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

RAFAEL AUGUSTO FREITAS OLIVEIRA
Orientador: Jadson Castro Gertrudes

UM COMPARATIVO ENTRE ALGORITMOS DE AGRUPAMENTO SEMISSUPERVISIONADOS PARA PARTICIONAMENTO DE HIERARQUIAS DE GRUPOS

Ouro Preto, MG
2024

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

RAFAEL AUGUSTO FREITAS OLIVEIRA

UM COMPARATIVO ENTRE ALGORITMOS DE AGRUPAMENTO SEMISSUPERVISIONADOS PARA PARTICIONAMENTO DE HIERARQUIAS DE GRUPOS

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

Orientador: Jadson Castro Gertrudes

Ouro Preto, MG
2024

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

FOLHA DE APROVAÇÃO

Rafael Augusto Freitas Oliveira

Um comparativo entre algoritmos de agrupamento semissupervisionados para particionamento de hierarquias de grupos

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

Aprovada em 15 de Fevereiro de 2024.

Membros da banca:

Jadson Castro Gertrudes (Orientador) - Doutor - Universidade Federal de Ouro Preto
Mardochee Ogécime (Examinador) - Doutor - PPGCC UFOP
Josemar Coelho Felix (Examinador) - Mestre - PPGCCC UFOP

Jadson Castro Gertrudes, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 15/02/2024.

Documento assinado eletronicamente por **Jadson Castro Gertrudes**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 18/02/2024, às 18:05, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0667171** e o código CRC **9B1183B1**.

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

A Jessica, por ter aturado minha ausência por dias sem querer me matar.

Agradecimentos

A Deus, por ter permitido que eu tivesse saúde e determinação para não desanimar durante a realização deste trabalho. Aos meus pais e ao meu amor, que me incentivaram nos momentos difíceis e compreenderam a minha ausência enquanto eu me dedicava à realização deste trabalho. Ao professor Jadson, por ter sido meu orientador e ter desempenhado tal função com dedicação. E aos que me ajudaram direta ou indiretamente. A todos, um muito obrigado.

Resumo

Algoritmos de aprendizado semi supervisionado visam preencher lacunas existentes entre as abordagens de aprendizado supervisionada e não supervisionada. Esses algoritmos são projetados para lidar com situações em que há uma quantidade limitada de dados rotulados, mas uma vasta quantidade de dados não rotulados está disponível para treinamento. Essa abordagem híbrida busca aproveitar o melhor dos dois mundos, combinando a orientação do aprendizado supervisionado com a flexibilidade do aprendizado não supervisionado, resultando em modelos mais robustos. Este estudo tem como proposito avaliar a medida semi supervisionada baseada em rótulos, fundamentada em B^3 *Precision* e B^3 *Recall*, utilizada para extração ótima de grupos no *framework* FOSC. Observamos o desempenho e comportamento da medida de qualidade quando aplicada e diferentes tipos de árvores de hierarquia. Os testes foram aplicados tanto em dados artificiais quanto em dados reais, de diversas áreas e assuntos, contento diferentes aspectos e atributos. Os resultados foram comparados a um medida baseada em restrição. Foi possível observar que vários fatores influenciaram no desempenho dos algoritmos, e que no final, uma das métricas se provou melhor que a outra.

Palavras-chave: Aprendizado de máquina. Aprendizado semi supervisionado. Agrupamento de dados. Medidas de qualidade para extração de grupos. Medida de qualidade baseada em rótulos.

Abstract

Semi-supervised learning algorithms aim to fill gaps between supervised and unsupervised learning approaches. These algorithms are designed to deal with situations where there is a limited amount of labeled data, but a vast amount of unlabeled data is available for training. This hybrid approach seeks to take advantage of the best of both worlds, combining the guidance of supervised learning with the flexibility of unsupervised learning, resulting in more robust models. The purpose of this study is to evaluate the semi-supervised label-based measure, based on B^3 *Precision* and B^3 *Recall*, used for optimal group extraction in the *framework* FOSC. We observed the performance and behavior of the quality measure when applied to different types of hierarchy trees. The tests were applied to both artificial and real data, from different areas and subjects, containing different aspects and attributes. The results were compared to a constraint-based measure. It was possible to observe that several factors influenced the performance of the algorithms, and that in the end, one of the metrics proved to be better than the other.

Keywords: Machine learning. Semi-supervised learning. Data clustering. Quality measures for optimal cluster. Label-based Quality M.

Lista de Ilustrações

Lista de Tabelas

Lista de Abreviaturas e Siglas

ABNT	Associação Brasileira de Normas Técnicas
ALOI	*Amsterdam Library of Object Images*
AM	Aprendizado de Máquina
API	*Application Programming Interface*
ARI	*Adjusted Rand Index*
B^3	BCubed
DBSCAN	*Density-Based Spatial Clustering of Applications with Noise*
DECOM	Departamento de Computação
FOSC	*Framework for Optimal Selection of Clusters*
HDBSCAN*	*Hierarchical Density-Based Spatial Clustering of Applications with Noise*
PCA	*Principal Component Analysis*
SSDBSCAN	*Spatial Scan Density-Based Spatial Clustering of Applications with Noise*
UCI	*University of California, Irvine*
UFOP	Universidade Federal de Ouro Preto

Sumário

1 Introdução e Justificativa

Na literatura de aprendizado de máquina atual, os algoritmos mais prevalentes incluem tanto os métodos de aprendizado supervisionado quanto os não-supervisionados (Wu *et al.*, 2008). Embora essas abordagens tenham demonstrado eficácia em diversas tarefas, ambas enfrentam desafios relacionados aos dados. No aprendizado supervisionado, a escassez de dados rotulados pode resultar na construção de modelos de baixa qualidade. Por outro lado, no aprendizado não supervisionado, a ausência de orientação durante o processo de aprendizado pode levar a resultados equivocados, como dito em Alteryx (2024).

Em resposta a essas limitações, surgem os algoritmos de aprendizado semissupervisionado, que visam preencher a lacuna entre as duas abordagens. Esses algoritmos são projetados para lidar com situações em que há uma quantidade limitada de dados rotulados, mas uma vasta quantidade de dados não rotulados está disponível para treinamento (Chapelle; Schölkopf; Zien, 2006). Essa abordagem híbrida busca aproveitar o melhor dos dois mundos, combinando a orientação do aprendizado supervisionado com a flexibilidade do aprendizado não supervisionado, resultando em modelos mais robustos (Gertrudes *et al.*, 2019).

Encontramos exemplos de cenários de aprendizado semissupervisionado em vários campos, como filtragem de e-mail, reconhecimento de som/fala, classificação de texto/página da Web, entre outros (Chapelle; Schölkopf; Zien, 2006). Em áreas como biologia e medicina, podem ser necessários especialistas no domínio e análises laboratoriais para para rotular as observações e, geralmente, só é possível ter uma pequena coleção de dados rotulados, que pode não ser suficientemente representativa para que o aprendizado supervisionado seja aplicado (Batista; Campello; Sander, 2016).

Neste estudo, concentramos nossa atenção nos algoritmos de agrupamento semissupervisionado, uma abordagem que utiliza dos dados rotulados para orientar seu processo de aprendizado. Estes algoritmos fundamentam-se na estratégia de agrupamento de dados, criando grupos (*clusters*) nos quais são agrupados dados que compartilham semelhanças entre si, enquanto mantêm separados, em *clusters* distintos, aqueles que não exibem afinidades significativas (Tan; Steinbach; Kumar, 2005). Por meio dessa abordagem, os algoritmos de agrupamento semissupervisionado buscam maximizar a eficiência com a utilização dos dados rotulados, oferecendo, assim, uma alternativa para aprimorar a qualidade e a precisão dos resultados de agrupamento. Essa estratégia pode ser aplicada utilizando vários métodos de agrupamento, como o hierárquico, particional, baseado em grid e em densidade (Faceli *et al.*, 2021). No presente trabalho, tomamos por base os algoritmos de agrupamento hierárquico.

Os algoritmos de agrupamento hierárquico têm como objetivo agrupar os dados com base em medida de distância entre eles, representando os *clusters* em forma de um dendrograma. Nessa

estrutura, a solução de agrupamento ilustra os relacionamentos hierárquicos entre os grupos aninhados (Faceli *et al.*, 2021). Para que seja possível analisar grupos dentro deste dendrograma de forma não aninhada faz-se necessária a utilização de “cortes"no dendrograma, em geral, selecionando um determinado nível para isso, ou escolhendo o ponto de corte que apresente o número de grupos especificados pelo usuário (Campello *et al.*, 2013). Em alguns cenários, é necessária a realização de vários cortes locais no dendrograma com o fim de obter uma solução ótima para o agrupamento.

Para determinar os cortes locais dentro de uma hierarquia de grupos Campello *et al.* (2013) propuseram um algoritmo para extração ótima de grupos, denominado FOSC (*Framework for Optimal Selection of Clusters*). Este modelo considera como entrada uma árvore de grupos e uma medida de qualidade. Assim, o processo de extração considera os grupos que maximizam a medida de qualidade. Vale destacar que neste trabalho foram apresentadas medidas de qualidade tanto para o cenário semissupervisionado quanto não-supervisionado.

Uma nova abordagem de avaliação de qualidade em agrupamento semissupervisionado foi apresentada por Gertrudes *et al.* (2019), onde os autores exploram a aplicação experimental dessa abordagem no contexto do modelo de agrupamento hierárquico baseado em densidade HDBSCAN* (Campello *et al.*, 2015), que emprega o método FOSC para a extração otimizada de grupos. Contudo, essa métrica não foi extensivamente examinada na extração de grupos em hierarquias geradas por algoritmos hierárquicos tradicionais, como *Single linkage* e *average linkage*, entre outros.

Portanto, surge a relevante oportunidade de avaliar a eficácia dessa medida em diferentes tipos de hierarquias de agrupamento. Sendo assim, dado uma base de dados real, qual seria o método e/ou algoritmo capaz de agrupar os dados dessa base de maneira eficaz e precisa? Qual característica uma base de dados precisa ter para um resultado relevante?

1.1 Objetivos

Este estudo tem como objetivo realizar uma análise do algoritmo FOSC, comparando a métrica semissupervisionada baseada em rotulo proposta por Gertrudes *et al.* (2019), com a métrica baseada em restrição apresentada em (Campello *et al.*, 2013), através das arvore geradas com os algoritmos de agrupamento hierárquico *Single*, *Complete* e *Average linkage*, não empregados no trabalho de Gertrudes *et al.* (2019). Para que o objetivo geral seja alcançado, delimitamos os seguintes objetivos específicos:

- Revisar trabalhos relacionados na literatura;
- Aplicar a métrica de qualidade em métodos tradicionais de agrupamento, como os de ligação simples (*single linkage*), média (*average linkage*) e completa (*complete linkage*);

- Avaliar os resultados por meio de índices de validação de agrupamento de dados;
- Comparar os resultados da métrica baseada em rotulo com os resultados da métrica baseada em restrição.

1.2 Organização do Trabalho

O restante do trabalho está organizado na seguinte forma: O Capítulo 2 é feita uma apresentação aprofundada sobre todo o processo e histórico do agrupamento semisupervisionado de dados, além de apresentar várias estratégias e algoritmos que serão utilizados no decorrer do trabalho. No Capítulo 3 é apresentada a metodologia e a base de dados utilizada durante as experimentações. O Capítulo 4 apresenta os resultados obtidos a partir das configurações dos experimentos. Por fim, no Capítulo 5 são apresentadas as conclusões do trabalho além de perspectiva de trabalhos futuros.

2 Revisão Bibliográfica

2.1 Fundamentação Teórica

No campo do Aprendizado de Máquina, os algoritmos mais predominantes são os de aprendizado supervisionado e não-supervisionado. No primeiro caso, o termo "supervisionado" denota a presença de um orientador externo que possui conhecimento sobre os rótulos verdadeiros dos dados, utilizados para guiar o processo de aprendizado na criação de um modelo com sólida capacidade preditiva. Em contraste, nos algoritmos não-supervisionados, a entrada consiste em um conjunto de dados sem rótulos, e o objetivo do algoritmo é identificar e extrair padrões intrínsecos presentes nos dados fornecidos.

Neste estudo, exploramos algoritmos que permeiam as duas abordagens descritas anteriormente, conhecidos como algoritmos de aprendizado semissupervisionado. Em particular, nos concentramos na análise dos algoritmos de agrupamento semissupervisionado. Com a intenção de facilitar a compreensão das técnicas de agrupamento semissupervisionado discutidas neste trabalho, forneceremos uma definição abrangente do conceito de agrupamento de dados, além de elucidar métodos que englobam a construção hierárquica de grupos e a subsequente extração desses grupos.

2.1.1 Agrupamento de dados

O agrupamento de dados, também conhecido como *clustering*, é uma subcampo do aprendizado de máquina que envolve a organização de um conjunto de objetos (ou dados) em grupos, denominados *clusters*, com base em alguma medida de similaridade entre eles (Faceli *et al.*, 2021). O objetivo principal é que objetos no mesmo grupo possuam maior similaridade entre eles do que em relação a objetos em diferentes grupos. Os padrões obtidos por meio desses algoritmos permitem a descoberta de informações e *insights* que estejam ocultos em um conjunto de dados.

De maneira geral, os algoritmos empregados no agrupamento de dados podem ser categorizados em três grupos: particionais, baseados em densidade e hierárquicos. No primeiro, o algoritmo trabalha com a definição prévia do número de grupos, representado por k, e se esforça para distribuir de maneira otimizada cada um dos objetos dentre esses k grupos. O clássico algoritmo particional k-médias exemplifica essa abordagem (MacQueen, 1967). Por sua vez, os algoritmos baseados em densidade fundamentam-se na premissa de que os *clusters* correspondem a regiões de alta densidade, separadas por áreas de baixa densidade. Quando um *cluster* é considerado um conjunto denso, ele pode expandir em qualquer direção, resultando em uma formação de *clusters* que tende a ser completamente arbitrária (Faceli *et al.*, 2021). O

destaque nesta categoria é o algoritmo DBSCAN (Ester *et al.*, 1996).

Os algoritmos hierárquicos precisam utilizar alguma métrica para integração entre os dados. As métricas servem para dividir ou aglomerar os *clusters* já existentes. Os dados podem ser fornecidos ao algoritmo em forma de uma matriz de proximidade, e o resultado do algoritmo é uma sequência de partições aninhadas, que podem ser visualizadas por meio de um dendrograma. Os dados são manipulados de uma maneira que, se em algum momento dois dados são colocados em um mesmo *cluster*, nos níveis acima esses dados permanecerão juntos, formando uma hierarquia de *cluster* (Faceli *et al.*, 2021).

2.1.2 Agrupamento semissupervisionado

O agrupamento semissupervisionado é um método que particiona dados não rotulados ao fazer uso do conhecimento do domínio. Geralmente, é expresso por meio de restrições em pares entre instâncias, denominadas *must-link* e *cannot-link*, ou apenas como um conjunto adicional de instâncias rotuladas. Essas restrições normalmente são fornecidas por especialistas no domínio.

2.1.3 Algoritmos de agrupamento hierárquico

Os algoritmos de agrupamento hierárquico aglomerativos utilizam de medidas de ligação (*linkage metrics*) para a geração da hierarquia de grupos. Elas representam medidas de distância entre *clusters*, capazes de gerar *clusters* de formas convexas próprias. Existem 3 algoritmos clássicos para o cálculo dessa distância como o de ligação simples *Single linkage*, ligação completa *Complete linkage* e ligação média *Average Linkage* (Faceli *et al.*, 2021). Esses algoritmos têm como entrada uma matriz de proximidade e produzem uma sequência de partições aninhadas, as hierarquias.

Dados dois *clusters* $\mathbf{C}_1 = \{x_1, x_2, \ldots, x_n\}$ e $\mathbf{C}_2 = \{x_1, x_2, \ldots, x_m\}$, onde x_i representa um objeto qualquer aninhado em um *cluster* $\mathbf{C}_j$ qualquer, os algoritmos hierárquicos aglomerativos calculam uma medida de (dis)similaridade entre todos os pares de objetos dos *clusters* distintos e definem uma estratégia de ligação entre eles. Por exemplo, *single linkage* visa agrupar os *clusters* que tenham a menor distância entre os pares de objetos de cada *cluster*. A fórmula de ligação do *Single Linkage* é dada pela Equação 2.1 e um exemplo visual é apresentado na Figura 2.1:

$$d_{single}(\mathbf{C}_1, \mathbf{C}_2) \; = \; min\{d(x_i, x_j)\}, \; com \; x_i \; \in \; \mathbf{C}_1 \; e \; x_j \; \in \; \mathbf{C}_2. \tag{2.1}$$

Um dos principais problemas do *Single linkage* é que alguns *clusters* podem se fundir por que apenas um único objeto de um cluster. $\mathbf{A}$ se encontra próximo a algum outro objeto de um *cluster* $\mathbf{B}$, porém todos os outros objetos do *cluster* $\mathbf{A}$ se encontram distantes dos objetos de $\mathbf{B}$. Isso é chamado de efeito de encadeamento e ocorre tipicamente quando a base de dados contém ruídos (Jarman, 2020).

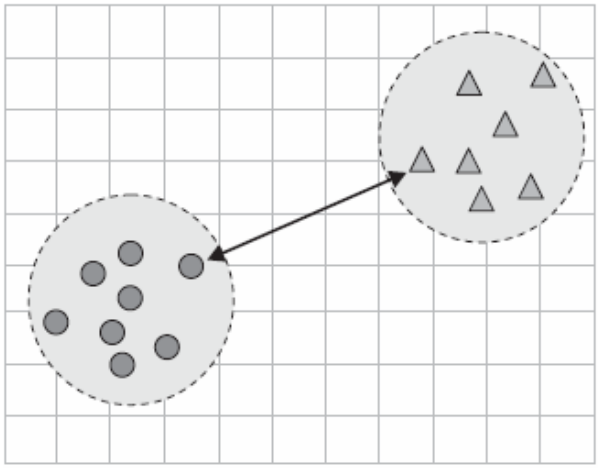

Figura 2.1 – Single Linkage
Fonte: Faceli *et al.* (2021), página 194

O *complete linkage* busca entre os pares de *clusters* a maior distância entre objetos de diferentes *clusters*. Assim como o *single linkage*, o *complete linkage* também tem vulnerabilidade a dados ruidosos Jarman (2020). A Equação 2.2 e a Figura 2.2 apresentam a formulação de ligação e um exemplo de ligação, respectivamente.

$$d_{complete}(\mathbf{C}_1, \mathbf{C}_2) \ = \ max\{d(x_i, x_j)\} \ com \ x_i \ \in \ \mathbf{C}_1 \ e \ x_j \ \in \ \mathbf{C}_2 \tag{2.2}$$

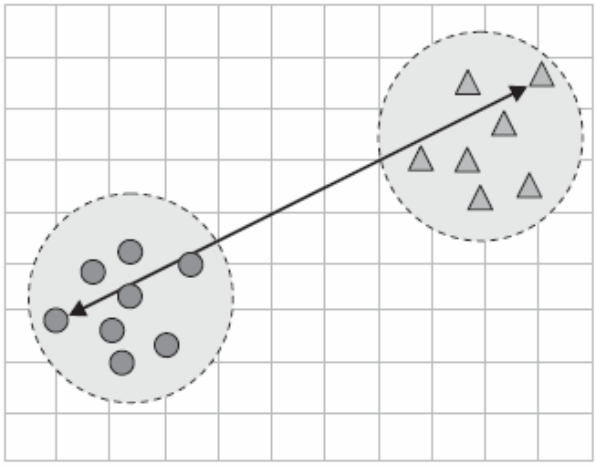

Figura 2.2 – Complete Linkage
Fonte: Faceli *et al.* (2021), página 194

Por fim, o método *Average linkage* calcula a média de todas as distâncias entre os pares de objetos de cada *cluster*. A Equação 2.3 e a Figura 2.3 apresentam o cálculo da ligação e um exemplo visual desta ligação, respectivamente.

$$d(\mathbf{C}_1, \mathbf{C}_2) \ = \ \frac{1}{n_1 n_2} \sum_{\substack{x_i \in \mathbf{C}_1 \\ x_j \in \mathbf{C}_2}} d(x_i, x_j) \tag{2.3}$$

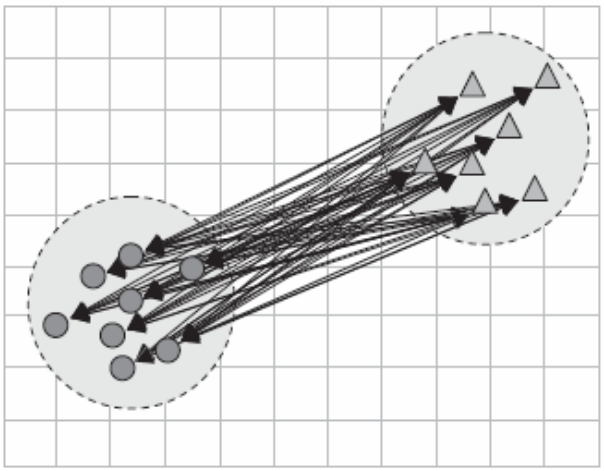

Figura 2.3 – Average Linkage.
Fonte: Faceli *et al.* (2021), página 194

2.1.4 FOSC

Uma das bases principais deste trabalho é o FOSC (*Framework for Optimal Selection of Clusters*), desenvolvido por Campello *et al.* (2013). Ele recebe como entrada uma árvore de grupos, e tem como objetivo extrair uma solução ótima local a partir dela, transformando o problema de corte global de um dendrograma em um problema de otimização, e retorna um conjunto de *clusters* que melhor representa o agrupamento dos dados de entrada.

Em geral, a árvore de *clusters* é obtida por meio da simplificação de um dendrograma, gerados via algoritmos de agrupamento hierárquico, como os algoritmos descritos na subseção 2.1.3. Para isso, faz-se o uso do parâmetro que indica o número mínimo de objetos que um *cluster* deve possuir, denominado m_{ClSize}. Tomemos como exemplo uma base de dados, a Figura 2.4, e a Tabela 2.1 a sua hierarquia de agrupamento com m_{ClSize} = 3, onde as linhas correspondem aos níveis hierárquicos e as colunas correspondem aos objetos (os valores 0 representa que aquele objeto se tornou um ruído).

	x1	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**x10**
1.44	1	1	1	1	1	1	1	1	1	1
1.37	2	2	2	3	3	3	3	3	3	3
1.18	2	2	2	4	4	4	5	5	5	5
0.72	2	2	2	4	4	4	5	5	5	0
0.68	2	2	2	0	0	0	5	5	5	0
0.64	2	2	2	0	0	0	0	0	0	0
0.55	0	0	0	0	0	0	0	0	0	0

Tabela 2.1 – Hierarquia de cluster com m_{ClSize} = 3.

Fonte: Elaborado pelo autor.

Formalmente, seja $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_k\}$ o conjunto dos *clusters* candidatos pertencentes a uma árvore de *clusters* da qual queremos extrair uma solução $\mathbf{P}$. Para isso, é assumido exista

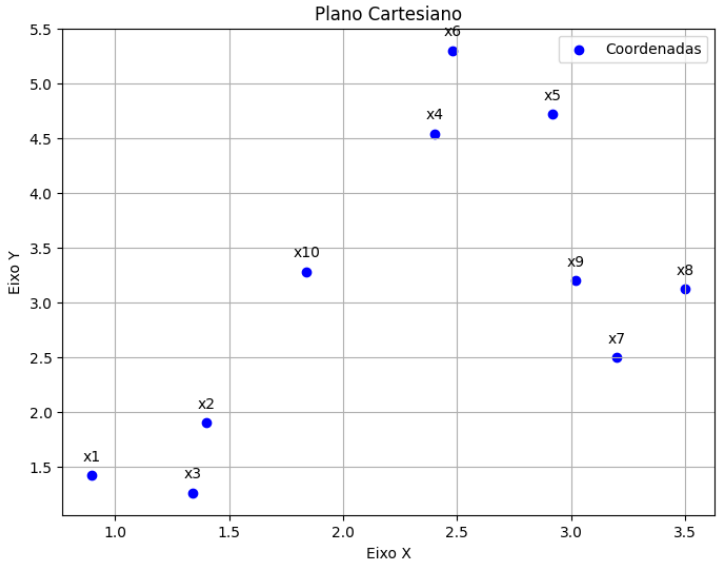

Figura 2.4 – Conjunto de Dados.

Fonte: Elaborado pelo autor

uma função objetivo $J(\mathbf{P})$ a ser maximizada, onde J possa avaliar a qualidade de cada solução candidata. Funcionalmente, $J(\mathbf{P})$ tem que satisfazer duas propriedades: aditividade, onde $J(\mathbf{P})$ pode ser reescrito como a soma dos componentes individuais $J(\mathbf{C}_i)$, cada um associado a um único *cluster* i; e localidade, onde cada componente $J(\mathbf{C}_i)$, possa ser computado localmente independente dos outros *clusters* candidatos. O valor $J(\mathbf{C}_i)$ pode ser calculado antecipadamente, devido às propriedades da aditividade. Devido a propriedade de aditividade, a função objetivo é definida por Campello *et al.* (2013):

$$J(\mathbf{P}) = \sum_{\mathbf{C}_i \in \mathbf{P}} \delta_i J(\mathbf{C}_i) \tag{2.4}$$

Matematicamente este problema pode ser decomposto como um problema de otimização formulado como (Campello *et al.*, 2013):

$$\max_{\delta_1,\ldots,\delta_k} \quad \sum_{i=1}^{k} \delta_i J(\mathbf{C}_i) \tag{2.5}$$

$$s.t \quad \delta_i \in \{0,1\}, i = 1, ..., k \tag{2.6}$$

$$\sum_{j \in I_h} \delta_j = 1, \forall h \text{ tal que } C_h \text{ é um nó/cluster folha} \tag{2.7}$$

Onde δi é uma valor booleano que demonstra se o cluster $\mathbf{C}_i$ faz parte (δi = 1) ou não (δi = 0) da solução final e Ih representa o conjunto de *cluster* de um caminho entre um nó externo $\mathbf{C}_h$, até a raiz $\mathbf{C}_1$.

Na Figura 2.5, os valores abaixo de cada nó representam o valor $J(\mathbf{C}_i)$ calculado antecipadamente, a forma de calcular esse valor, é baseada no tempo de vida do *cluster* que é basicamente o comprimento da escala do dendrograma ao longo do qual o *cluster* existe (Campello *et al.*,

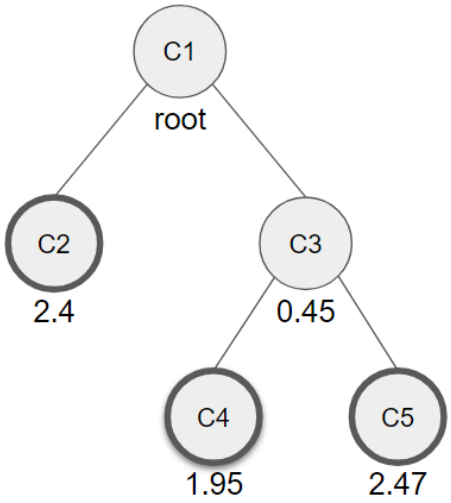

Figura 2.5 – Árvore de cluster representando a exceção do FOSC nos dados da tabela 2.1.

Fonte: Elaborado pelo autor.

2013). Dessa maneira, *clusters* mais proeminentes persistem em vários níveis hierárquicos, logo, tem uma vida útil mais longa. Campello *et al.* (2013) explicam que a medida foi adotada, pois em certas hierarquias, incluindo a baseada em densidade, nem todos os objetos permanecem no *cluster* durante toda a sua existência, pois alguns deles se tornam ruídos. Esta medida de estabilidade (*stability*) proposta por Campello *et al.* (2013) é a soma dos tempos de vida de cada objeto naquele *cluster*, e pode ser descrita como:

$$J(\mathbf{C}_i) = \sum_{x_j \in \mathbf{C}_i} lifetime(x_j), \tag{2.8}$$

onde $\mathbf{C}_i$ é um *cluster* arbitrário e x_j é um objeto que pertence ao *cluster*.

Por exemplo, se olharmos na Tabela 2.1, o *cluster* $\mathbf{C}_5$ começa com os objetos $\mathbf{x}_7$, $\mathbf{x}_8$ e $\mathbf{x}_9$ no nível $0,68$ e termina no nível $1,37$, onde se funde com $\mathbf{C}_4$ dando origem ao $\mathbf{C}_3$. Porém, ao longo da vida de $\mathbf{C}_5$ o objeto $\mathbf{x}_{10}$ se une a ele no nível $1,18$, ou seja a estabilidade desse *cluster* pode ser dada por $J(\mathbf{C}_5) = 3 * (1,37 - 0,61) + (1,37 - 1,18) = 2,47$.

Observe que na Figura 2.5 o valor de $J(\mathbf{C}_4) = 1,95$ e o de $J(\mathbf{C}_5) = 2,47$, e a soma desses dois valores é igual a $4,42$, que é maior do que o valor de $J(\mathbf{C}_3) = 0,45$, portanto $\mathbf{C}_4$ e $\mathbf{C}_5$ são selecionados como clusters candidatos. Como $\mathbf{C}_2$ não tem filhos, ele é selecionado como cluster candidato, logo, a solução final é dada por $\mathbf{P} = \{\mathbf{C}_2, \mathbf{C}_4, \mathbf{C}_5\}$ com $J_T(\mathbf{P}) = 6,82$.

2.1.5 Extração de grupos baseado em rótulos

Gertrudes *et al.* (2019) apresentam uma nova medida de qualidade semissupervisionada para o procedimento de extração ótima de grupos usado pelo método HDBSCAN* (Campello *et al.*, 2015). O artigo também descreve como a medida não supervisionada Stability, descrita na

subseção 2.1.4, pode ser combinada com a sua medida baseada em rótulo , tornando o resultado eficaz, independente de apenas parte, todos ou nenhum dos *clusters* serem representados por observações rotuladas.

A medida utilizada por (Gertrudes *et al.*, 2019) é baseada nos critérios B^3 *Precision* e B^3 *Recall*, que foram propostos por Bagga e Baldwin (1998). Esses critérios levam em consideração pares de objetos na comparação, porém são computados individualmente para cada objeto pré-rotulado presente na base, denotado por $x \in \mathbf{X}_L$. O B^3 *Precision* de um objeto x mede a proporção de objetos pré-rotulados, incluindo o próprio x, que compartilham o mesmo *cluster* que x. Já o B^3 *Recall* determina a quantidade de objetos que tenham o mesmo rótulo e compartilhem o mesmo *cluster* que x, sobre o total de objetos com o mesmo rótulo de x. Ambos os casos deixam de fora dos cálculos os objetos que ainda não foram rotulados.

Dado um objeto $x \in \mathbf{X}_L$, um *cluster* $\mathbf{C}_i$ e $class(x)$ a função que retorna a classe de um objeto, o *Precision* B^3 e o *Recall* B^3 de um objeto x pode ser definido como

$$P_{B^3}(x, \mathbf{C}_i) = \frac{|\{x' \mid x' \in \{\mathbf{C}_i \cap \mathbf{X}_L\} \wedge class(x) = class(x')\}|}{|\{x' \mid x' \in \{\mathbf{C}_i \cap \mathbf{X}_L\}|} \tag{2.9}$$

$$R_{B^3}(x, \mathbf{C}_i) = \frac{|\{x' \mid x' \in \{\mathbf{C}_i \cap \mathbf{X}_L\} \wedge class(x) = class(x')\}|}{|\{x' \mid x' \in \mathbf{X}_L \wedge class(x) = class(x')\}|} \tag{2.10}$$

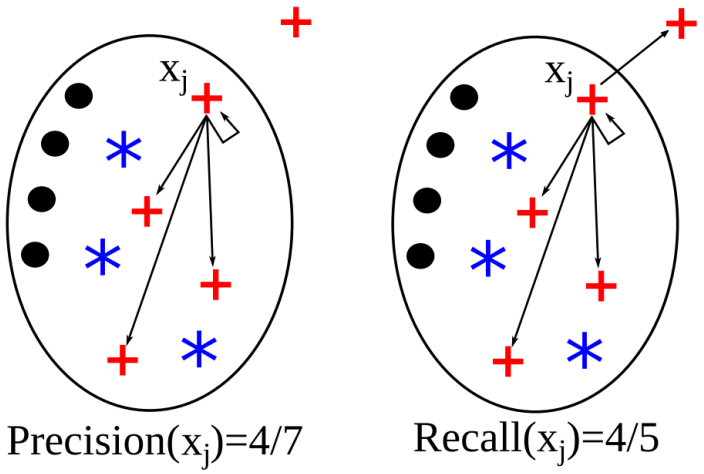

Figura 2.6 – Representação visual do B^3 *Precision* e B^3 *Recall*
Fonte: Gertrudes *et al.* (2019), página 28.

Uma das falhas dessas duas métricas é que elas capturam dois aspectos diferentes em cada um de seus resultados, onde um vai negligenciar o outro que foi escolhido. Porém esse dois critérios podem ser fundidos tomando sua média harmônica, vamos chamá-lo de B^3 F-Measure, definido como:

$$F_{B^3}(x, \mathbf{C}_i) = \frac{2P_{B^3}(x, \mathbf{C}_i) \times R_{B^3}(x, \mathbf{C}_i)}{P_{B^3}(x, \mathbf{C}_i) + R_{B^3}(x, \mathbf{C}_i)} \tag{2.11}$$

A equação B^3 F-Measure pode ser usada como critério de otimização para extrair *clusters* de árvores de grupos usando o FOSC. No trabalho de Gertrudes *et al.* (2019) a árvore hierárquica

foi gerada usando o HDBSCAN*. Porém, no presente trabalho ele será adaptado para ser utilizado com as hierarquias obtidas a partir dos dendrogramas gerados pelos métodos de ligação. A formulação geral para o F-Measure B^3 é dada por

$$Overall_{F_{B^3}} = \frac{1}{|\mathbf{X}_L|} \sum_{i=1}^{k} \left(\sum_{x \in \{\mathbf{C}_i \cap \mathbf{X}_L\}} \delta_i * F_{B^3}(x, \mathbf{C}_i) \right), \tag{2.12}$$

onde δ_i simboliza se $\mathbf{C}_i$ vai ($\delta_i = 1$) ou não ($\delta_i = 0$) participar da partição ótima. Esta equação foi decomposta em Gertrudes *et al.* (2019) como:

$$Overall_{F_{B^3}} = \sum_{i=1}^{k} \delta_i * \omega(\mathbf{C}_i), \tag{2.13}$$

onde

$$\omega(\mathbf{C}_i) = \frac{1}{|\mathbf{X}_L|} \left(\sum_{x \in \{\mathbf{C}_i \cap \mathbf{X}_L\}} \delta_i * F_{B^3}(x, \mathbf{C}_i) \right), \tag{2.14}$$

ou seja, *Overall* B^3 F-Measure é uma soma de componentes individuais ($\mathbf{C}_i$) que podem ser calculados independentemente dos outros *clusters* candidatos.

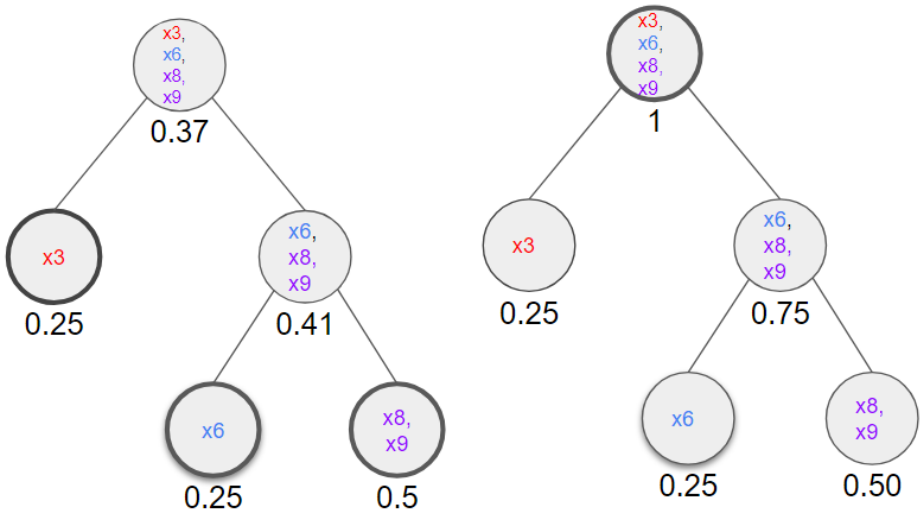

Figura 2.7 – Representação visual da execução do *Overall* B^3 *Precision* e *Overall* B^3 *Recall* respectivamente. Fonte: Elaborado pelo Autor

A figura 2.7 demonstram a árvore de agrupamento para a hierarquia da tabela 2.1 gerada pelo *Overall* B^3 *Precision* e *Overall* B^3 *Recall*, respectivamente, com as coleções de objetos pré rotulados XL = {x3, x6, x8, x9} com 3 rótulos, x3 vermelho, x6 azul e x8 e x9 roxos (Cores diponiveis somente online). Podemos ver que o B^3 *Precision* levou o FOSC a escolher os nós folhas pois eles tendem a ser mais puros nos rótulos de classe, em contraste o B^3 *Recall* orientou o FOSC a extrair a raiz.

Na figura 2.8 podemos ver a árvore de hierarquia da tabela 2.1 gerada pelo *Overall* B^3 F-Measure, o resultado do acabou sendo o mesmo que a execução usando *Overall* B^3 *Preci-*

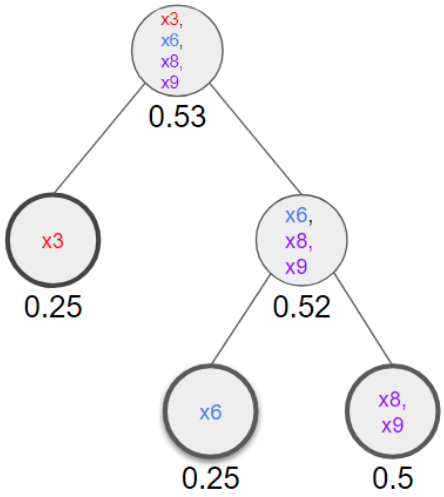

Figura 2.8 – Representação visual da execução do *Overall* B^3 F-Measure. Fonte: Elaborado pelo Autor

sion, porem a solução extraída não precisa coincidir com o *Precision* ou o *Recall*. Os objetos considerados ruídos foram omitidos e definidos como zero para não afetarem os cálculos.

2.1.6 Extração de grupos baseado em restrição

O artigo Campello *et al.* (2013) também apresenta um modelo de extração ótima de grupos baseado em restrições. Temos como entrada um conjunto de dados **X** e um conjunto de restrições do tipo *should-link* (deve conectar) e *should-not-link* (não deve conectar), que representam o conhecimento prévio dos dados, e que serão violados ou satisfeitos. Como apresentado na Equação 2.8, o FOSC precisa de uma função objetivo para calcular a qualidade de cada cluster, nesse caso, essa função pode ser apresentada como:

$$J = \frac{1}{2n_c}\sum_{j=1}^{n}\gamma(\mathbf{x}_j), \tag{2.15}$$

onde n_c é o número de restrições existentes e $\gamma(x_j)$ é o número de restrições relacionadas ao objeto $\mathbf{x}_j$ que foram satisfeitas. a função é multiplicada por 1/2 pois uma única restrição é considerada por um par de objetos, exemplo, *should-link***(a, b)** e *should-link***(b, a)** são duas restrições distintas que valem para os mesmos objetos, onde uma conta para o objeto **a** e a outra para o objeto **b**. O temo n_c no denominador acaba por normalizar J, sendo assim, a Equação 2.15 representa a fração de restrições que são satisfeitas, logo, maximizar J é minimizar o número de restrições violadas.

Supondo que C seja o conjunto de clusters existentes, iremos retirar C_1 (a raiz) dos clusters candidatos, ou seja, teremos E = $\{C_2, ..., C_k\}$ como o conjunto de clusters candidatos para a solução final, e F $\subseteq$ E qualquer solução candidata de clusters obtido pela Equação 2.5. Seja também $X_L \subseteq X$ o subconjunto de objetos que possuem um rotulo não nulo em tal solução candidata, logo, $X_L = \{ x_j \mid \exists C_i \in F: x_j \in C_i \}$. Além disso, tomemos $\overline{X}_L$ o subconjunto de objetos com rótulo nulo (ruídos), ou seja, $\overline{X}_L = X - X_L$. Por fim, podemos reescrever a Equação 2.15

como:

$$J = \frac{1}{2n_c} \sum_{\mathbf{x}_j \in \mathbf{X}_L} \gamma(\mathbf{x}_j) + \frac{1}{2n_c} \sum_{\overline{\mathbf{X}}_j \in \mathbf{X}_L} \gamma(\mathbf{x}_j) \tag{2.16}$$

$$= \frac{1}{2n_c} \sum_{i=2}^{k} \delta_i \Gamma(\mathbf{C}_i) + \frac{1}{2n_c} \sum_{\overline{\mathbf{X}}_j \in \mathbf{X}_L} \gamma(\mathbf{x}_j), \tag{2.17}$$

onde:

$$\Gamma(\mathbf{C}_i) = \frac{1}{2n_c} \sum_{\mathbf{x}_j \in \mathbf{C}_i} \gamma(\mathbf{x}_j), \tag{2.18}$$

representa a fração das restrições satisfeitas envolvendo os objetos do cluster $\mathbf{C}_i$.

Considerando a arvore da Figura 2.8, vamos representar os dois nós folhas mais abaixo da arvore como C_4 e C_5, da esquerda para a direita, e o seu cluster pai como C_3. Tomemos também uma restrição de *should-link*, (x8, x9), e uma restrição de *should-not-link*, (x6, x8). Podemos calcular o $\Gamma(C_3) = \frac{1}{2*2}(\gamma(x_6) + \gamma(x_8) + \gamma(x_9)) = \frac{1}{4}(0 + 1 + 1) = 0.5$, $\Gamma(C_4) = \frac{1}{2*2}(\gamma(x_6)) = \frac{1}{4}(1) = 0.25$ e $\Gamma(C_5) = \frac{1}{2*2}(\gamma(x_8) + \gamma(x_9)) = \frac{1}{4}(1 + 1) = 0.5$. O valor dos outros clusters na arvore podem ser calculados analogamente.

2.2 Trabalhos relacionados

O estudo conduzido por Belkin, Niyogi e Sindhwani (2006) introduz uma família de algoritmos de aprendizado que incorporam uma forma de regularização, permitindo explorar a geometria da distribuição na margem. Essa abordagem cria uma estrutura semissupervisionada que abarca dados com e sem rótulos, abrindo caminho para um aprendizado de uso geral. A metodologia emprega tanto algoritmos de aprendizado de grafos transdutivos quanto métodos convencionais, incluindo máquinas de vetores de suporte e mínimos quadrados regularizados.

Os autores Lelis e Sander (2009) introduzem o SSDBSCAN (*Semi Supervised DBSCAN*), um algoritmo semissupervisionado de agrupamento de dados. Esse algoritmo é capaz de ajustar automaticamente os parâmetros de densidade para cada *cluster* natural em um conjunto de dados. O estudo descreve como instâncias rotuladas podem auxiliar o algoritmo na determinação adequada dos parâmetros de densidade para extrair *clusters* baseados em densidade em partes específicas do espaço de características Lelis e Sander (2009). A singularidade do SSDBSCAN reside na sua capacidade de operar com um único parâmetro robusto extraído do conjunto de dados, eliminando a necessidade de intervenção do usuário. Ele pode identificar automaticamente ruídos e compreender a estrutura intrínseca dos *clusters*, mesmo quando existe uma variação considerável em suas densidades.

Outra contribuição, trazida por Li *et al.* (2014), apresenta estratégias que colaboram com o SSDBSCAN. Apesar da robustez do algoritmo, que é capaz de extrair *clusters* de conjuntos de dados com diversos níveis de densidade, ele requer ao menos um rótulo para cada cluster natural

a partir dos dados fornecidos. O trabalho também introduz uma nova abordagem de aprendizado ativo para selecionar os objetos mais representativos dos conjuntos de dados, a fim de fornecê-los como entrada para o SSDBSCAN. Isso é realizado por meio do uso de um Regularizador de Grafos Laplaciano em um método de Reconstrução Linear Local.

Campello *et al.* (2015) introduzem uma estrutura unificada para a análise de *clusters* baseada em densidade, juntamente com detecção de valores discrepantes e visualização de dados. O HDBSCAN* é uma ferramenta projetada para calcular estimativas hierárquicas dos conjuntos de diferentes níveis de densidade. Esse algoritmo generaliza e aprimora as técnicas de agrupamento baseadas em densidade, resultando em uma hierarquia completa de agrupamentos que engloba todos os *clusters* possíveis relacionados à densidade. Esses resultados podem ser facilmente processados para permitir visualização e exploração dos dados.

O estudo realizado por Gertrudes *et al.* (2019) oferece uma investigação aprofundada sobre os algoritmos de agrupamento fundamentados na densidade dos dados. A pesquisa revela uma estreita conexão entre algoritmos de agrupamento baseados em densidade e uma abordagem de classificação transdutiva baseada em grafos. Com base nesses *insights*, foi elaborada uma nova estrutura para classificação semissupervisionada, centrada na densidade. Além disso, o estudo generaliza o algoritmo HDBSCAN* para uma aplicação de classificação semissupervisionada.

3 Proposta de desenvolvimento

O propósito deste capítulo é expor as etapas metodológicas que guiarão a realização deste estudo. Neste segmento, serão delineados os conjuntos de dados empregados, as abordagens de aplicação dos métodos de ligação, as medidas de desempenho empregadas, além do procedimento de avaliação dos resultados alcançados.

3.1 Conjunto de dados

As bases de dados que serão utilizadas estão sumarizadas na Tabela 3.1, com a descrição do conjunto, número de instâncias/objetos, número de atributos, número de classes, além da métrica que será aplicada em cada conjunto.

Vale ressaltar que esta coleção de dados também foi utilizada por (Gertrudes *et al.*, 2019) para fins de comparação entre algoritmos de aprendizado semissupervisionado. As coleções de dados artificiais foram obtidas de Handl e Knowles (2007) [1], contendo 160 conjuntos de dados sintéticos, sendo metade de baixa dimensão, 2D e 10D, e a outra metade de alta, 50D e 100D. Os conjuntos têm de 4 a 40 clusters, com tamanho de 10 a 500 objetos. Será utilizada distância euclidiana para essa coleção.

Assim como utilizado em (Gertrudes *et al.*, 2019), os conjuntos Articles-1442-5, Articles-1442-80, CellCycle-237 e Ecoli também serão utilizados no presente trabalho. Articles-1442-5 e Articles-1442-80 contém representações de alta dimensão de documentos de texto, formados por 253 artigos de 5 categorias cada artigo, representados por 4636 e 388 dimensões respectivamente, onde será utilizada similaridade de cosseno para esse conjunto. CellCycle-237 contém os níveis de expressão de 237 genes, com 17 dimensões, com 4 categorias, onde será utilizado distância euclidiana com normalização z-score. Ecoli pertence ao repositório UCI que contém 336 objetos com 7 dimensões e 8 classes, onde será utilizado a distância euclidiana.

A coleção ALOI contém características de imagens extraídas da Biblioteca de Imagens de Objetos de Amsterdã (ALOI - *Amsterdam Library of Object Images*). Esse conjunto de dados foi criado selecionando aleatoriamente C categorias de imagens como rótulo de classe, 100 vezes para cada $C = \{2, 3, 4, 5\}$, em seguida amostrando sem substituição 25 imagens de cada uma das C categorias selecionadas. Com um total de 400 conjuntos, cada um contendo de 2 a 5 classes, entre 50, 75, 100 ou 125 objetos (imagens). O ALOI-TS88 usa descritor de estatística de textura, já o ALOI-PCA representa os dados em 6 dimensões, combinando o primeiro componente principal extraído de cada um dos seis descritores, usando PCA. Para ambos, será utilizado a distância euclidiana.

[1] https://personalpages.manchester.ac.uk/staff/Julia.Handl/generators.html

Tabela 3.1 – Lista de conjuntos de dados coletados para realizar os experimentos de agrupamento semissupervisionados.

Conjunto	**#objetos**	**#atributos**	**#classes**	**Distância**
Reais				
Articles-1442-5	253	4636	5	Cosseno
Articles-1442-80	253	388	5	Cosseno
Bank note–Authentication	1372	5	2	Euclidiana
Cardiotocography	2126	36	10	Euclidiana
CellCycle-237	237	17	4	Euclidiana
Chowdary	104	183	2	Cosseno
Diggle table	310	8	9	Euclidiana
Ecoli	336	7	8	Euclidiana
Gordon	181	1627	2	Cosseno
Iris	150	5	3	Euclidiana
Mfeat-factors	2000	216	10	Euclidiana
Mfeat-Karhunen	2000	65	10	Euclidiana
Seeds	210	8	3	Euclidiana
Segmentation	2100	20	7	Euclidiana
Stock	950	10	2	Euclidiana
WDBC	569	32	2	Euclidiana
Wine	178	13	3	Euclidiana
Yeast galactose	205	81	4	Euclidiana
Coleções ALOI				
ALOI PCA	[50, 125]	6	[2, 5]	Euclidiana
ALOI 88	[50, 125]	88	[2, 5]	Euclidiana
Coleções Artificiais				
Gaussian (Handl; Knowles, 2007)	[200, 5000]	[2, 10]	[4, 40]	Euclidiana
Ellipsoid (Handl; Knowles, 2007)	[200, 5000]	[50, 100]	[4, 40]	Euclidiana

Fonte: Adaptado de Gertrudes *et al.* (2019), página 36.

Os demais conjuntos, são dados reais pertencentes a vários domínios diferentes, onde, ao menos um algoritmo de agrupamento foi capaz de alcançar uma solução com ARI de pelo menos 0.5, garantindo que as combinações envolvem conjuntos de dados onde é possível recuperar pelo menos parcialmente uma estrutura de agrupamento. Exceto as coleções artificiais, todos as demais bases podem ser encontradas no repositório disponibilizado por Gertrudes *et al.* (2019)[2].

3.2 Extração ótima de grupos

Durante os experimentos utilizaremos os três métodos de ligação descritos na Capítulo 2: *Single linkage*, *Average linkage* e *Complete linkage*. A árvore de grupos será extraída a partir do dendrograma gerado por esses métodos, utilizando como base o parâmetro m_{ClSize}. A princípio,

[2] https://github.com/jadsoncastro/UnifiedView/tree/master/data

utilizaremos o valor padrão reportado em (Campello *et al.*, 2013), com $m_{ClSize} = 4$. Entretanto, este parâmetro poderá ser modificado para fins de desempenho do FOSC.

Em relação ao FOSC, utilizaremos em cada árvore gerada a medida de qualidade baseada em rótulos B^3 F-measure para fins de extração ótima de grupos. Para fins de comparação, poderá ser utilizada também a medida baseada em restrições proposta por Campello *et al.* (2013).

Para a etapa de cortes locais nas árvores geradas, será utilizado o algoritmo FOSC, explicado anteriormente, com a métrica semisupervisionada baseada em rótulos e a métrica baseada em restrição,

3.3 Subconjuntos de dados pré-rotulados

A base de dados pré-rotulada, designada como $\mathbf{X}_L$, será construída ao selecionar, de forma aleatória, objetos rotulados dos conjuntos de dados de acordo com duas abordagens, tal como demonstrado por Gertrudes *et al.* (2019): seleção aleatória não controlada e seleção aleatória controlada. Na metodologia de seleção aleatória não controlada, os objetos serão retirados sem reposição, sem imposição de outras restrições, embora possa ocorrer a exclusão de certos rótulos de classe. Para tal, a proporção de objetos pré-rotulados variará entre 1%, 2% e 5% do conjunto de dados total. Por sua vez, na estratégia de seleção aleatória controlada, serão realizados experimentos ao escolher subconjuntos de dados que contenham todas as classes e, adicionalmente, $\lceil \frac{c}{2} \rceil$, o próximo número inteiro para a metade das classes. Novamente, as proporções de objetos pré-rotulados oscilarão entre 1%, 2% e 5% do total de dados disponíveis.

3.4 Medidas de desempenho

A medida de validação a ser utilizada nos experimentos será o índice Rand ajustado (ARI), proposto em Hubert e Arabie (1985). Este índice compara um conjunto de *clusters* gerados por um algoritmo de agrupamento e uma partição verdadeira do conjunto de dados. Na literatura de agrupamento de dados, este índice é considerado um índice de validação externa, pois utiliza de resultados externos para avaliar a qualidade de um agrupamento. Para o cálculo deste índice, são consideradas as seguintes variáveis: a que representam os objetos no mesmo *cluster* tanto no gerado pelo algoritmo de agrupamento quanto no *cluster* da partição verdadeira; b que representa os objetos que estão nos mesmos *clusters* na partição verdadeira mas em *clusters* distintos no resultado gerado pelo algoritmo de agrupamento; c que representa os objetos que estão em diferentes *clusters* na partição verdadeira mas nos mesmos *clusters* no resultado gerado pelo algoritmo de agrupamento e d que representam os objetos em *cluster* distintos tanto no gerado pelo algoritmo de agrupamento quanto no *cluster* da partição verdadeira;

Matematicamente, o índice de ARI é dado por

$$ARI = \frac{a - \frac{(a+c)(a+b)}{M}}{\frac{(a+c)+(a+b)}{2} - \frac{(a+c)(a+b)}{M}}, \tag{3.1}$$

onde $M = a + b + c + d = \frac{N(N-1)}{2}$. O ARI uma medida que se estende na faixa de -1 a 1. Valores próximos de zero, ou menores, indicam que qualquer similaridade entre as divisões de conjuntos ocorre devido a acasos aleatórios, enquanto o valor 1 denota que as divisões são exatamente iguais (Faceli *et al.*, 2021). Embora o índice possa teoricamente chegar ao mínimo de -1, na prática, esse ponto não é alcançado. Normalmente, quando as divisões são significativamente discrepantes, o índice tende a se aproximar de 0. Apesar de ser possível ter um limite inferior de 0 rigidamente estabelecido, a normalização necessária para tal refinamento não leva a nenhuma vantagem, uma vez que valores negativos não têm relevância nenhuma (Faceli *et al.*, 2021). O índice será calculado sem utilizar nenhum dos objetos que pertencem ao conjunto dos objetos pré-rotulados.

3.5 Fluxo de execução

Primeiramente os dados são carregados das bases de dados, seus rótulos são removidos e uma porcentagem dos rótulos são armazenadas em uma variável, seguindo o passo-a-passo apresentado na seção 3.3, vamos chama-la de X_L. Com os dados sem rotulo, eles são inseridos no algoritmo de ligação para a geração da arvore hierárquica, com as medidas de distancia *Single*, *Complete* ou *Average Linkage*. Apos a arvore pronta, o algoritmo FOSC é executado fazendo cortes nela, e nesse momento é escolhido qual medida de desempenho ele vai utilizar, sendo a baseada em rotulo, apresentada na subseção 2.1.5, ou a baseada em restrição, apresentada na subseção 2.1.6. Para o calculo da qualidade do *cluster*, é inserido junto ao algoritmo o conjunto de dados X_L, que contem os dados pré rotulados.

4 Resultados

4.1 Analise dos resultados

O teste de postos sinalizados de Wilcoxon (*Wilcoxon signed-ranks test*) é uma técnica estatística não paramétrica utilizada para determinar se há diferença significativa entre duas amostras independentes (Wilcoxon, 1947). Ele calcula a soma dos postos das observações de ambas as amostras e determina se a diferença entre essas somas é estatisticamente significativa. Este teste é uma alternativa ao teste T-pareado (Demšar, 2006) quando os pressupostos deste último não são atendidos, como no caso de dados que não seguem uma distribuição normal ou quando as variáveis são ordinais, além de apresentar uma perda de desempenho na presença de *outliers* (Demšar, 2006).

O método de Wilcoxon, da API *scipy*[1], foi escolhido por se tratar de um teste comum na literatura (Faceli *et al.*, 2021), além de se encaixar nas condições dos resultados gerado pelos algoritmos deste trabalho, como a possibilidade de *outliers*, devido ao uso de aleatoriedade na geração dos dados pré rotulados, além dos resultados não seguirem uma distribuição, como a normal.

Para avaliar o comportamento em tempo de execução dos algoritmos, eles foram executados em uma máquina Linux Intel i9-10900 com 10 cores físicos (20 *threads*) de 2.80GHz e 128GB RAM DDR4. O algoritmo inteiro foi implementado em Python, e executado na versão 3.10.8.

Os resultado que serão apresentados a seguir foram obtidos após a execução do algoritmo em quatro cenários: com 31 bases de dados reais em ambiente supervisionado e não supervisionado, e com 105 bases de dados artificias em ambiente supervisionado e não supervisionado. O algoritmo contem o fator de aleatoriedade apresentado em seção 3.3, logo foi interessante calcular a média de 30 resultados para cada combinação de parâmetros, exemplo, para a execução com a medida de ligação *Single-Linkage*, 1% dos dados rotulado, em ambiente controlado, para a medida de qualidade *BCubed*, foram executadas 30 vezes e retornado a média dos ARIs.

4.2 Bases de dados reais

O primeiro ambiente a ser analisado é o Real Controlado. Os resultados primeiramente foram colocados em um *BoxPlot* para visualização, como na Figura 4.1, onde cada sub imagem contem uma comparação entre os resultados das medidas de qualidade *BCubed*(em azul a esquerda) e baseada em restrição (*Constraint*, em laranja na direita). Organizado em uma tabela

[1] disponivel em https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html

3 por 3, as linhas selecionam a medida de ligação, e as colunas a porcentagem de dados pré rotulados. Abaixo de cada sub imagem contem um valor indicando o resultado do teste de Wilcoxon, apresentado na seção 4.1, onde o valor em verde, menor que 0.05, indica que existe uma diferença significativa entre os resultados, e em vermelho, maior que 0.05, indica que não existe uma diferença significativa entre os resultados que comprove que *BCubed* é melhor que *Constraint*.

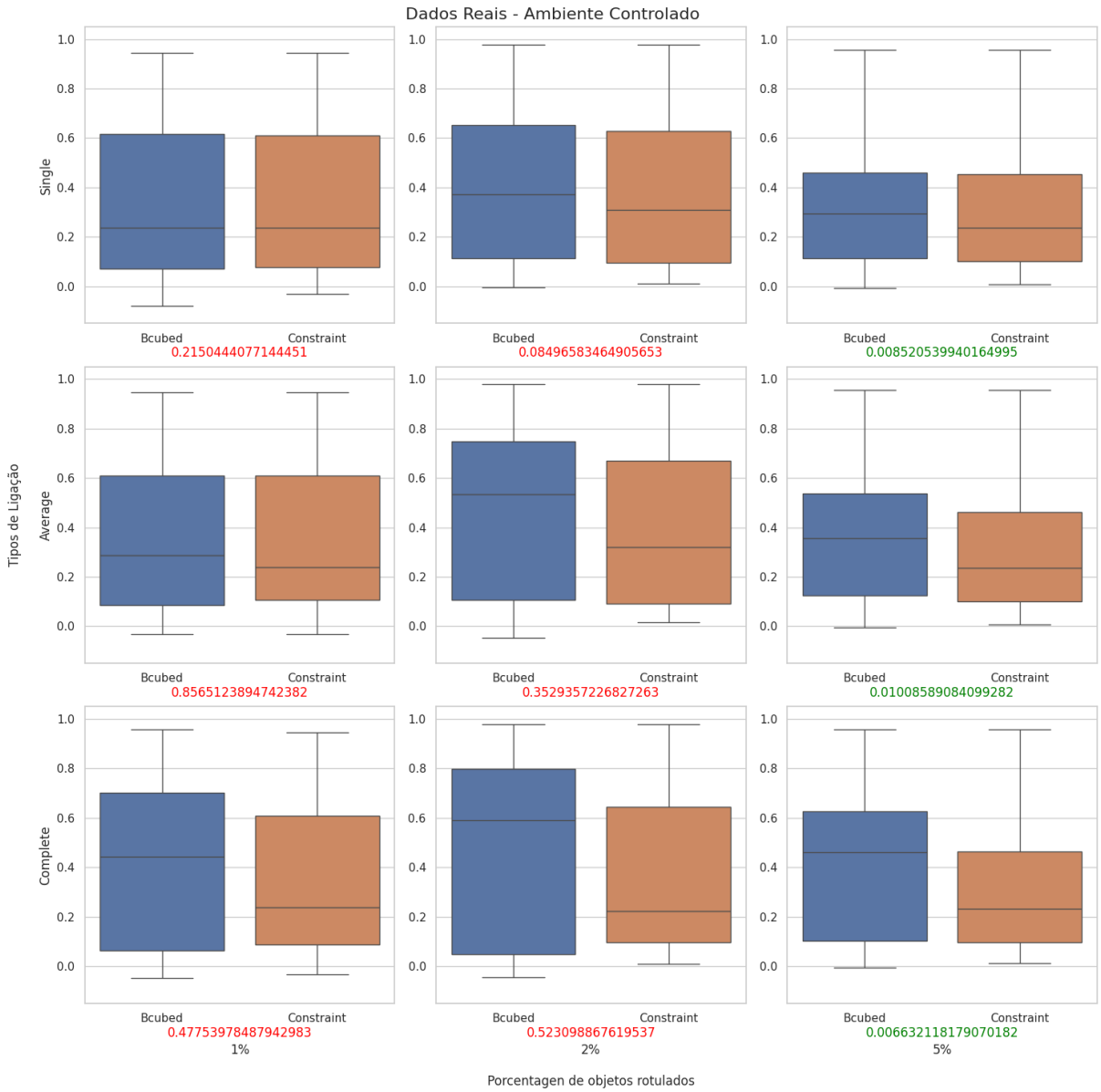

Figura 4.1 – Visualização em *BoxPlot* dos resultados da base de dados real em ambiente controlado

Fonte: Elaborado pelo autor

Com a analise dos resultados da Figura 4.1, foi possível perceber que apenas as execuções com 5% dos dados rotulados obtiveram uma diferença significativa, independente da medida de ligação. Percebendo isso, optou-se por juntar o resultado em 3 grupos divididos pela porcentagem, 1%, 2% e 5%. Para cada porcentagem existem 3 resultados, *Single*, *Complete* e *Average-Linkage*,

para cada base de dados, foi então decidido escolher o maior entre os 3 para representar o resultado. A Figura 4.5 apresenta como ficou a distribuição dos resultados.

Analisando a primeira linha da Figura 4.5, percebe-se que em todos os casos houve diferença significativa, perceptível ao comparar as médias de cada resultado. Para uma ultima analise, os 3 grupos presentes nos resultados foram concatenados, geram assim dois vetores principais a serem comparados. O resultado final pode ser visto na Figura 4.6

Para o caso do ambiente real não controlado, os resultados foram similares, com vários casos de diferença e sem diferença na primeira analise e a analise final com diferença, porem na analise dividida por porcentagens, apenas o caso de 5% obteve diferença. As imagens pode ser vistas na Figura 4.2, Figura 4.5 e Figura 4.6

4.3 Bases de dados artificiais

Para a execução dos dados artificias, foram feitos testes similares aos dados reais. A Figura 4.3 ilustra as comparações do resultados para as bases de dados artificiais em ambiente controlado, e é perceptível em todos os casos duas coisas: em todos os casos ouve uma diferença significativa entre os resultados dos algoritmos, e todos se mantiveram com as médias do ARIs acima de 0.5.

Ao analisar a terceira linha da Figura 4.5, temos uma analise dividida por porcentagens, e todos os casos foram similares a analise anterior. Por fim, ao observar o resultado final na Figura 4.6, os resultados obtiveram uma diferença significativa como em todos os outros casos.

Para o caso do ambiente artificial não controlado, os resultados foram similares, com todos os casos com diferença na primeira analise e a analise final. As imagens pode ser vistas na Figura 4.4, Figura 4.5 e Figura 4.6

4.4 Resultados gerais

A Figura 4.6 apresenta o resultado geral, onde segue uma escala vertical de 0 a 1 pois o valor de ARI vai de -1 a 1, porem como quase nenhum valor deu negativo, a escala de -1 a 0 foi omitida.

4.5 Analise Especifica

Devido aos resultados apresentados pelas bases reais terem sidos não muito bons em relação a base artificial, foi decidido então analisar mais separadamente os dados reais. Foi feito uma analise na quantidade de linhas e colunas de cada arquivo na base onde, apenas 6 dos 31 arquivos tinham mais de 1000 linhas de dados, e 12 dos 31 tinham mais de mil colunas, sendo o

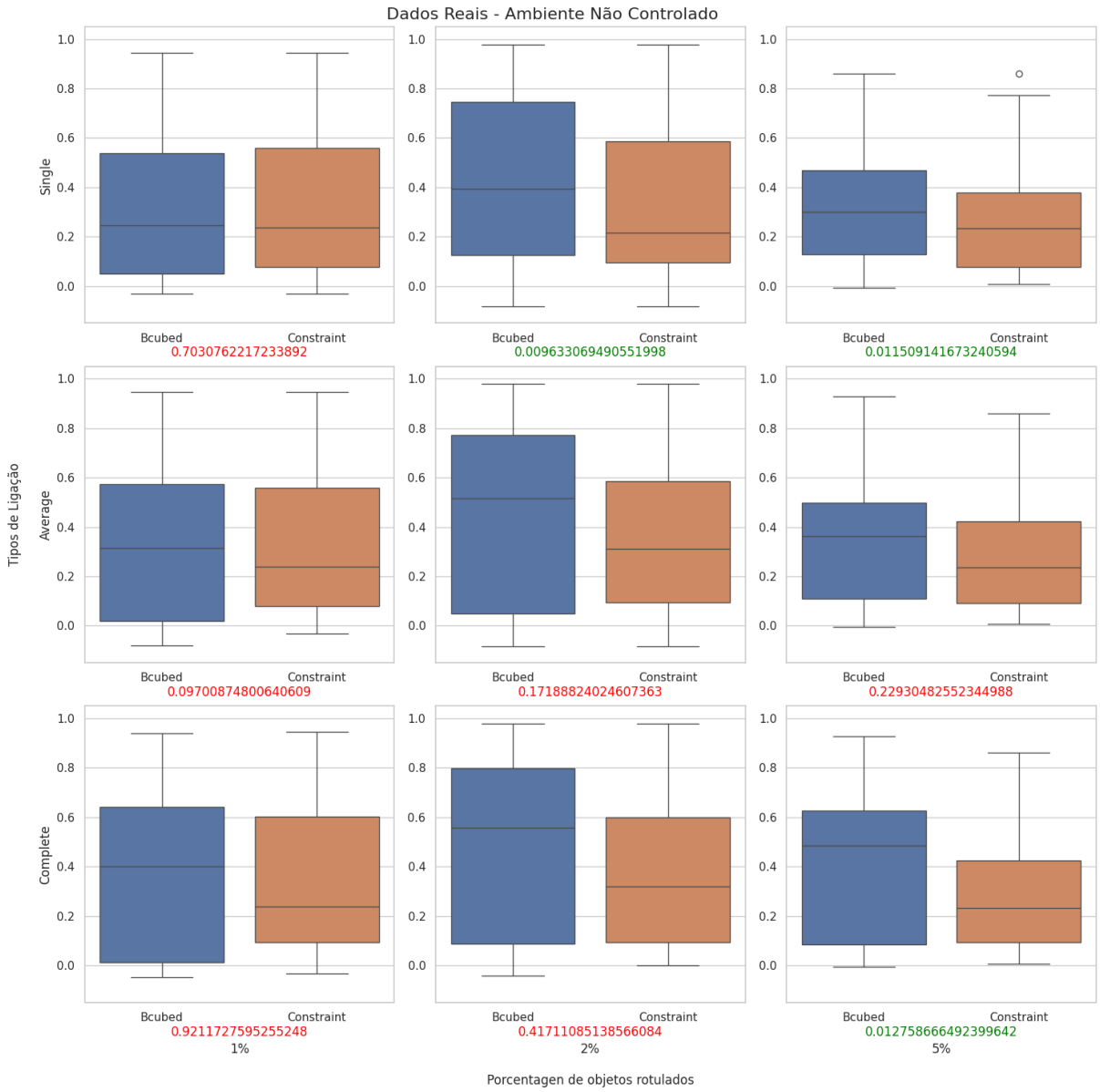

Figura 4.2 – Visualização em *BoxPlot* dos resultados da base de dados real em ambiente não controlado

Fonte: Elaborado pelo autor

articles_1442_5 o maior, com 4637 colunas. Primeiramente foram analisados as diferenças de colunas, apresentados na Figura 4.7.

Foi possível observar que as bases de dados com menos de mil colunas obtiveram resultados melhores, em relação as bases com mais de mil colunas, esse fato também foi observado para a base de dados artificial, pois nenhum arquivo dela tem mais de 100 colunas.

Sobre a quantidade de linhas, a Figura 4.8 apresenta os resultados para essa analise mais aprofundada. Nela pode-se notar que, apesar das bases com menos de mil linhas também terem alcançados alguns resultados de ARI próximos de 1, as bases com mais de mil linhas tiveram seus resultados mais concentrados próximo de 1. Esse fato também pode ser observado para as

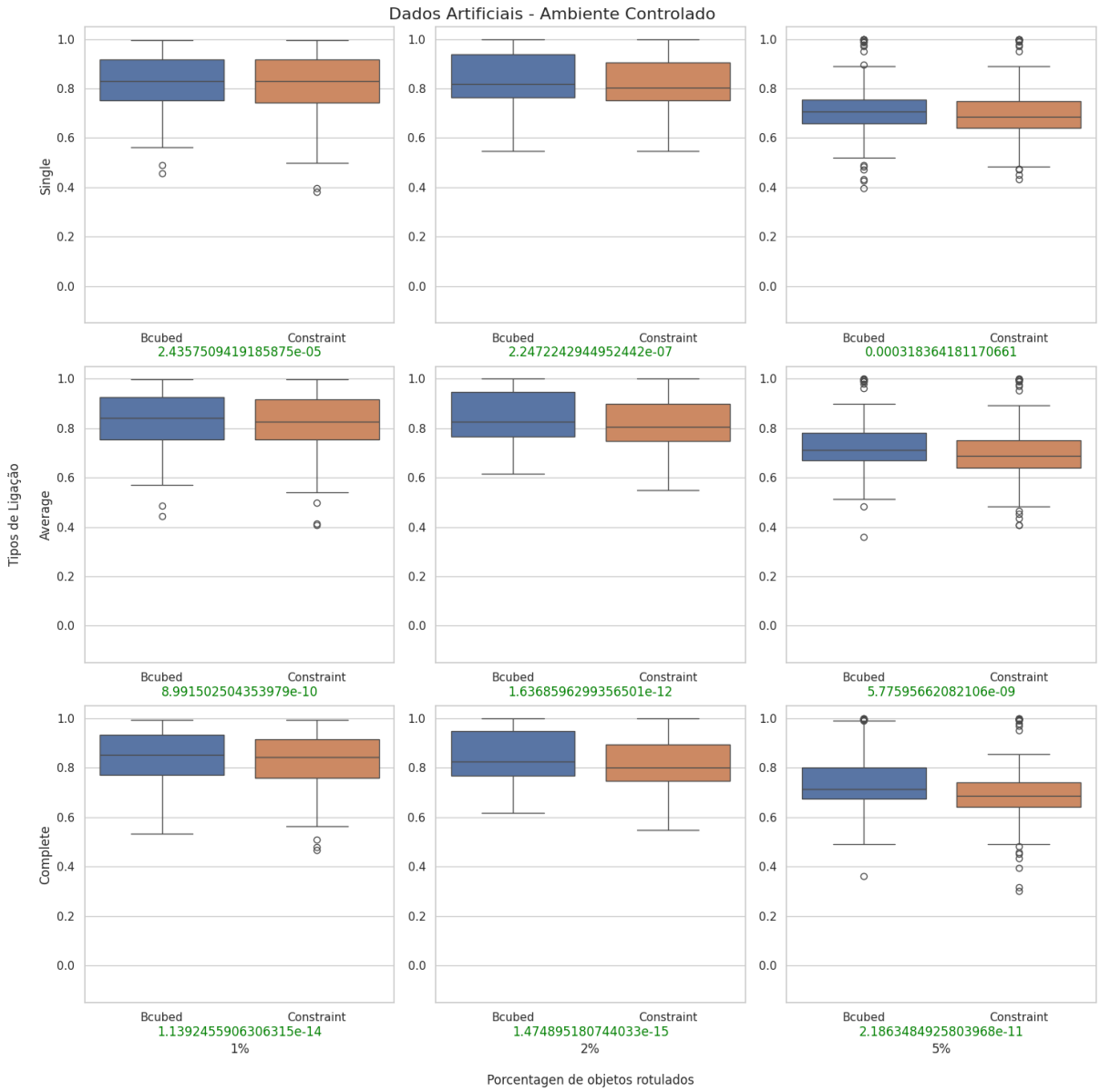

Figura 4.3 – Visualização em *BoxPlot* dos resultados da base de dados artificial em ambiente controlado

Fonte: Elaborado pelo autor

bases artificiais, pois elas tem mais de mil linhas em cada arquivo.

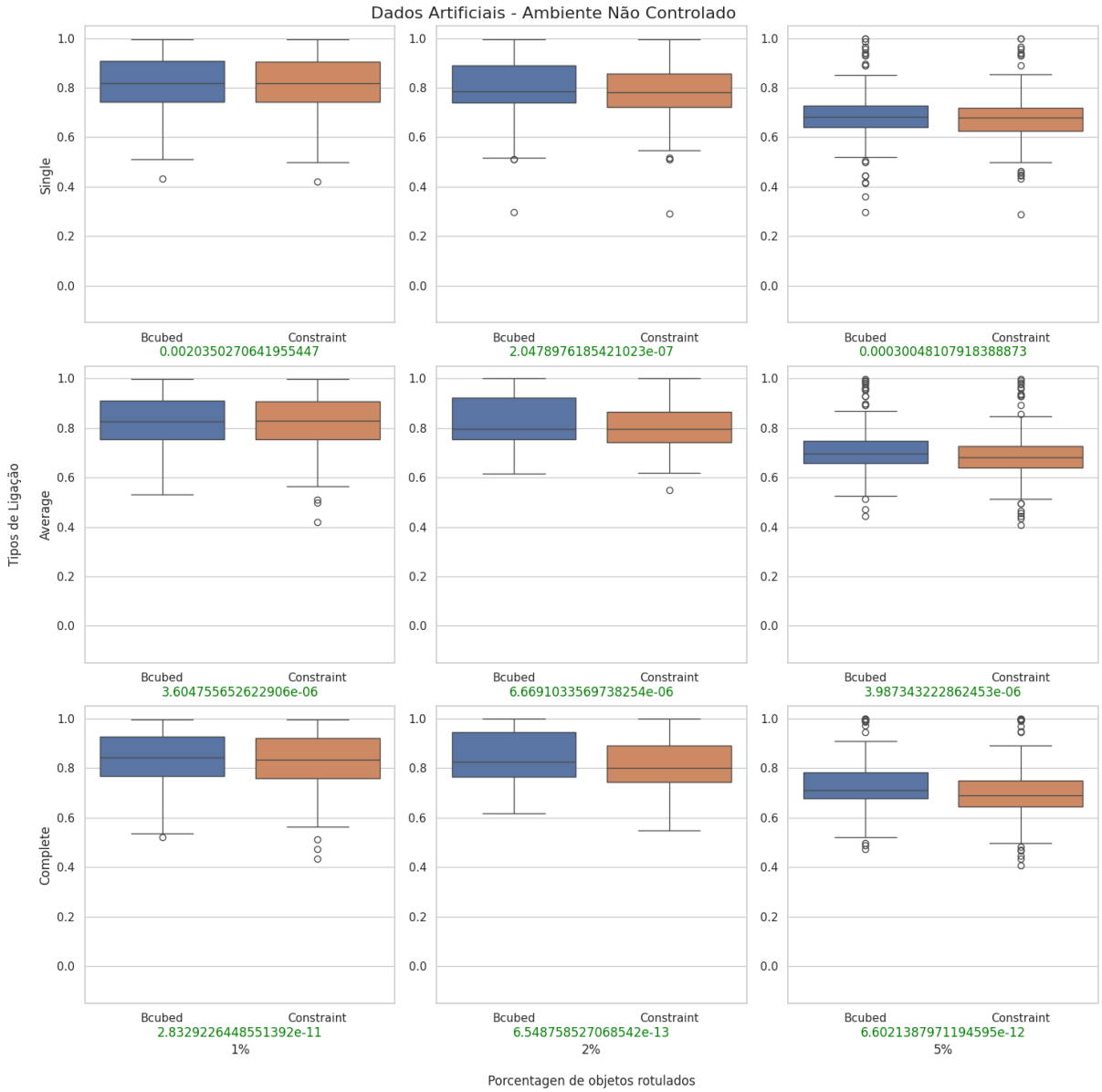

Figura 4.4 – Visualização em *BoxPlot* dos resultados da base de dados artificial em ambiente não controlado

Fonte: Elaborado pelo autor

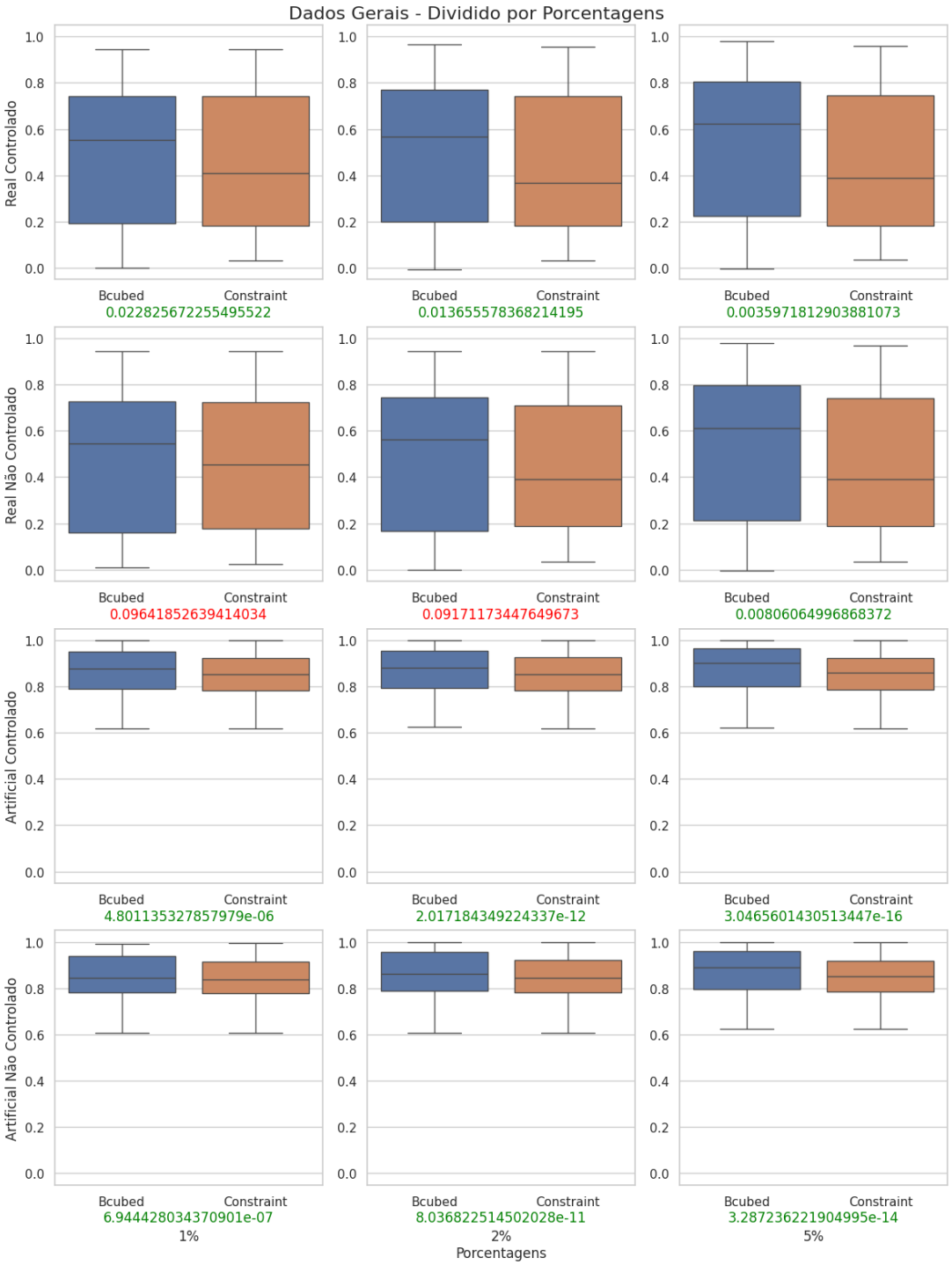

Figura 4.5 – Visualização em *BoxPlot* dos resultados gerais dividido por porcentagem

Fonte: Elaborado pelo autor

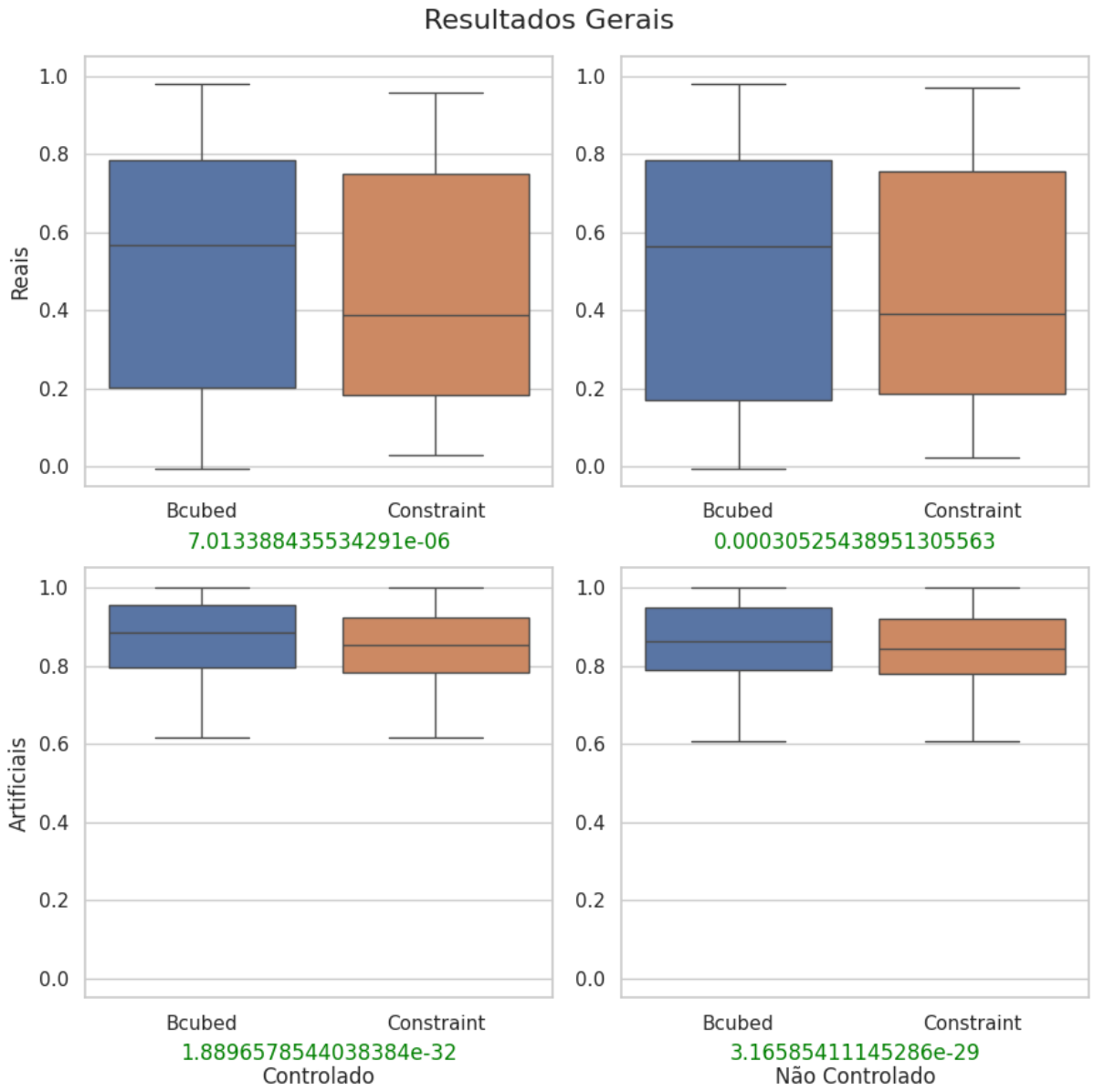

Figura 4.6 – Visualização em *BoxPlot* dos resultados gerais

Fonte: Elaborado pelo autor

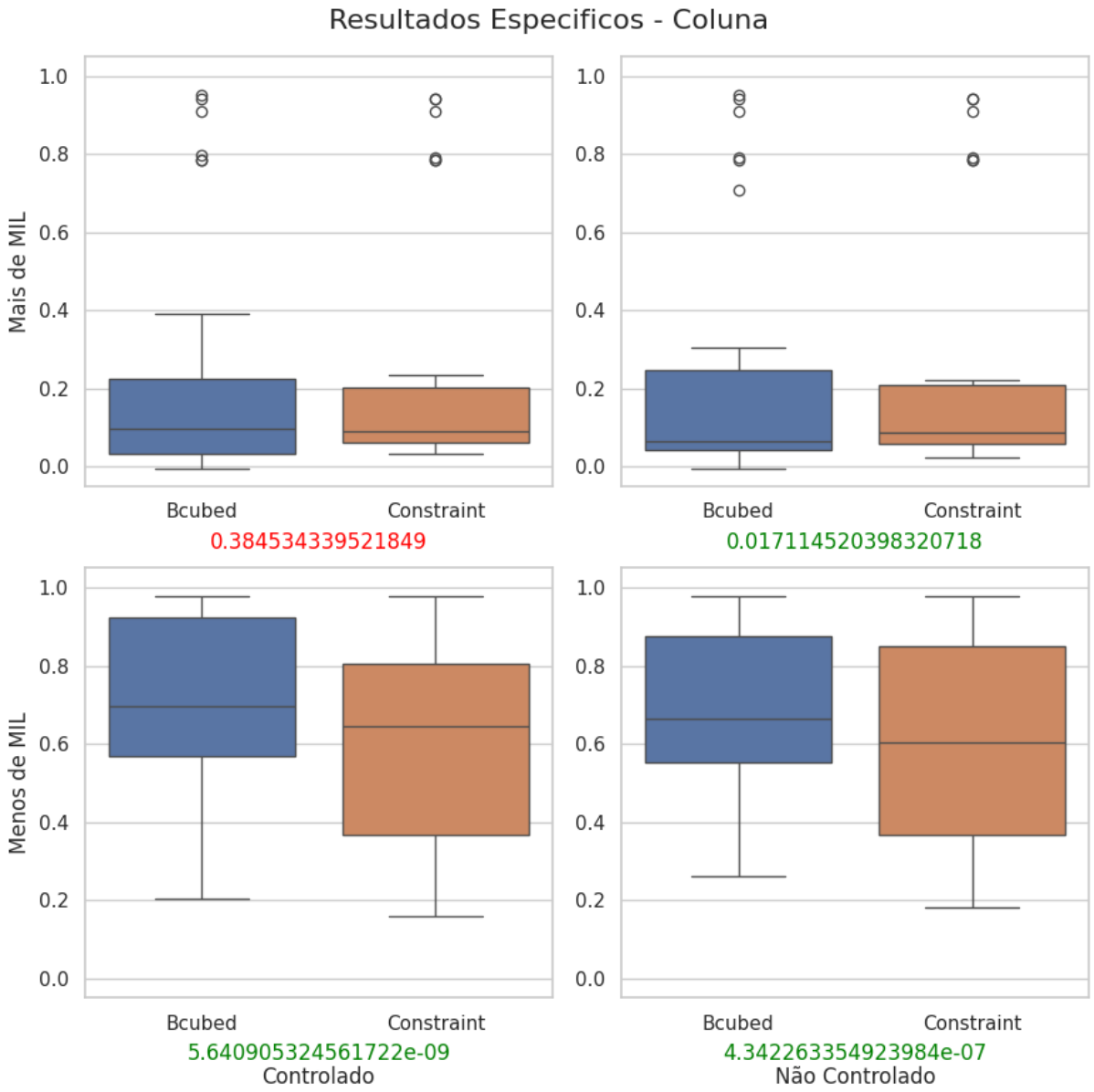

Figura 4.7 – Visualização em *BoxPlot* dos resultados específicos por coluna

Fonte: Elaborado pelo autor

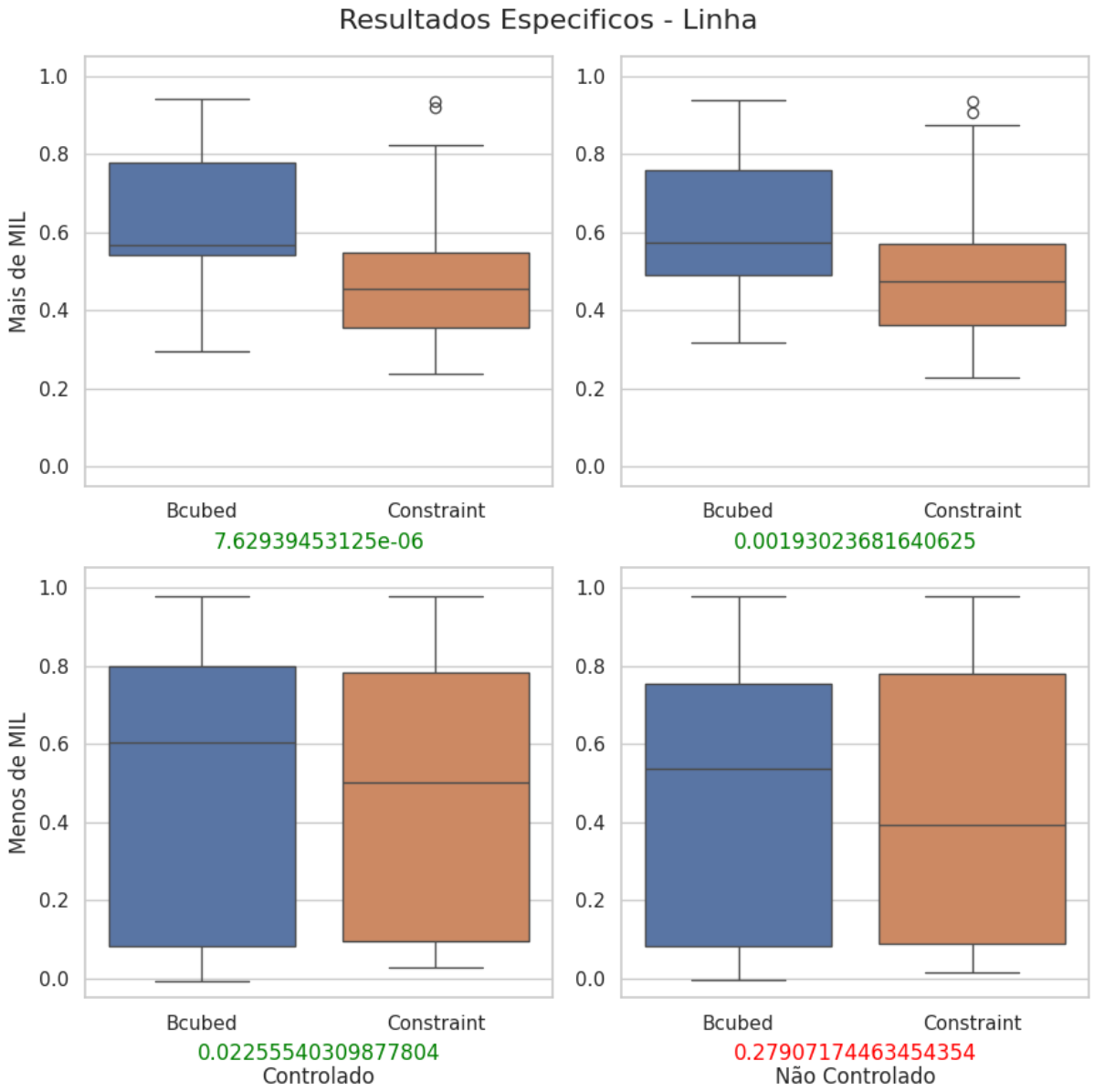

Figura 4.8 – Visualização em *BoxPlot* dos resultados específicos por linhas

Fonte: Elaborado pelo autor

5 Considerações Finais

5.1 Conclusão

Neste trabalho foi descrito os métodos para extração de agrupamento em conjunto de dados de maneira semissupervisionada. A classificação de dados não totalmente rotulados é de suma importância pois a maioria dos dados rotulados encontrados é incrivelmente pequena e cara, enquanto a quantidade da dados pouco ou não rotulados é muito maior (Chapelle; Schölkopf; Zien, 2006).

Nos métodos descritos, houve um enfoque maior nos métodos hierárquicos, algoritmos de corte locais e métricas de qualidade de *clusters*. Em especial foi estudado os métodos hierárquicos clássicos *single*, *complete*, e *average linkage*, o algoritmo FOSC para extração de clusters planos a partir de cortes locais em uma árvore de hierarquia, a métrica de analise de qualidade de clusters baseada em B^3 *Precision* e B^3 *Recall*, além da métrica baseada em restrição proposta em (Campello *et al.*, 2013).

Sobre os resultados obtidos, em sua grande maioria a métrica semi supervisionada baseada em rótulos apresentada na subseção 2.1.5 apresentou resultados similares aos da métrica baseada em restrição apresentada na subseção 2.1.6, o que dificultou a conclusão de qual algoritmo teve um desempenho melhor. Para isso foi feito o teste de postos sinalizados de Wilcoxon, apresentado na seção 4.1. Através dele foi possível concluir que a métrica de extração baseada em rótulos, nomeada como *BCubed*, obteve melhores resultados em relação a métrica baseada em restrição, pois, em quase todas os testes estatísticos feitos, ele teve uma diferença significativa ao seu favor.

Sobre cada tipo de base, as execuções em conjunto de dados reais tiveram resultados inferiores, principalmente no ambiente não controlado, pois suas bases de dados, em sua grande maioria, eram, ou pequenos e com poucos atributos, ou grandes e com mais de mil atributos, analise apresentada na seção 4.5, fato esse que explica o motivo dos testes em bases artificias terem sido melhores, pois elas são bases grandes com menos de mil colunas.

5.2 Trabalhos futuros

Para buscar uma solução com dados reais, seria necessário bases de dados com algumas características, como poucas colunas e muitas linhas. Para as bases que não se encaixam nessas características, seria interessante a aplicação de técnicas de redução de dimensionalidade, como *Autoencoders*, Análise de Componentes Principais (PCA), Regularização L1, entre outros. Além de que, seria instigante fazer comparações entre a métrica baseada em rótulo com outros algoritmos de clusterização, como o K-Means, OPTICS, e as medidas *Weighted* e *Centroid-Linkage*.

Referências

ALTERYX. Aprendizado supervisionado vs. não supervisionado. fevereiro 2024. Disponível em: <https://www.alteryx.com/pt-br/glossary/supervised-vs-unsupervised-learning#:~:text=%C3%80s%20vezes%2C%20o%20aprendizado%20n%C3%A3o,humana%20para%20validar%20os%20valores.>

BAGGA, A.; BALDWIN, B. Entity-based cross-document coreferencing using the vector space model. In: **COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics**. [s.n.], 1998. Disponível em: <https://sci-hub.se/10.3115/980845.980859>.

BATISTA, A. J. L.; CAMPELLO, R. J. G. B.; SANDER, J. Active semi-supervised classification based on multiple clustering hierarchies. In: **Proc. DSAA**. [S.l.: s.n.], 2016. p. 11–20.

BELKIN, M.; NIYOGI, P.; SINDHWANI, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. **Journal of machine learning research**, v. 7, n. 11, 2006.

CAMPELLO, R. J. *et al.* A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies. **Data Mining and Knowledge Discovery**, Springer, v. 27, p. 344–371, 2013.

CAMPELLO, R. J. G. B. *et al.* Hierarchical density estimates for data clustering, visualization, and outlier detection. **ACM TKDD**, v. 10, n. 1, p. 1–51, 2015.

CHAPELLE, O.; SCHÖLKOPF, B.; ZIEN, A. Introduction to semi-supervised learning. MIT press, 2006.

CHAPELLE, O.; SCHöLKOPF, B.; ZIEN, A. (Ed.). **Semi-supervised learning**. Cambridge, MA: MIT Press, 2006.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. **The Journal of Machine learning research**, JMLR. org, v. 7, p. 1–30, 2006.

ESTER, M. *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. In: **Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA**. AAAI Press, 1996. p. 226–231. Disponível em: <http://www.aaai.org/Library/KDD/1996/kdd96-037.php>.

FACELI, K. *et al.* **Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina**. [s.n.], 2021. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788521637509>.

GERTRUDES, J. C. *et al.* A unified view of density-based methods for semi-supervised clustering and classification. **Data mining and knowledge discovery**, Springer, v. 33, p. 1894–1952, 2019.

HANDL, J.; KNOWLES, J. An evolutionary approach to multiobjective clustering. **IEEE transactions on Evolutionary Computation**, IEEE, v. 11, n. 1, p. 56–76, 2007.

HUBERT, L.; ARABIE, P. Comparing partitions. **Journal of classification**, Springer, v. 2, p. 193–218, 1985.

JARMAN, A. M. Hierarchical cluster analysis: Comparison of single linkage, complete linkage, average linkage and centroid linkage method. **Georgia Southern University**, v. 29, 2020.

LELIS, L.; SANDER, J. Semi-supervised density-based clustering. In: IEEE. **2009 Ninth IEEE International Conference on Data Mining**. [S.l.], 2009. p. 842–847.

LI, J. *et al.* Active learning strategies for semi-supervised dbscan. In: SPRINGER. **Advances in Artificial Intelligence: 27th Canadian Conference on Artificial Intelligence, Canadian AI 2014, Montréal, QC, Canada, May 6-9, 2014. Proceedings 27**. [S.l.], 2014. p. 179–190.

MACQUEEN, J. B. Some methods for classification and analysis of MultiVariate observations. In: CAM, L. M. L.; NEYMAN, J. (Ed.). **Proc. of the 5th berkeley symposium on mathematical statistics and probability**. [S.l.]: University of California Press, 1967. v. 1, p. 281–297.

TAN, P.-N.; STEINBACH, M. S.; KUMAR, V. **Introduction to data mining**. Addison-Wesley, 2005. ISBN 0-321-32136-7. Disponível em: <http://www-users.cs.umn.edu/%7Ekumar/dmbook/>.

WILCOXON, F. Probability tables for individual comparisons by ranking methods. **Biometrics**, JSTOR, v. 3, n. 3, p. 119–122, 1947.

WU, X. *et al.* Top 10 algorithms in data mining. **Knowl. Inf. Syst.**, v. 14, n. 1, p. 1–37, 2008. Disponível em: <https://doi.org/10.1007/s10115-007-0114-2>.