nessa coleção possuem quantidades variadas de *clusters*, a saber, $k = 4, 10, 20$ e 40 . O tamanho (quantidade de objetos) de cada *cluster* foi determinado aleatoriamente no intervalo $[50, 500]$ para os conjuntos com 4 e 10 grupos e no intervalo $[10, 100]$ para os conjuntos com 20 e 40 grupos.

Metade (80) dos conjuntos de dados foram gerados utilizando distribuições normais multivariadas (Gaussianas alongadas a partir de matrizes de covariância pré-definidas) em um espaço com 2 e 10 dimensões. O gerador utilizado nessa subcoleção contém parâmetros que permitem a criação de grupos alongados com orientações arbitrárias. No entanto,

¹ Disponível em <http://personalpages.manchester.ac.uk/staff/Julia.Handl/generators.html>

pelas razões discutidas em (HANDL; KNOWLES, n.d.), o gerador não tem capacidade de produzir grupos com as mesmas características em uma dimensão maior que 10, por esta razão essa subcoleção possui conjuntos de dados com no máximo 10 dimensões.² Para cada combinação de dimensões (2 ou 10) e número de *clusters* (4, 10, 20 ou 40), tem-se 10 conjuntos de dados, totalizando 80 conjuntos de dados. Essa subcoleção será denominada *Coleção Sintética 1*.

A outra metade (80) dos conjuntos de dados foi produzida com base em um segundo gerador, que cria grupos em forma de elipsoides, definindo arbitrariamente a orientação do eixo mais longo do elipsoide e gerando pontos segundo uma distância com distribuição Gaussiana do eixo principal, buscando minimizar sobreposição com outros *clusters*. Dado que este gerador permite a geração de dados em alta dimensionalidade, os respectivos conjuntos de dados possuem 50 e 100 dimensões. Para cada combinação de dimensões (50 ou 100) e número de *clusters* (4, 10, 20 ou 40), tem-se 10 conjuntos de dados, totalizando 80 conjuntos de dados. Essa subcoleção será denominada *Coleção Sintética 2*.

5.1.1.2 Conjuntos de Dados Reais

No presente trabalho utilizamos conjuntos de dados reais com diferentes características (tais como número de objetos, número de atributos e número de categorias) oriundas de diferentes domínios, tais como análise de dados de expressão gênica (3 bases), análise de texto (4 bases) e imagens (2 coleções de bases), além de dados obtidos do repositório UCI (LICHMAN, 2013) (5 bases). A Tabela 4 apresenta um sumário dos dados com as respectivas características (número de objetos, número de atributos e número de classes), bem como as medidas de dissimilaridade aplicadas no experimento.³ Essas bases de dados e a escolha das respectivas medidas de dissimilaridade são as mesmas utilizadas nas publicações originais dos algoritmos HDBSCAN* e FOSC (CAMPELLO; MOULAVI; SANDER, 2013; CAMPELLO *et al.*, 2013; CAMPELLO *et al.*, 2015).

As duas últimas linhas da Tabela 4 representam coleções de bases de dados ao invés de bases individuais. Os dados dessas coleções foram extraídos da *Amsterdam Library of Object Images* (ALOI) (GEUSEBROEK; BURGHOUTS; SMEULDERS, 2005) e processados por Horta e Campello (2012). Esses dados foram adquiridos por requisição aos autores. Os conjuntos de dados foram gerados de forma aleatória, selecionando k diferentes categorias de imagens 100 vezes para cada valor de k ($k = 2, 3, 4$, and 5) e, logo após, amostrando sem substituição, de cada vez, 25 imagens para cada uma das k categorias selecionadas. A seleção produziu 400 conjuntos, cada um contendo 2, 3, 4, ou 5 grupos e 50, 75, 100, ou 125 imagens (objetos). As imagens foram representadas utilizando 6 tipos

² Resumidamente, as distâncias entre os objetos se tornam menos significativas e os grupos se tornam mais esféricos com o aumento da dimensionalidade.

³ Note que em bases de expressão gênica e texto utilizou-se respectivamente Pearson e Cosseno ao invés de distância Euclideana, que são as escolhas padrão nesses domínios de aplicação.

Tabela 4 – Coleção de dados reais utilizados durante os experimentos.

	Dados	#Obj.	#Attr.	#Cl.	#dist
Cell cycle 237	(YEUNG <i>et al.</i> , 2001)	237	17	4	Pearson
Cell cycle 384	(YEUNG <i>et al.</i> , 2001)	384	17	4	Pearson
Yeast galactose	(YEUNG; MEDVEDOVIC; BUMGARNER, 2003)	205	81	4	Pearson
Wine	(LICHMAN, 2013)	178	13	3	Euclideana
Glass	(LICHMAN, 2013)	244	10	6	Euclideana
Iris	(LICHMAN, 2013)	150	5	3	Euclideana
Ecoli	(LICHMAN, 2013)	336	7	8	Euclideana
WDBC	(LICHMAN, 2013)	569	32	2	Euclideana
Articles-1442-5	(NALDI <i>et al.</i> , 2011)	253	4636	5	Cosseno
Articles-1442-80	(NALDI <i>et al.</i> , 2011)	253	388	5	Cosseno
Cbrilpirivson	(PAULOVICH <i>et al.</i> , 2008)	945	1431	5	Cosseno
Cbrilpirson	(PAULOVICH <i>et al.</i> , 2008)	675	1423	4	Cosseno
ALOI-TS88	(HORTA; CAMPELLO, 2012)	*	88	*	Euclidiana
ALOI-PCA	(HORTA; CAMPELLO, 2012)	*	6	*	Euclidiana

Fonte: Elaborada pelo autor.

de descritores: momentos de cor (144 atributos), estatísticas de textura extraídas da matriz de coocorrência de níveis de cinza (88 atributos), histograma de borda de Sobel (128 atributos), estatísticas de primeira ordem do histograma de níveis de cinza (6 atributos), propriedades da *run-length matrix* de níveis de cinza (44 atributos) e histograma de níveis de cinza (256 atributos) (HORTA; CAMPELLO, 2012). Os resultados obtidos utilizando as estatísticas de textura são apresentados como “ALOI-TS88”. Uma representação com 6 dimensões que combina a primeira componente principal extraída de cada um dos seis descritores por meio da PCA é representada por “ALOI-PCA”.

5.1.2 Avaliação dos Resultados

Os resultados serão avaliados utilizando o índice externo de validação de agrupamento *Adjusted Rand Index* — ARI (HUBERT; ARABIE, 1985), que é uma versão ajustada por chance do clássico índice de *Rand* e que têm se tornado escolha padrão na literatura para avaliação externa de resultados de agrupamento de dados frente a uma solução de referência (*ground truth*). Assim como usualmente feito na literatura, objetos rotulados como ruído são tratados como *singletons* (*cluster* com um único elemento) para fins do cálculo do índice.

5.1.3 Algoritmo e Parâmetros

Todos os experimentos são executados utilizando o algoritmo HDBSCAN* sem o parâmetro m_{ClSize} , que é opcional e não será aqui utilizado (o que, em termos práticos,

seria equivalente a configurar $m_{\text{CLSize}} = 1$). O único parâmetro do algoritmo, m_{pts} , será variado no intervalo entre 4 e 40, mais precisamente $m_{\text{pts}} = 4, 8, 12, \dots, 40$. Para bases de dados envolvendo não mais que alguns poucos milhares de objetos, como aquelas utilizadas nos experimentos deste trabalho, essa faixa compreende os valores usuais nos principais trabalhos da literatura envolvendo algoritmos que possuem este mesmo parâmetro, tais como DBSCAN, OPTICS, SSDSCAN, HISSCLU e HDBSCAN* (GERTRUDES *et al.*, n.d.).

5.1.4 Baseline

O algoritmo HDBSCAN* representa conceitualmente o estado-da-arte em agrupamento de dados e tem se tornado a escolha *de facto* em termos de algoritmos baseados em densidade na literatura, com implementações muito eficientes já disponíveis nas principais bibliotecas de linguagens tais como R (disponível, por exemplo, no pacote R “dbscan”) e Python (disponível, por exemplo, na biblioteca “scikit-learn”). Na prática, o algoritmo já foi mostrado experimentalmente ser superior a vários algoritmos bem estabelecidos (CAMPELLO *et al.*, 2015; GERTRUDES *et al.*, n.d.). Neste trabalho, a medida originalmente utilizada pelo algoritmo para avaliação de qualidade de *clusters*, a medida de Estabilidade (*Stability*) discutida anteriormente no Capítulo 2, será utilizada aqui como *baseline*, de forma que os resultados da extração de soluções planas de agrupamento a partir da hierarquia HDBSCAN* utilizando FOSC com esta medida serão comparados com aqueles obtidos utilizando a medida proposta neste trabalho (Modularidade Q baseada no grafo de similaridade KNN, denominada aqui Modularidade KNN ou *Mod. knn* por simplicidade).

5.1.5 Teste Estatístico

Para analisar os resultados de forma global nós utilizamos dois tipos de análise estatística. O primeiro tipo, mais simples, consiste na contagem do número vitórias, empates e derrotas que uma medida obtém em relação a outra medida dado os conjuntos de dados sob análise. O segundo é o teste pareado de Wilcoxon (WILCOXON, 1945), também denominado *Wilcoxon signed-rank test*, que é um teste não paramétrico que ranqueia as diferenças em performance de duas medidas para cada conjunto de dados, comparando os ranques para as diferenças positivas e negativas. As diferenças são ranqueadas de acordo aos seus valores absolutos. A média dos ranques é computada em caso de empates. O intervalo de confiança é selecionado com base no mínimo entre os ranques positivos e negativos. Por fim, a definição de rejeição / aceitação da hipótese nula (hipótese de que as medidas possuem desempenhos semelhantes) é realizada com base no valor obtido a partir do intervalo de confiança. Utilizou-se o pacote R “scmamp” (CALVO; RODRIGO, 2016) que segue precisamente a descrição do teste em (DEMSAR, 2006).

5.2 Análise dos Resultados

5.2.1 Resultados para Bases Sintéticas

A Tabela 5 apresenta os resultados do ARI em função de m_{pts} para as bases de dados da Coleção Sintética 1 (Misturas de Gaussianas Multivariadas 2D e 10D). Valores em verde ou vermelho indicam respectivamente que a medida correspondente à respectiva linha obteve resultado melhor ou pior que a medida concorrente. Valores em preto indicam que ambas as medidas obtiveram o mesmo resultado (empate). Células destacadas em amarelo indicam o melhor de todos os resultados (dentre as diferentes medidas e valores de m_{pts}) para cada base.

Observe na Tabela 5 que os resultados foram subdivididos para os conjuntos de dados com 4 ou 10 *clusters* (Caso 1) e 20 ou 40 *clusters* (Caso 2). Com esta divisão, fica evidente que, para esta coleção, que envolve apenas conjuntos de dados de baixa dimensionalidade, a medida original de Estabilidade domina os resultados para os conjuntos de dados contendo quantidades maiores de *clusters* (Caso 2). Já para os conjuntos de dados com menores quantidades de *clusters* (Caso 1), os resultados são mais equilibrados, com alguma superioridade da nova medida proposta (Mod. knn), mais proeminentemente para valores menores de m_{pts} (até 12). Se apenas os melhores resultados obtidos para cada base de dados para os diferentes valores de m_{pts} forem considerados (células em amarelo), há uma evidente divisão dos resultados, com a Estabilidade alcançando os melhores resultados em todas as bases do Caso 2 e a Mod. knn apresentando os melhores resultados em todas as bases do Caso 1.

Os resultados da Tabela 5 também são ilustrados na Figura 8, onde as bases do Caso 1 correspondem às subfiguras da esquerda, enquanto as bases do Caso 2 correspondem às subfiguras da direita.

Tabela 5 – Resultados ARI para dados artificiais (Coleção Sintética 1 — Misturas de Gaussianas Multivariadas 2D e 10D): valores em verde (vermelho) indicam que a medida em questão obteve resultado melhor (pior) que a medida concorrente, enquanto valores em preto representam empate. Células destacadas em amarelo indicam o melhor de todos os resultados (dentre as diferentes medidas e valores de m_{pts}) para cada base. Os resultados foram subdivididos para os conjuntos de dados com 4 ou 10 *clusters* (Caso 1) e 20 ou 40 *clusters* (Caso 2).

	Medida	Conjunto de dados	m_{pts}									
			4	8	12	16	20	24	28	32	36	40
Caso 1	Stability	2d-4c	0.6853	0.9335	0.9316	0.9284	0.9175	0.905	0.8941	0.8892	0.8855	0.879
		2d-10c	0.6139	0.893	0.8894	0.8985	0.8943	0.8839	0.8758	0.8693	0.8669	0.8561
		10d-4c	0.5413	0.5483	0.4965	0.4605	0.4384	0.4172	0.3928	0.3817	0.3696	0.3745
		10d-10c	0.5439	0.4202	0.4044	0.385	0.3667	0.3053	0.3261	0.2641	0.2535	0.2966
	Mod. knn	2d-4c	0.9039	0.9468	0.9373	0.9284	0.9146	0.9051	0.9004	0.8959	0.8854	0.8803
		2d-10c	0.9399	0.9233	0.8992	0.8915	0.8714	0.8631	0.8466	0.8414	0.8301	0.8243
		10d-4c	0.7082	0.6147	0.5583	0.5157	0.4887	0.4679	0.454	0.4386	0.4268	0.4186
		10d-10c	0.6032	0.4978	0.4724	0.4446	0.4183	0.3939	0.3779	0.35	0.3389	0.3274
Caso 2	Stability	2d-20c	0.8602	0.9149	0.9053	0.8884	0.8743	0.8623	0.8484	0.8287	0.8007	0.7907
		2d-40c	0.7667	0.8394	0.8316	0.815	0.7839	0.7653	0.752	0.7339	0.715	0.6945
		10d-20c	0.9536	0.9290	0.8861	0.8768	0.8617	0.8497	0.8134	0.8041	0.7909	0.7284
		10d-40c	0.9374	0.9032	0.8789	0.8627	0.8163	0.792	0.7306	0.6926	0.6929	0.6788
	Mod. knn	2d-20c	0.8736	0.798	0.7703	0.7324	0.7225	0.6987	0.7045	0.6769	0.6644	0.6657
		2d-40c	0.7252	0.6428	0.5964	0.5727	0.5601	0.5265	0.5047	0.4877	0.4903	0.4519
		10d-20c	0.9235	0.8517	0.792	0.7751	0.7369	0.7301	0.7187	0.6773	0.6561	0.6762
		10d-40c	0.8419	0.7363	0.7159	0.659	0.5861	0.5816	0.5652	0.5598	0.5544	0.5708
Caso geral	Stability	Média	0.7378	0.7977	0.7779	0.7644	0.7441	0.7226	0.7041	0.6829	0.6719	0.6623
	Mod. knn	Média	0.8149	0.7514	0.7177	0.6899	0.6623	0.6459	0.634	0.616	0.6058	0.6019

Fonte: Dados da pesquisa.

Figura 8 – Resultados ARI para a coleção de dados sintéticos 1 (baixa dimensionalidade): sub-figuras à esquerda correspondem a bases com número de *clusters* menor (4 ou 10); subfiguras à direita correspondem a bases com número de *clusters* maior (20 ou 40).

Fonte: Dados da pesquisa.

A Tabela 6 apresenta os resultados do ARI em função de m_{pts} para as bases de dados da Coleção Sintética 2 (*Clusters* elipsoidais 50D e 100D). Assim como na Tabela 5, valores em verde ou vermelho indicam respectivamente que a medida correspondente à respectiva linha obteve resultado melhor ou pior que a medida concorrente, enquanto valores em preto indicam empate. Células destacas em amarelo indicam o melhor de todos os resultados (dentre as diferentes medidas e valores de m_{pts}) para cada base.

Observe na Tabela 6 que os resultados também foram subdivididos para os conjuntos de dados com 4 ou 10 *clusters* (Caso 1) e 20 ou 40 *clusters* (Caso 2), conforme

feito na análise da coleção sintética anterior. Porém, diferente do que foi observado na análise anterior, para esta coleção de dados, que envolve apenas conjuntos de dados de alta dimensionalidade, fica claro que a nova medida Mod. knn domina os resultados em ambos os casos, sendo que de forma mais proeminente para valores menores de m_{pts} (até 16). Note que, se apenas os melhores resultados obtidos para cada base de dados para os diferentes valores de m_{pts} forem considerados (células em amarelo), há um domínio da medida Mod. knn, que apresenta os melhores resultados em todas as bases.

Os resultados da Tabela 6 também são ilustrados na Figura 9, onde as bases do Caso 1 correspondem às subfiguras da esquerda, enquanto as bases do Caso 2 correspondem às subfiguras da direita. Nesta figura fica mais claro que, para esta coleção (Sintética 2), o desempenho relativo da medida Mod. knn com relação à Estabilidade é melhor para as bases com maiores quantidades de *clusters* (Caso 2), enquanto o inverso ocorre para as bases da Coleção Sintética 1, onde o desempenho Mod. knn é melhor em termos relativos para o Caso 1.

Esses experimentos mostram que o desempenho relativo das medidas não é consistente com relação ao número de *clusters*, sendo dependente do cenário. Mesmo que houvesse uma dependência clara com o número de *clusters*, esta não seria útil para guiar a escolha de uma medida em particular em uma dada aplicação real porque, na prática, o número de *clusters* nos dados é obviamente desconhecido.

O mesmo não é verdade com relação à dimensionalidade dos dados, que é conhecida *a priori*. Os experimentos descritos acima sugerem que a nova medida Mod. knn é mais robusta a espaços de dados de alta dimensionalidade, uma vez que o desempenho relativo desta medida quando comparado à Estabilidade melhora significativamente ao se comparar os resultados da Coleção Sintética 1 (baixa dimensionalidade — sumarizados na Figura 10) com os resultados da Coleção Sintética 2 (alta dimensionalidade — sumarizados na Figura 11).

Outra observação que pode ser útil para o analista em aplicações reais é que Mod. knn também apresenta resultados mais robustos que a Estabilidade para valores pequenos de m_{pts} , tais como $m_{pts} = 4$, que é uma escolha muito usual na prática (CAMPELLO *et al.*, 2015; GERTRUDES *et al.*, n.d.).

Tabela 6 – Resultados ARI para dados artificiais (Coleção Sintética 2 — *Clusters* Elipsoidais 50D e 100D): valores em verde (vermelho) indicam que a medida em questão obteve resultado melhor (pior) que a medida concorrente, enquanto valores em preto representam empate. Células destacadas em amarelo indicam o melhor de todos os resultados (dentre as diferentes medidas e valores de m_{pts}) para cada base. Os resultados foram subdivididos para os conjuntos de dados com 4 ou 10 *clusters* (Caso 1) e 20 ou 40 *clusters* (Caso 2).

			Medida	Conjunto de dados	m_{Pts}							
					4	8	12	16	20	24	28	32
Caso 1	Stability	50d-4c	0.8168	0.8011	0.7781	0.7371	0.7203	0.7042	0.686	0.669	0.6463	0.6302
		50d-10c	0.8496	0.8366	0.8221	0.809	0.7943	0.7757	0.7594	0.7404	0.6498	0.5697
		100d-4c	0.7758	0.7589	0.7369	0.72	0.6958	0.6714	0.6461	0.6154	0.5952	0.5618
		100d-10c	0.8488	0.847	0.8332	0.815	0.798	0.7786	0.7588	0.7378	0.72	0.7009
	Mod. knn	50d-4c	0.8208	0.8022	0.7792	0.7466	0.7226	0.7073	0.686	0.669	0.6513	0.6353
		50d-10c	0.8521	0.839	0.8239	0.8105	0.7886	0.7767	0.7562	0.7404	0.7159	0.6954
		100d-4c	0.7802	0.7623	0.739	0.7218	0.6971	0.6721	0.6485	0.622	0.5953	0.5716
		100d-10c	0.8624	0.8482	0.8333	0.8171	0.7985	0.7762	0.7579	0.7357	0.7178	0.7002
Caso 2	Stability	50d-20c	0.6706	0.704	0.5945	0.388	0.3849	0.349	0.206	0.1999	0.1314	0.083
		50d-40c	0.7079	0.7236	0.653	0.5726	0.3998	0.3639	0.1691	0.1324	0.0877	0.0561
		100d-20c	0.7248	0.7009	0.6948	0.6181	0.4617	0.43	0.3306	0.2029	0.1283	0.0943
		100d-40c	0.7706	0.7752	0.7242	0.6046	0.5798	0.3982	0.3099	0.0493	0.0642	0.0272
	Mod. knn	50d-20c	0.8185	0.7347	0.6595	0.5532	0.4891	0.4005	0.3382	0.2443	0.1666	0.127
		50d-40c	0.8353	0.7512	0.665	0.5622	0.4872	0.4074	0.3095	0.1693	0.0618	0.0634
		100d-20c	0.8625	0.7969	0.7224	0.6398	0.5703	0.4733	0.3819	0.2842	0.2526	0.2105
		100d-40c	0.8700	0.8194	0.7376	0.6617	0.5717	0.4791	0.3985	0.2912	0.1378	0.0718
Caso geral	Stability	Média	0.7706	0.7684	0.7296	0.658	0.6043	0.5589	0.4833	0.4184	0.3779	0.3404
	Mod. knn	Média	0.8377	0.7943	0.745	0.6891	0.6406	0.5866	0.5346	0.4695	0.4124	0.3844

Fonte: Dados da pesquisa.

Figura 9 – Resultados ARI para a coleção de dados sintéticos 2 (alta dimensionalidade): sub-figuras à esquerda correspondem a bases com número de *clusters* menor (4 ou 10); subfiguras à direita correspondem a bases com número de *clusters* maior (20 ou 40).

Fonte: Dados da pesquisa.

5.2.2 Resultados para Bases Reais

A Tabela 7 apresenta os resultados do ARI em função de m_{pts} para as bases de dados reais (sumarizadas na Tabela 4). Note que a medida proposta, Mod. knn, alcança os melhores resultados em 10 das 14 bases (2 delas em empate com a Estabilidade — células destacadas em azul). Essas bases incluem todas aquelas com alta dimensionalidade (> 50 dimensões), reforçando a hipótese de que Mod. knn é mais robusta que Estabilidade em espaços de alta dimensão. Os resultados são também visualizados na Figura 12. É inte-

Figura 10 – Resultados agregados da coleção de dados sintéticos 1. As áreas sombreadas representam os intervalos de confiança das médias (95%).

Fonte: Dados da pesquisa.

Figura 11 – Resultados agregados da coleção de dados sintéticos 2. As áreas sombreadas representam os intervalos de confiança das médias (95%).

Fonte: Dados da pesquisa.

ressante observar que, exceto por uma única base de dados (Wine, onde os resultados não são muito favoráveis em geral para ambas as medidas), Mod. knn supera a Estabilidade sistematicamente para m_{pts} no limite inferior da faixa ($m_{pts} = 4$), o que também está alinhado com as observações obtidas anteriormente a partir dos resultados em dados sintéticos.

Tabela 7 – Resultados ARI para dados reais: valores em verde (vermelho) indicam que a medida em questão obteve resultado melhor (pior) que a concorrente; valores em preto representam empate. Destaque em amarelo indica o melhor de todos os resultados (dentre as diferentes medidas e valores de m_{pts}) para cada base; destaque em azul indica que este melhor resultado foi alcançado por ambas as medidas.

Medida	Conjunto de dados	m_{pts}								
		4	6	8	10	12	14	16	18	20
Stability	CellCycle-237	0.2848	0.4749	0.4836	0.5727	0.5728	0.5273	0.5114	0.5023	0.5023
	CellCycle-384	0.3659	0.3794	0.3725	0.3724	0.3711	0.3579	0.3488	0.334	0.3447
	YeastGalactose	0.8715	0.8435	0.9349	0.9312	0.8636	0.8547	0.8288	0.8112	0.7945
	Wine	0.2867	0.2690	0.2765	0.2871	0.2686	0.2673	0.2220	0.2093	0.2353
	Glass	0.2790	0.2438	0.2517	0.2455	0.2498	0.2476	0.2516	0.2516	0.2599
	Iris	0.5681	0.5681	0.5681	0.5681	0.5681	0.5681	0.5657	0.5636	0.5615
	Ecoli	0.0384	0.0382	0.0381	0.4691	0.4638	0.4639	0.4626	0.4431	0.4542
	WDBC	0.0346	0.5520	0.5394	0.4588	0.4549	0.5445	0.4573	0.5373	0.4478
	Articles-1442-5	0.8999	0.8827	0.8664	0.8586	0.7197	0.7158	0.7084	0.7021	0.6954
	Articles-1442-80	0.9318	0.924	0.9218	0.9218	0.9143	0.8798	0.859	0.8365	0.8061
	Cbrilpirivson	0.1288	0.1378	0.1570	0.1243	0.0143	0.0181	0.0200	0.0226	0.0256
	Cbrilpirson	0.0495	0.0501	-0.0025	-0.0065	0.0155	0.0140	0.0111	0.0081	0.0073
	ALOI-TS88	0.5763	0.6472	0.6438	0.6190	0.5788	0.5314	0.4865	0.4372	0.3884
	ALOI-PCA	0.6751	0.7281	0.7199	0.6936	0.6648	0.6323	0.5981	0.5647	0.5178
Mod. knn	CellCycle-237	0.4831	0.4490	0.4630	0.5538	0.5663	0.5273	0.5114	0.5023	0.5023
	CellCycle-384	0.3744	0.3889	0.3776	0.3735	0.3711	0.3579	0.3488	0.3340	0.3447
	YeastGalactose	0.9528	0.9337	0.9349	0.9312	0.8636	0.8547	0.8288	0.8112	0.7945
	Wine	0.2650	0.2148	0.1489	0.1583	0.2686	0.2673	0.1345	0.1004	0.2353
	Glass	0.2505	0.2438	0.2419	0.2455	0.2498	0.2476	0.2516	0.2516	0.2599
	Iris	0.6177	0.5681	0.5681	0.5681	0.5681	0.5681	0.5657	0.5636	0.5615
	Ecoli	0.3895	0.4710	0.4720	0.4691	0.4638	0.4639	0.4626	0.4431	0.4542
	WDBC	0.1704	0.2778	0.4304	0.4192	0.4221	0.4129	0.4095	0.4086	0.4015
	Articles-1442-5	0.8999	0.8827	0.8664	0.8586	0.8270	0.8139	0.8101	0.8022	0.6954
	Articles-1442-80	0.9318	0.924	0.9218	0.9218	0.9143	0.8798	0.8590	0.8365	0.7030
	Cbrilpirivson	0.2879	0.2271	0.1881	0.1611	0.1530	0.1724	0.1667	0.1559	0.1562
	Cbrilpirson	0.1689	0.1413	0.1270	0.0938	0.0795	0.0738	0.064	0.0475	0.0615
	ALOI-TS88	0.7059	0.6926	0.6621	0.6283	0.5831	0.5356	0.4945	0.4464	0.4008
	ALOI-PCA	0.7741	0.7725	0.7482	0.7182	0.6870	0.6554	0.6241	0.5903	0.5471

Fonte: Dados da pesquisa.

5.2.3 Teste Estatístico

Para fins de análise estatística, o melhor valor de ARI obtido por cada medida em cada base de dados foi considerado. A Tabela 8 sumariza os resultados. Para as bases reais, há 8 vitórias para Mod. knn, 4 derrotas e 2 empates. O teste estatístico de Wilcoxon indica um *p-value* de 0.1169, o que representa uma probabilidade de aproximadamente 12% de que tal diferença ou maior seja observada entre os desempenhos das duas medidas sob a hipótese nula. Para as bases sintéticas o resultado é ainda mais claro, com uma probabilidade de apenas 2% associada à diferença entre os desempenhos na Coleção Sintética 1 (bases de baixa dimensionalidade, onde há 41 vitórias para a Estabilidade contra 32 vitórias para Mod. knn e 7 empates) e menos de 1% na Coleção Sintética 2 (bases de alta dimensionalidade, onde há 60 vitórias para a Mod. knn contra 1 vitória para Estabilidade e 19 empates).

Tabela 8 – Resultados gerais obtidos ao comparar as medidas Mod. knn e Stability. Para cada coleção de dados, foram realizadas a contagem de vitórias, empates e derrotas da Mod. knn em relação à Stability. Também foi realizado o teste estatístico de Wilcoxon com o fim de verificar diferença estatística entre os resultados obtidos pelas medidas.

Coleção de dados	Mod.knn × Stability			Teste de Wilcoxon: p-value
	Vitórias/	Empates/	Derrotas	
Dados Reais	8 /	2 /	4	0.1165
Coleção Sintética 1	32 /	7 /	41	0.0207
Coleção Sintética 2	60 /	19 /	1	< 0.01

Fonte: Dados da pesquisa.

5.3 Sumário

Neste capítulo foram discutidos os resultados experimentais do método proposto tanto em bases de dados sintéticas como em bases de dados reais. Foram utilizadas três coleções de bases de dados, duas coleções sintéticas contendo 80 conjuntos de dados cada e uma coleção contendo 14 conjuntos de dados reais. O algoritmo HDBSCAN* foi executado em todas as bases de dados para diferentes valores do parâmetro m_{pts} . FOSC foi então aplicado para extrair soluções planas de agrupamento a partir de cada uma das hierarquias geradas, utilizando tanto a medida original de Estabilidade como a nova medida proposta neste trabalho, Mod. knn. Os resultados foram comparados na forma de tabelas, figuras e teste estatístico. Os resultados sugerem fortemente que a medida proposta, Mod. knn, é mais robusta que a Estabilidade em bases de dados de alta dimensionalidade e para valores no extremo inferior do parâmetro m_{pts} , em particular $m_{pts} = 4$, que é uma escolha comum na literatura.

Figura 12 – Resultados ARI para a coleção de dados reais.

Fonte: Dados da pesquisa.

CONCLUSÕES

Análise de agrupamento (*cluster analysis*) é uma tarefa fundamental em mineração de dados onde se busca encontrar, de forma não supervisionada, um conjunto finito de categorias que evidencie as relações entre os objetos de um conjunto de dados de interesse. Algoritmos hierárquicos de agrupamento são particularmente úteis por conseguirem representar, inclusive visualmente, agrupamentos em diferentes níveis de granularidade. Mesmo quando se quer obter uma solução plana ou partição de agrupamento, existem vantagens em fazê-lo indiretamente a partir de uma hierarquia, ao invés de diretamente a partir dos dados. De fato, extrair *clusters* de uma hierarquia proporciona meios para a exploração de uma miríade de possíveis soluções candidatas a partir de uma visão global das estruturas de grupos presentes nos dados. Isso permite avaliar e quantificar diferentes aspectos de cada *cluster* candidato, não como uma fotografia estática daquele *cluster* em um nível de granularidade/especificidade/densidade particular, mas de acordo com o seu comportamento e relação a outros *clusters* dos dados ao longo de múltiplos níveis hierárquicos. No caso de algoritmos baseados em densidade hierárquicos, como HDBSCAN*, isso proporciona uma oportunidade única de detectar *clusters* a partir de diferentes níveis de densidade, o que não é possível com algoritmos baseados em densidade do tipo particional. Outra vantagem da extração de soluções planas a partir de hierarquias é que, ao contrário de muitos algoritmos particionais, não é preciso que o usuário defina *a priori* a quantidade de grupos.

Para extrair soluções de agrupamento planas a partir de hierarquias é necessário que seja realizado um corte sobre o dendrograma ou árvore hierárquica de grupos. A escolha de como realizar esse corte é um problema clássico que vem sendo investigado há décadas. Mais recentemente, este problema tem ganho especial importância no contexto de algoritmos hierárquicos baseados em densidade, pois somente estratégias mais sofisticadas de corte, em particular cortes não-horizontais denominados *cortes locais* conseguem selecionar grupos de densidades diferentes para compor a solução final. Recentemente, o

arcabouço FOSC foi introduzido na literatura para a realização de cortes locais em hierarquias de agrupamento como a solução de um problema de otimização. Dada uma árvore de *clusters* e uma função objetivo que agrega valores de qualidade individual de cada *cluster* candidato a ser inserido na solução, o método opera um algoritmo de programação dinâmica simples, efetivo e extremamente eficiente que percorre a árvore de baixo para cima comparando localmente a qualidade de cada nó (*cluster*) não folha da árvore com seus nós filhos (*sub-clusters*) e propagando as decisões acima na árvore para compor uma solução global e globalmente ótima em tempo linear.

A seleção dos *clusters* realizada por FOSC depende da função objetivo que quantifica a qualidade das soluções de agrupamento planas candidatas que se deseja maximizar. Tal função objetivo precisa ser tal que possa ser decomposta em valores numéricos individuais que quantificam a qualidade de cada *cluster* da solução. Em particular, esta função deve ser *aditiva*, o que significa que pode ser escrita como a soma de componentes individuais referentes a cada *cluster*, e *local*, o que significa que os valores de qualidade individuais possam ser calculados localmente apenas com informações referentes ao *cluster* em questão, independente de outros *clusters* existentes na solução.

Como o conceito de cortes locais e o método FOSC são muito recentes, até o momento são quase inexistentes medidas de qualidade que satisfazem as propriedades acima. Basicamente, a única medida já consolidada na literatura é a medida de Estabilidade proposta no próprio artigo original em que se introduziu FOSC, bem como a especialização desta medida para o caso de hierarquias baseadas em densidade que é utilizada por HDBSCAN*. O desenvolvimento de um novo critério de avaliação de grupos e agrupamentos, que satisfaz as propriedades de aditividade e localidade requeridas por FOSC, foi proposto como principal contribuição deste trabalho. Este critério é denominado Modularidade Q Baseada em Grafo de Similaridade KNN, ou simplesmente Modularidade KNN (Mod. knn). Esse critério se baseia na modelagem do problema de agrupamento e seleção de modelos utilizando conceitos de teoria dos grafos e detecção de comunidades em redes complexas.

Um amplo estudo experimental foi realizado envolvendo um total de 174 bases de dados tanto artificiais como reais com diferentes características (tamanho, dimensionalidade, número de categorias existentes) e de diferentes domínios de aplicação (texto, expressão gênica, imagens, e UCI). Especificamente, foram utilizadas três coleções de bases de dados, duas coleções sintéticas contendo 80 conjuntos de dados cada e uma coleção contendo 14 conjuntos de dados reais. O algoritmo HDBSCAN* foi executado em todas as bases de dados e FOSC foi então aplicado para extrair soluções planas de agrupamento a partir de cada uma das hierarquias geradas, utilizando tanto a medida original de Estabilidade como a nova medida proposta neste trabalho, Mod. knn. Os resultados foram comparados na forma de tabelas, figuras e teste estatístico. Os resultados sugerem for-

temente que a medida proposta, Mod. knn, é mais robusta que a Estabilidade em bases de dados de alta dimensionalidade e para valores no extremo inferior do parâmetro m_{pts} , em particular $m_{pts} = 4$, que é uma escolha comum na literatura. A medida também apresenta resultados competitivos de uma forma geral, se apresentando como uma medida complementar (ao invés de substituta) à medida de Estabilidade. Isso permite análise exploratória de dados com o uso de múltiplas medidas de avaliação e seleção de modelos, o que é recomendado especialmente em cenários de aprendizado não supervisionado.

Existem diferentes linhas de investigação para trabalhos futuros. Uma possibilidade seria investigar se é possível combinar as medidas de Estabilidade e Modularidade KNN em uma única medida mais robusta que se comporte de forma similar ou superior às duas medidas individuais na maioria dos cenários de aplicação.

Outra possível linha de pesquisa seria o uso dos conceitos de teoria de grafos e redes complexas utilizados neste trabalho não apenas para a seleção *a posteriori* de resultados de agrupamentos planos a partir da hierarquia baseada em densidade produzida por HDBSCAN*, mas para produzir tal hierarquia em problemas de detecção de comunidades em redes complexas. Em outras palavras, adaptar o algoritmo HDBSCAN*, atualmente apropriado para agrupamento de dados, também para detecção de comunidades em grafos.

REFERÊNCIAS

AGGARWAL, C. C.; REDDY, C. K. **Data Clustering: Algorithms and Applications**. 1st. ed. [S.l.]: Chapman & Hall/CRC, 2013. ISBN 1466558210, 9781466558212. Citado nas páginas 25 e 29.

ARRUDA, G. F. de; COSTA, L. da F.; RODRIGUES, F. A. A complex networks approach for data clustering. **Physica A: Statistical Mechanics and its Applications**, v. 391, n. 23, p. 6174–6183, 2012. Citado na página 64.

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. **Science**, v. 286, n. October, p. 11, 1999. Citado nas páginas 51 e 52.

BEZDEK, J. C.; PAL, N. R. Some new indexes of cluster validity. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, v. 28, n. 3, p. 301–315, 1998. Citado na página 63.

BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387310738. Citado nas páginas 25, 29 e 59.

BOUDAILLIER, E.; HÉBRAIL, G. Interactive interpretation of hierarchical clustering. In: **Principles of Data Min. and Knowl. Disc.** [S.l.]: Springer, 1997, (LNCS, v. 1263). p. 288–298. Citado na página 59.

_____. Interactive interpretation of hierarchical clustering. **Intell. Data Anal.**, v. 2, p. 229–244, 1998. Citado na página 59.

CALVO, B.; RODRIGO, G. S. scmamp: Statistical comparison of multiple algorithms in multiple problems. **The R Journal, Vol. 8/1, Aug. 2016**, The R Foundation, 2016. Citado na página 76.

CAMPELLO, R. J. G. B.; MOULAVI, D.; SANDER, J. Density-based clustering based on hierarchical density estimates. In: **Advances in Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2013. p. 160–172. Citado nas páginas 37, 39, 40, 59 e 74.

CAMPELLO, R. J. G. B.; MOULAVI, D.; ZIMEK, A.; SANDER, J. A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies. **Data Mining and Knowledge Discovery**, v. 27, n. 3, p. 344–371, 2013. Citado nas páginas 27, 35, 43, 45, 60, 61, 62 e 74.

_____. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. **ACM Transactions on Knowledge Discovery from Data**, v. 10, n. 1, p. 1–51, 2015. Citado nas páginas 38, 39, 40, 46, 59, 71, 74, 76 e 80.

CLAUSET, A.; SHALIZI, C. R.; NEWMAN, M. E. J. Power-law distributions in empirical data. **SIAM Rev.**, Society for Industrial and Applied Mathematics, v. 51, n. 4, p. 661–703, nov. 2009. Citado na página 52.

- DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. **Journal of Machine Learning Research**, v. 7, p. 1–30, 2006. Disponível em: <<http://www.jmlr.org/papers/v7/demsar06a.html>>. Citado na página 76.
- ERDÖS, P.; RÉNYI, A. On random graphs I. **Publ. Math. Debrecen**, v. 6, p. 290–297, 1959. Citado na página 51.
- ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. In: **KDD**. [S.l.: s.n.], 1996. v. 96, n. 34, p. 226–231. Citado nas páginas 31, 36 e 59.
- EVERITT, B. S.; LANDAU, S.; LEESE, M. **Cluster Analysis**. 4th. ed. [S.l.]: Oxford University Press, 2001. Citado nas páginas 30 e 58.
- FENG, Z.; XU, X.; YURUK, N.; SCHWEIGER, T. A. J. A Novel Similarity-Based Modularity Function for Graph Partitioning. In: **Data Warehousing and Knowledge Discovery**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 385–396. Citado nas páginas 65 e 66.
- FORTUNATO, S. Community detection in graphs. **Physics Reports**, Elsevier B.V., v. 486, n. 3-5, p. 75–174, feb 2010. Citado na página 53.
- GERTRUDES, J. C.; CAMPELLO, R. J. G. B.; SANDER, J.; ZIMEK, A. A unified framework of density-based clustering for semi-supervised classification. In: **submitted**. [S.l.: s.n.], n.d. Citado nas páginas 76 e 80.
- GEUSEBROEK, J.; BURGHOUTS, G. J.; SMEULDERS, A. W. M. The amsterdam library of object images. **International Journal of Computer Vision**, v. 61, n. 1, p. 103–112, 2005. Disponível em: <<https://doi.org/10.1023/B:VISI.0000042993.50813.60>>. Citado na página 74.
- GOWER, J. C.; ROSS, G. J. S. Minimum Spanning Trees and Single Linkage Cluster Analysis. **Journal of the Royal Statistical Society**, v. 18, n. 1, p. 54–64, 1969. Citado na página 41.
- HAN, J. **Data Mining: Concepts and Techniques**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005. ISBN 1558609016. Citado nas páginas 25 e 29.
- HANDL, J.; KNOWLES, J. An evolutionary approach to multiobjective clustering. **IEEE transactions on Evolutionary Computation**, v. 11, n. 1, p. 56–76, 2007. Citado na página 73.
- _____. **Cluster generators for large high-dimensional data sets with large numbers of clusters**. n.d. <<http://personalpages.manchester.ac.uk/staff/Julia.Handl/generators.pdf>>. Online; acessado 30 April 2018. Citado nas páginas 73 e 74.
- HARTIGAN, J. A. **Clustering algorithms**. [S.l.]: Wiley, 1975. Citado nas páginas 25, 26, 29, 30 e 38.
- HINNEBURG, A.; KEIM, D. A. *et al.* An efficient approach to clustering in large multimedia databases with noise. In: **KDD**. [S.l.: s.n.], 1998. v. 98, p. 58–65. Citado nas páginas 31 e 59.

- HORTA, D.; CAMPELLO, R. J. G. B. Automatic aspect discrimination in data clustering. **Pattern Recognition**, v. 45, n. 12, p. 4370–4388, 2012. Citado nas páginas 74 e 75.
- HUANG, J.; SUN, H.; SONG, Q.; DENG, H.; HAN, J. Revealing density-based clustering structure from the core-connected tree of a network. **IEEE Transactions on Knowledge and Data Engineering**, v. 25, n. 8, p. 1876–1889, 2013. Citado nas páginas 65 e 67.
- HUBERT, L.; ARABIE, P. Comparing partitions. **Journal of Classification**, v. 2, n. 1, p. 193–218, 1985. Citado na página 75.
- JAIN, A. K. Data clustering: 50 years beyond K-means. **Pattern Recognition Letters**, v. 31, n. 8, p. 651–666, 2010. Citado nas páginas 25 e 31.
- JAIN, A. K.; DUBES, R. C. **Algorithms for Clustering Data**. Englewood Cliffs, New Jersey 07632: Prentice-Hall, Inc., 1988. 304 p. ISBN 0-13-022278-X. Citado nas páginas 25, 26, 29, 30, 40, 41, 45, 58, 59, 62, 63 e 64.
- JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM Computing Surveys**, v. 31, n. 3, p. 264–323, 1999. Citado nas páginas 25 e 29.
- JASKOWIAK, P. A.; MOULAVI, D.; FURTADO, A. C.; CAMPELLO, R. J.; ZIMEK, A.; SANDER, J. On strategies for building effective ensembles of relative clustering validity criteria. **Knowledge and Information Systems**, v. 47, n. 2, p. 329–354, 2016. Citado na página 63.
- KAUFMAN, L.; ROUSSEEUW, P. J. **Finding groups in data: an introduction to cluster analysis**. [S.l.]: Wiley, 2005. Citado nas páginas 25, 29, 30 e 63.
- KRIEGEL, H.-P.; KRÖGER, P.; SANDER, J.; ZIMEK, A. Density-based clustering. **WIRES: Data Mining and Knowledge Discovery**, v. 1, n. 3, p. 231–240, 2011. Citado nas páginas 26 e 31.
- LESKOVEC, J.; KLEINBERG, J.; FALOUTSOS, C. Graphs over time: Densification laws, shrinking diameters and possible explanations. In: **Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining**. [S.l.]: ACM, 2005. (KDD '05), p. 177–187. ISBN 1-59593-135-X. Citado na página 52.
- LICHMAN, M. **UCI Machine Learning Repository**. 2013. Disponível em: <<http://archive.ics.uci.edu/ml>>. Citado nas páginas 74 e 75.
- MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. **Proceedings of the fifth Berkeley symposium on mathematical statistics and probability**. [S.l.], 1967. v. 1, n. 14, p. 281–297. Citado nas páginas 32 e 59.
- MEILĂ, M. Comparing clusterings—an information based distance. **Journal of Multivariate Analysis**, v. 98, n. 5, p. 873–895, 2007. Citado na página 27.
- MILLIGAN, G. W.; COOPER, M. C. An examination of procedures for determining the number of clusters in a data set. **Psychometrika**, v. 50, n. 2, p. 159–179, jun 1985. Citado nas páginas 26, 30 e 59.

- MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw Hill, 1997. Citado nas páginas 25 e 29.
- MOULAVI, D.; JASKOWIAK, P. A.; CAMPELLO, R. J. G. B.; ZIMEK, A.; SANDER, J. Density-Based Clustering Validation. **Proceedings of the 2014 SIAM International Conference on Data Mining**, n. i, p. 839–847, 2014. Citado nas páginas 63 e 64.
- NALDI, M. C.; CAMPELLO, R. J. G. B.; HRUSCHKA, E. R.; CARVALHO, A. C. P. L. F. Efficiency issues of evolutionary k-means. **Appl. Soft Comput.**, v. 11, n. 2, p. 1938–1952, 2011. Disponível em: <<https://doi.org/10.1016/j.asoc.2010.06.010>>. Citado na página 75.
- NEWMAN, M. E. The structure and function of complex networks. **SIAM review**, SIAM, v. 45, n. 2, p. 167–256, 2003. Citado na página 53.
- NEWMAN, M. E. J. Assortative Mixing in Networks. **Physical Review Letters**, v. 89, n. 20, p. 4, oct 2002. Citado na página 51.
- NEWMAN, M. E. J.; GIRVAN, M. Finding and evaluating community structure in networks. **Physical Review E**, v. 69, n. 2, p. 026113, feb 2004. Citado nas páginas 27 e 54.
- PAULOVICH, F. V.; NONATO, L. G.; MINGHIM, R.; LEVKOWITZ, H. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. **IEEE Trans. Vis. Comput. Graph.**, v. 14, n. 3, p. 564–575, 2008. Disponível em: <<https://doi.org/10.1109/TVCG.2007.70443>>. Citado na página 75.
- RABBANY, R.; TAKAFFOLI, M.; FAGNAN, J.; ZAIĀNE, O. R.; CAMPELLO, R. J. G. B. Communities validity: methodical evaluation of community mining algorithms. **Social Network Analysis and Mining**, v. 3, n. 4, p. 1039–1062, 2013. Citado na página 64.
- RAVASZ, E.; BARABÁSI, A.-L. Hierarchical organization in complex networks. **Physical Review E**, APS, v. 67, n. 2, p. 026112, 2003. Citado na página 53.
- SUN, H.; HUANG, J.; HAN, J.; DENG, H.; ZHAO, P.; FENG, B. gSkeletonClu: Density-Based Network Clustering via Structure-Connected Tree Division or Agglomeration. In: **2010 IEEE International Conference on Data Mining**. [S.l.]: IEEE, 2010. v. 1, p. 481–490. Citado nas páginas 65 e 67.
- TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining, (First Edition)**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321321367. Citado nas páginas 25, 29, 31 e 62.
- VENDRAMIN, L.; CAMPELLO, R. J. G. B.; HRUSCHKA, E. R. Relative clustering validity criteria: A comparative overview. **Statistical Analysis and Data Mining**, v. 3, n. 4, p. 209–235, jun 2010. Citado nas páginas 27 e 63.
- VINH, N. X.; EPPS, J.; BAILEY, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. **The Journal of Machine Learning Research**, v. 11, p. 2837–2854, 2010. Citado na página 27.

- WATTS, D. J.; STROGATZ, S. H. Collective dynamics of 'small-world' networks. **Nature**, v. 393, n. 6684, p. 440–2, 1998. Citado nas páginas [51](#) e [52](#).
- WILCOXON, F. Individual comparisons by ranking methods. **Biometrics**, JSTOR, v. 1, n. 6, p. 80–83, 1945. Citado na página [76](#).
- XU, R.; WUNSCH II, D. Survey of clustering algorithms. **IEEE Transactions on Neural Networks**, v. 16, n. 3, p. 645–678, 2005. Citado nas páginas [25](#), [29](#) e [30](#).
- XU, X.; YURUK, N.; FENG, Z.; SCHWEIGER, T. A. Scan: a structural clustering algorithm for networks. In: **Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.: s.n.], 2007. p. 824–833. Citado nas páginas [65](#) e [67](#).
- YEUNG, K. Y.; FRALEY, C.; MURUA, A.; RAFTERY, A. E.; RUZZO, W. L. Model-based clustering and data transformations for gene expression data. **Bioinformatics**, v. 17, n. 10, p. 977–987, 2001. Citado na página [75](#).
- YEUNG, K. Y.; MEDVEDOVIC, M.; BUMGARNER, R. E. Clustering gene-expression data with repeated measurements. **Genome Biol**, v. 4, n. 5, p. R34, 2003. Citado na página [75](#).