Classificação semi-supervisionada ativa baseada
em múltiplas hierarquias de agrupamento

Antônio José de Lima Batista

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Antônio José de Lima Batista

Classificação semi-supervisionada ativa baseada em múltiplas hierarquias de agrupamento

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 2016

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

B333c

Batista, Antônio José de Lima

Classificação semi-supervisionada ativa baseada em
múltiplas hierarquias de agrupamento / Antônio José
de Lima Batista; orientador Ricardo José Gabrielli
Barreto Campello. - São Carlos - SP, 2016.
99 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, 2016.

1. Aprendizado de Máquina. 2. Aprendizado
Ativo. 3. Classificação Ativa Semi-Supervisionada.
4. Hierarquia de Grupos. I. Campello, Ricardo José
Gabrielli Barreto, orient. II. Título.

Antônio José de Lima Batista

**Active semi-supervised classification based on multiple
clustering hierarchies**

Master dissertation submitted to the Instituto de
Ciências Matemáticas e de Computação – 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 2016

À minha família!

À minha mãe Deodata e ao meu pai Antônio,

Às minhas irmãs Ana Paula e Ana Gabriela e ao meu irmão Antônio Carlos.

À minha pequena sobrinha Cecília.

Aos amigos Everaldo Gomes, Antônio Gustavo e Fabrício Torres.

*Agradeço o apoio, o incentivo, a força nos momentos mais difíceis
e por saber que sempre terei com quem contar.*

Obrigado!

AGRADECIMENTOS

À minha família pelo apoio, incentivo e ajuda nos momentos difíceis.

Ao professor Ricardo Campello pela orientação no desenvolvimento deste trabalho, pela paciência e pelos valiosos ensinamentos que foram fundamentais para o meu crescimento pessoal e profissional.

Ao professor Jörg Sander da Universidade de Alberta pelas discussões e contribuições para o desenvolvimento deste trabalho.

Aos professores do Instituto de Ciências Matemáticas e de Computação (ICMC) pelos ensinamentos.

Aos professores Estevam Rafael Hruschka Júnior e Rodrigo Coelho Barros pelas contribuições ao trabalho.

À Universidade de São Paulo e ao ICMC pelo ambiente de estudo adequado à minha formação.

Aos colegas integrantes do Laboratório de Análise de Padrões em Dados (LAPaD) por momentos de discussões e contribuições.

Ao apoio de todos os amigos que participaram dessa trajetória.

À agência de fomento FAPESP pela bolsa de mestrado.

RESUMO

BATISTA, A. J. L.. **Classificação semi-supervisionada ativa baseada em múltiplas hierarquias de agrupamento**. 2016. 99 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Algoritmos de aprendizado semi-supervisionado ativo podem se configurar como ferramentas úteis em cenários práticos em que os dados são numerosamente obtidos, mas atribuir seus respectivos rótulos de classe se configura como uma tarefa custosa/difícil. A literatura em aprendizado ativo destaca diversos algoritmos, este trabalho partiu do tradicional *Hierarchical Sampling* estabelecido para operar sobre hierarquias de grupos. As características de tal algoritmo o coloca à frente de outros métodos ativos, entretanto o mesmo ainda apresenta algumas dificuldades. A fim de aprimorá-lo e contornar suas principais dificuldades, incluindo sua sensibilidade na escolha particular de uma hierarquia de grupos como entrada, este trabalho propôs estratégias que possibilitaram melhorar o algoritmo na sua forma original e diante de variantes propostas na literatura. Os experimentos em diferentes bases de dados reais mostraram que o algoritmo proposto neste trabalho é capaz de superar e competir em qualidade dentro do cenário de classificação ativa com outros algoritmos ativos da literatura.

Palavras-chave: Aprendizado de Máquina, Aprendizado Ativo, Classificação Ativa Semi-Supervisionada, Hierarquia de Grupos.

ABSTRACT

BATISTA, A. J. L.. **Classificação semi-supervisionada ativa baseada em múltiplas hierarquias de agrupamento**. 2016. 99 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Active semi-supervised learning can play an important role in classification scenarios in which labeled data are laborious and/or expensive to obtain, while unlabeled data are numerous and can be easily acquired. There are many active algorithms in the literature and this work focuses on an active semi-supervised algorithm that can be driven by clustering hierarchy, the well-known Hierarchical Sampling (HS) algorithm. This work takes as a starting point the original Hierarchical Sampling algorithm and perform changes in different aspects of the original algorithm in order to tackle its main drawbacks, including its sensitivity to the choice of a single particular hierarchy. Experimental results over many real datasets show that the proposed algorithm performs superior or competitive when compared to a number of state-of-the-art algorithms for active semi-supervised classification.

Key-words: Machine Learning, Active Learning, Active Semi-Supervised Classification, Clustering Hierarchies.

LISTA DE ILUSTRAÇÕES

Figura 1 – Viés de Amostragem.	29
Figura 2 – Classificador f_n obtido a partir das observações rotuladas (representadas pelos círculos preenchidos).	31
Figura 3 – Exemplo de uma Hierarquia de Grupos sobre 5 observações	36
Figura 4 – Exemplo de Dendrograma. Sob a abordagem aglomerativa, a escala indica a união dos grupos.	37
Figura 5 – Exemplos de aplicações em dendrogramas	38
Figura 6 – Hierarquia <i>Ward's</i> para grupos com formatos esféricos	41
Figura 7 – Hierarquia <i>Ward's</i> para grupos com formatos arbitrários	41
Figura 8 – Exemplo de partição obtida sobre um corte horizontal na hierarquia <i>Complete-Linkage</i>	44
Figura 9 – Exemplo de partição obtida sobre um corte horizontal na hierarquia <i>Average-Linkage</i>	45
Figura 10 – <i>Single-Linkage</i> e o conhecido " <i>chaining effect</i> ". Em (c) as poucas observações encadeadas destacadas conectam regiões distantes produzindo soluções de agrupamento inesperadas.	46
Figura 11 – Exemplo para ilustrar o <i>Single-Linkage</i> sob a perspectiva de algoritmos em grafos.	47
Figura 12 – Em (a) as observações $\mathbf{x}_p, \mathbf{x}_q$ e $\mathbf{x}_r$ são <i>cores</i> , enquanto que $\mathbf{x}_s$ é <i>noise</i> ; $\mathbf{x}_q$ e $\mathbf{x}_r$ são ε -alcançáveis e $\mathbf{x}_p$ e $\mathbf{x}_q$ são densamente conectados. Em (b) são apresentadas as noções de Distância <i>core</i> e de Distância de Alcançabilidade Mútua.	49
Figura 13 – AGM no espaço de $d_{alcm}(\cdot, \cdot)$ com as <i>self</i> -arestas, d_{core} , de cada observação.	50
Figura 14 – Dendrograma correspondente à hierarquia HDBSCAN* para $m_{pts} = 3$ aplicando o Algoritmo 2 sobre a AGM estendida da Figura 13.	51
Figura 15 – Exemplo de observações e seu respectivo dendrograma <i>Ward's</i>	52
Figura 16 – Exemplo de solução do algoritmo HS. Em (a) a distribuição das observações considerando um problema de classificação com 5 possíveis classes; em (b) a partição encontrada pelo HS, as cores diferenciam os rótulos de classes definidos para cada grupo pelo algoritmo; em (c) a solução final do HS, as observações em preto foram consultadas ao oráculo, as demais têm suas saídas definidas pelos rótulos de seus grupos correspondentes.	56

Figura 17 – Exemplo de solução do algoritmo HS. Em (a) a distribuição das observações conforme apresentadas anteriormente pela Figura 1, em (b) a partição encontrada pelo HS que permite contornar o problema do viés de amostragem.	60
Figura 18 – Exemplo de solução do algoritmo HS que se trata de um corte não horizontal na hierarquia de grupos.	60
Figura 19 – Exemplo de solução do algoritmo <i>CLT</i>	61
Figura 20 – Exemplo de solução obtida pelo HS sobre a hierarquia com 35 observações. A partição final $\mathbf{P}(t)$ após $t = 8$ é dada pelos grupos 67, 66 e 64 (círculos destacados em negrito). Os valores são identificadores para os grupos, sendo os grupos folhas com identificadores correspondentes as observações. As cores correspondem as classes do problema (Vermelho, Verde, Azul e Amarelo) e estão destacados para aqueles grupos que estabelecem um rótulo estritamente representativo. Os pares em $\mathcal{A}(t)$ são apresentados e aqueles destacados em negrito representam as consultas.	64
Figura 21 – Desempenho do HS sobre os mesmos conjuntos de dados, mas a partir de diferentes hierarquias.	65
Figura 22 – Exemplo de hierarquia de grupos simplificada.	67
Figura 23 – Hierarquia simplificada representando os grupos virtuais.	68
Figura 24 – Exemplo de observações candidatas a seleção a partir de hierarquias simplificadas HDBSCAN*.	68
Figura 25 – Observações e seus rótulos de classe definidos pelas cores.	69
Figura 26 – Hierarquia <i>Ward's</i> . Os grupos são identificados pelos valores numéricos. . .	70
Figura 27 – Hierarquias Simplificadas <i>Ward's</i> obtidas a partir do dendrograma da Figura 26.	70
Figura 28 – Observações candidatas em branco e observações em nós virtuais destacadas em vermelho.	71
Figura 29 – Erro de Transdução após cada consulta sobre a hierarquia <i>Ward's</i>	75
Figura 30 – Erro de Transdução após cada consulta sobre a hierarquia <i>Single Linkage</i> (SL)	76
Figura 31 – Erro de Transdução após cada consulta sobre a hierarquia <i>Complete Linkage</i> (CL)	77
Figura 32 – Erro de Transdução após cada consulta sobre a hierarquia <i>Average Linkage</i> (AL)	78
Figura 33 – Erros de Transdução para diferentes hierarquias	79
Figura 34 – Erros de Transdução para diferentes hierarquias com hierarquia de densidade e simplificada	81
Figura 35 – Resultados de Classificação Indutiva	83
Figura 36 – Resultados de Classificação Indutiva com hierarquia de densidade e simplificada	84

LISTA DE ALGORITMOS

Algoritmo 1 – Algoritmo Hierárquico Aglomerativo	37
Algoritmo 2 – Algoritmo <i>Single-Linkage</i> a partir de uma AGM	47
Algoritmo 3 – Algoritmo HS	59

SUMÁRIO

1	INTRODUÇÃO	19
1.1	Contextualização	19
1.2	Motivação	20
1.3	Objetivos Gerais e Específicos	21
1.4	Estrutura	22
2	APRENDIZADO DE MÁQUINA	23
2.1	Aprendizado Indutivo e Suas Abordagens	23
2.2	Aprendizado Semi-Supervisionado	25
2.3	Aprendizado Ativo	27
2.3.1	<i>Amostragem Informativa</i>	28
2.3.2	<i>Amostragem Representativa</i>	30
2.3.3	<i>Amostragem Baseada em Pré-Agrupamento</i>	32
2.4	Conclusão	33
3	HIERARQUIAS DE GRUPOS	35
3.1	Algoritmos Hierárquicos	35
3.1.1	<i>Ward's</i>	38
3.1.2	<i>Average-Linkage</i>	41
3.1.3	<i>Complete-Linkage</i>	43
3.1.4	<i>Single-Linkage</i>	45
3.1.5	<i>HDBSCAN*</i>	48
3.1.6	<i>Hierarquias Simplificadas</i>	51
3.2	Conclusão	53
4	CLASSIFICAÇÃO ATIVA COM HIERARQUIAS DE GRUPOS . . .	55
4.1	Amostragem Hierárquica	55
4.1.1	<i>Determinando Rótulos Representativos</i>	56
4.1.2	<i>Determinando uma Partição na Hierarquia</i>	57
4.1.3	<i>Definindo a Estratégia Ativa de Consulta</i>	58
4.1.4	<i>Transdução dos Rótulos Representativos</i>	58
4.2	Trabalhos Relacionados	59
5	ESTRATÉGIAS DE MELHORIAS PARA O HS	63

5.1	Transdução <i>Bottom-Up</i>	63
5.2	Múltiplas Hierarquias	64
5.3	Representatividade com Hierarquias de Densidade e Simplificadas .	66
5.4	Complexidade Computacional	71
6	EXPERIMENTOS E DISCUSSÕES	73
6.1	Bases de Dados	73
6.2	Resultados com Transdução <i>Bottom-Up</i>	74
6.3	Resultados com Múltiplas Hierarquias	75
6.4	Resultados com Hierarquias de Densidade e Simplificadas	80
6.5	Comparação com outros Algoritmos Ativos	80
7	CONCLUSÃO	85
REFERÊNCIAS		87
APÊNDICE A	INTERVALO DE CONFIANÇA PARA PROPORÇÃO	93
APÊNDICE B	TABELA DE CLASSIFICAÇÃO INDUTIVA	97
APÊNDICE C	TEMPO MÉDIO DE EXECUÇÃO	99

INTRODUÇÃO

1.1 Contextualização

O Aprendizado de Máquina (AM), subárea da Inteligência Artificial, concentra algoritmos capazes de aprender a resolver automaticamente um problema a partir de experiências que o descreve. As características e premissas desses algoritmos possibilitam suas aplicações em problemas de diversas áreas de conhecimento, fazendo do AM um campo multidisciplinar (MITCHELL, 1997). Devido à forma como a experiência pode estar representada, os algoritmos do AM são encontrados dentro de duas abordagens principais de aprendizado: o aprendizado supervisionado e não supervisionado (LUXBURG; SCHÖLKOPF, 2008).

Dentro da abordagem supervisionada estão algoritmos capazes de aprender com a experiência que contém algum auxílio (supervisão) externo. Uma classe de tarefa comum nessa abordagem é a classificação, onde o auxílio externo está presente como rótulos de classes e o objetivo é aprender uma função para associar corretamente qualquer instância do problema à sua respectiva classe (LUXBURG; SCHÖLKOPF, 2008).

Quando os rótulos de classe não estão pré-definidos, uma alternativa é buscar, sem qualquer auxílio externo, estruturas de grupos que possam auxiliar no entendimento da natureza do problema (FACELI *et al.*, 2011). Esse é o objetivo dos algoritmos de agrupamento de dados que, pela ausência do auxílio externo, estão dentro da abordagem não-supervisionada (JAIN; DUBES, 1988).

Há certas situações em que o auxílio externo está parcialmente presente, seja fornecido como rótulos de classe ou como informações sobre a estrutura de grupos. Embora em tarefas de agrupamento de dados, o auxílio externo se configura como informações úteis para obter a estrutura de grupos, em tarefas de classificação tais auxílios podem não ser representativos para o problema, resultando no super-ajustamento ou sub-ajustamento das funções (LUXBURG; SCHÖLKOPF, 2008; FACELI *et al.*, 2011).

Situações em que se dispõe do auxílio externo parcial têm se tornando comuns (SETTLES, 2009; FU *et al.*, 2013; CHAPELLE *et al.*, 2010) e uma das causas é o crescente avanço tecnológico em diversos ramos de conhecimento, possibilitando uma rápida e numerosa aquisição de dados, enquanto que obter algum tipo de auxílio externo acaba sendo uma tarefa difícil.

Com o objetivo de propor soluções eficientes para os problemas de aprendizado com pouco auxílio externo e, portanto, minimizando possíveis custos/dificuldades no fornecimento dos mesmos, destacam-se os algoritmos das abordagens: semi-supervisionada e o semi-supervisionada ativa (CHAPELLE *et al.*, 2010; SETTLES, 2009).

Em tarefas de classificação, por exemplo, o objetivo dos algoritmos em ambas abordagens é permitir que uma função de perda possa ser obtida a partir de poucas experiências com rótulos de classe, minimizando assim dificuldades que podem surgir durante o processo de obtenção dos rótulos.

No aprendizado semi-supervisionado o conjunto mínimo de auxílio externo está fixo e pré-definido, no aprendizado ativo, por outro lado, tal conjunto é construído à medida que as consultas são realizadas a um “oráculo”. Na prática, o “oráculo” é um usuário, um especialista do problema ou o resultado de análises laboratoriais, por exemplo.

1.2 Motivação

Um importante componente que diferencia um algoritmo ativo de um algoritmo semi-supervisionado é a criação de uma estratégia de consulta a fim de estabelecer soluções eficientes com o mínimo possível de consultas ao oráculo. Mais do que isso, estratégias de consultas devem permitir que o conhecimento adquirido por consultas realizadas anteriormente sejam úteis para aprimorar as futuras (TONG; KOLLER, 2002).

A literatura destaca diferentes tipos de estratégias de consultas e, portanto, diferentes algoritmos ativos (LI *et al.*, 2014; HUANG *et al.*, 2010; DASGUPTA, 2009). De maneira geral, elas se encaixam em três categorias principais: estratégias baseadas em informatividade, baseadas em representatividade e baseadas em pré-agrupamento.

Na categoria de estratégias baseadas em pré-agrupamento, Dasgupta e Hsu (2008) propuseram um algoritmo com características que o coloca à frente de principais algoritmos ativos dentro das demais categorias. Tal algoritmo foi elaborado para tarefas de classificação e opera sobre alguma hierarquia de grupos, aprimorando decisões a partir de informações anteriormente concedidas pelo oráculo.

A principal premissa desse algoritmo é que, se a hierarquia de grupos fornecida como entrada, for capaz de estabelecer uma relação dos grupos com as possíveis classes da tarefa de classificação, então será possível, com poucas consultas ao oráculo, propor uma solução eficiente para o problema.

Embora o algoritmo tenha suas vantagens, há algumas dificuldades associadas. Uma das que se pode questionar é quando a hierarquia de grupos, particularmente escolhida e fornecida como entrada, está longe de satisfazer o tal pressuposto, enquanto outras hierarquias, de natureza distinta, poderiam satisfazer a premissa. A literatura em algoritmos de agrupamento de dados estabelece diferentes formas de se obter hierarquias de grupos ([TAN et al., 2005](#)), sendo assim, diferentes entradas para o algoritmo. Se a hierarquia de grupos escolhida estiver longe de satisfazer o pressuposto, então a qualidade do algoritmo estará comprometida, isto é, muitas consultas terão de ser realizadas ao oráculo até que uma solução eficiente para o problema de classificação possa ser garantida. Entretanto, determinar antecipadamente qual a melhor hierarquia para o algoritmo, em princípio, não é factível, primeiro porque as hierarquias são obtidas sem qualquer auxílio externo e, mesmo que parte do auxílio externo esteja pré-disponível usá-lo para guiar a escolha da hierarquia não é uma tarefa simples. Outras dificuldades do algoritmo se relacionam com sua estratégia de consulta e com a qualidade final da solução ao encerrar seu processo ativo. Superar tais dificuldades constitui o foco central deste trabalho.

1.3 Objetivos Gerais e Específicos

Dados os aspectos positivos e as dificuldades do algoritmo de [Dasgupta e Hsu \(2008\)](#), esta pesquisa busca definir estratégias para contornar tais dificuldades e permitir que a qualidade de resposta do algoritmo dentro de um cenário semi-supervisionado ativo para tarefas de classificação seja aprimorada.

Dentro do cenário onde se encontram os algoritmos de aprendizado semi-supervisionado ativo para classificação, espera-se que a capacidade de mapear as instâncias do problema aos seus respectivos rótulos de classe, isto é, a qualidade de resposta do algoritmo ativo para o problema, seja, com poucas consultas ao oráculo, comparável e/ou superior àquelas que seriam obtidas por algoritmos supervisionados dado um número maior de instâncias rotuladas ([SETTLES, 2009](#)). Em particular, este trabalho tem como objetivos:

- Definir estratégias para estabelecer soluções finais para tarefas de classificação que sejam comparáveis e/ou superiores às soluções do algoritmo de [Dasgupta e Hsu \(2008\)](#) na sua forma original e/ou diante de estratégias de melhoria já propostas na literatura;
- Investigar o algoritmo em diferentes tipos de hierarquias de grupos. Sua aplicação em hierarquias tradicionais como *Ward's* ([JAIN; DUBES, 1988](#)) e em hierarquias baseadas em densidade recentemente propostas ([CAMPELLO et al., 2015](#)), a fim de verificar e discutir as soluções do algoritmo com as estratégias de melhorias propostas por este trabalho sobre diferentes tipos de hierarquias;
- Permitir que as estratégias de melhorias deste trabalho sejam independentes do tipo de hierarquias de grupos fornecida como entrada para o algoritmo;

- Avaliar as soluções das estratégias definidas em dados que caracterizam problemas reais e, portanto, verificar e discutir as melhorias;
- Avaliar em dados que caracterizam problemas reais a competitividade em soluções para tarefas de classificação do algoritmo proposto quando comparado a outros algoritmos ativos na literatura.

1.4 Estrutura

Este trabalho está estruturado da seguinte forma:

- O Capítulo 2 apresenta uma revisão bibliográfica abrangendo as abordagens de aprendizado semi-supervisionado e ativo;
- O Capítulo 3 descreve os principais algoritmos hierárquicos que se configuram como elementos importantes no contexto deste trabalho.
- O Capítulo 4 descreve o algoritmo proposto por [Dasgupta e Hsu \(2008\)](#), que serve como ponto de partida para os desenvolvimentos nesta dissertação, bem como trabalhos relacionados da literatura;
- O Capítulo 5 descreve o algoritmo proposto neste trabalho;
- O Capítulo 6 apresenta os resultados obtidos, avaliando o algoritmo proposto junto a outros algoritmos ativos da literatura;
- O Capítulo 7 apresenta as conclusões desta pesquisa, as principais contribuições e possíveis investigações futuras.

APRENDIZADO DE MÁQUINA

Este capítulo apresenta um breve descrição sobre Aprendizado de Máquina e das principais abordagens de aprendizado: o aprendizado supervisionado e não supervisionado. Em seguida, o aprendizado semi-supervisionado, o aprendizado ativo e os seus principais algoritmos são destacados.

2.1 Aprendizado Indutivo e Suas Abordagens

O crescente avanço tecnológico dos últimos anos possibilita gerar e armazenar numerosas quantidades de dados (ALPAYDIN, 2004). No intuito de diminuir e/ou substituir a intervenção humana para lidar com esses dados e propor soluções eficientes para uma aplicação, tornou-se fundamental o desenvolvimento de ferramentas computacionais capazes de realizar tais tarefas de forma eficaz e automatizada (FACELI *et al.*, 2011). Essas ferramentas têm sido desenvolvidas e aprimoradas nas últimas décadas dentro de um dos ramos da Inteligência Artificial: o Aprendizado de Máquina.

Há diferentes formas de caracterizar o que vem a ser o Aprendizado de Máquina (AM). Segundo Mitchell (1997), o AM é definido como a capacidade que uma máquina, com poder de processamento computacional, possui de melhorar uma medida de desempenho **P** a partir de experiências **E**, que se referem a alguma classe de tarefas **T**.

Em caráter de ilustração, pode-se considerar um sistema de reconhecimento de dígitos, que reconhece números manuscritos representados por imagens (tarefa **T**), buscando maximizar o número de dígitos corretamente reconhecidos (desempenho **P**) a partir de um conjunto de imagens com as informações sobre os dígitos que cada imagem representa (experiência **E**). Alternativamente, um sistema de categorização automática pode aprender a identificar quais são os dígitos manuscritos representados nas imagens (tarefa **T**) visando estabelecer conjuntos de imagens de dígitos semelhantes (desempenho **P**) a partir de (dis)similaridades existentes nos

padrões das imagens (experiência $\mathbf{E}$).

Conforme [Mitchell \(1997\)](#) as características e premissas dos algoritmos do AM permitem abranger diferentes tipos de tarefas, fazendo de seus algoritmos ferramentas eficazes para a aplicação em problemas de diferentes áreas do conhecimento. Por exemplo, na agropecuária para a avaliação da qualidade de carnes ([CORTEZ et al., 2006](#)), na medicina para o diagnóstico de câncer ([FUREY et al., 2000](#)), na bioinformática para a exploração de informações em expressões gênicas ([YEUNG; RUZZO, 2001](#)), no processamento de linguagem natural para o reconhecimento de voz ([DENG; LI, 2013](#)), em finanças para a detecção de fraudes ([BOLTON; HAND, 2002](#)), entre outras áreas.

Em aplicações dos algoritmos do AM, a experiência $\mathbf{E}$ se refere a uma parte do problema a ser investigado e é comumente representada por um conjunto $\mathbf{X} \subset \mathcal{X}$, denominado conjunto de dados ([MITCHELL, 1997](#)). Cada elemento ou observação do problema, $\mathbf{x} \in \mathbf{X}$, é, por sua vez, comumente representado como um vetor de n componentes $\mathbf{x} = (x_1, x_2, \dots, x_n)$, em que cada componente x_i é uma variável ou atributo que assume algum valor para descrever o problema. Caso o problema em estudo seja, por exemplo, reconhecer os dígitos em imagens então, pode-se, dentre outras alternativas, tomar cada imagem disponível como uma observação e associar cada componente a um pixel em particular, que assume o valor numérico referente ao nível de cinza correspondente.

Uma vez que o conjunto de dados tenha sido estabelecido, o algoritmo do AM poderá utilizar este conjunto em um processo conhecido como Aprendizado Indutivo, por meio do qual serão construídas estratégias que possibilitem soluções genéricas para o problema ([FACELI et al., 2011](#)). No Aprendizado Indutivo há duas abordagens de aprendizado principais: o aprendizado supervisionado e o não supervisionado.

A abordagem supervisionada caracteriza algoritmos que necessitam de alguma supervisão externa que é pré-definida como um conjunto especial $\mathbf{Y} \subset \mathbf{E}$ para realizar o aprendizado. O objetivo é estabelecer uma função f que leve cada observação $\mathbf{x}_i \in \mathbf{X}$ ao seu respectivo $y_i \in \mathbf{Y}$, bem como encaminhar qualquer outra observação da mesma natureza do problema $\mathbf{x} \in \mathcal{X}$ para o valor de $y \in \mathcal{Y}$ correspondente ([LUXBURG; SCHÖLKOPF, 2008](#)).

Para os algoritmos dentro da abordagem não supervisionada a informação externa $\mathbf{Y}$ está ausente. Nessa abordagem, o aprendizado será realizado, único e exclusivamente, a partir de informações obtidas diretamente das observações $\mathbf{x}_i \in \mathbf{X}$. Objetiva-se, assim, identificar padrões ou tendências que possam auxiliar no entendimento da natureza do problema que se deseja investigar ([FACELI et al., 2011](#)).

Em cada uma das duas abordagens estão importantes classes de tarefas. No aprendizado supervisionado, por exemplo, uma classe de tarefa que se destaca quando $y \in \mathbf{Y}$ assume valores categóricos é a tarefa de classificação, que produz como saída um mapeamento $f : \mathcal{X} \rightarrow \mathcal{Y}$ (também denominado de classificador) para associar qualquer observação do problema em estudo

ao seu respectivo rótulo de classe. Na abordagem não supervisionada tem-se, por exemplo, a tarefa de agrupamento que busca estabelecer grupos para as observações segundo suas inter-relações ou (dis)similaridades (JAIN; DUBES, 1988; LUXBURG; SCHÖLKOPF, 2008). Dos exemplos anteriores, o sistema de reconhecimento de dígitos representados em imagens trata-se de uma tarefa de classificação (perspectiva supervisionada) e o sistema de categorização automática dos dígitos refere-se a uma tarefa de agrupamento (perspectiva não supervisionada).

Embora o aprendizado supervisionado e não supervisionado se destaquem como principais abordagens do aprendizado indutivo, outros trabalhos também apontam a importância e crescente contribuição de outras duas abordagens de aprendizado: o semi-supervisionado e o ativo (CHAPELLE *et al.*, 2010; SETTLES, 2009).

Algoritmos dentro do aprendizado semi-supervisionado e ativo buscam apresentar soluções dentro de um cenário em comum: situação em que as observações do problema de aprendizado são obtidas de maneira fácil e muitas vezes em grande quantidades, mas fornecer alguma forma de supervisão externa se torna uma tarefa onerosa.

Esses cenários são comuns e crescentes em diferentes ramos de aplicação. Tuia *et al.* (2012) discutem, por exemplo, a crescente obtenção de imagens da superfície terrestre devido ao avanço dos sensores em satélites e ressalta dificuldades como o tempo exigido para a análise e categorização dos tipos de superfícies nas imagens. McCallum e Nigam (1998) e Tong e Koller (2002) expõem a situação do desenvolvimento de aplicações para a classificação de páginas da web. Nesse cenário, diversas pessoas devem estar aptas para atribuir rótulos para as páginas de interesse, entretanto, devido ao crescente avanço da Internet e o tempo exigido para atribuir os rótulos, pode ser de interesse para a empresa que desenvolve a aplicação empregar o menor número de pessoas para realizar essa tarefa. Além de exigências no tempo e recursos humanos, outras aplicações podem exigir também recursos financeiros (SAHA *et al.*, 2015), por exemplo para uma análise laboratorial rigorosa (WARMUTH *et al.*, 2003).

Nesse sentido, o objetivo dos algoritmos em ambas abordagens, semi-supervisionada e ativa, é estabelecer soluções eficientes para o problema de aprendizado a partir de uma supervisão externa mínima, possibilitando, dessa forma, minimizar qualquer custo associado ao fornecê-las (CHAPELLE *et al.*, 2010; SETTLES, 2009).

No intuito de delimitar o contexto de desenvolvimento dessa pesquisa, nas próximas seções são discutidas as abordagens de aprendizado semi-supervisionado e ativo.

2.2 Aprendizado Semi-Supervisionado

O aprendizado semi-supervisionado engloba algoritmos que buscam usar para o aprendizado de um modelo a fração de observações que possuem alguma supervisão externa $\mathbf{X}_l = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ em conjunto com as demais observações cuja informação externa está

ausente $\mathbf{X}_u = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ (ZHU, 2005). Há algoritmos que resolvem tanto problemas de classificação como problemas de agrupamento de dados. Quando se deseja explorar alguma informação presente nos dados não rotulados, com o objetivo de aumentar a qualidade do classificador parcialmente obtido de $\mathbf{X}_l$ ou apenas estimar os rótulos de classe das observações em $\mathbf{X}_u$, temos a *classificação semi-supervisionada* (ZHU, 2005). Quando as informações parciais externas são usadas juntamente com as informações que se pode obter de $\mathbf{X}_u$ para melhorar uma estrutura de grupos dos dados temos o *agrupamento semi-supervisionado* (BASU *et al.*, 2002).

Algoritmos de classificação semi-supervisionada são especialmente úteis em situações onde o conjunto $\mathbf{X}_l$ não é representativo o suficiente para o aprendizado indutivo de um classificador. Problemas como *overfitting* e *underfitting* são comuns em classificadores se as observações rotuladas não são capazes representar adequadamente o problema de classificação (FACELI *et al.*, 2011). Sendo assim, tais algoritmos recorrem às observações não rotuladas na esperança de que elas possam fornecer informações que contribuam para a qualidade final do classificador (ZHU, 2005).

Algoritmos de classificação semi-supervisionada podem ser transdutivos ou indutivos (CHAPELLE *et al.*, 2010). Se forem capazes de estimar apenas os rótulos de classe das observações em $\mathbf{X}_u$ então tais algoritmos são considerado transdutivos. Se além de estimar os rótulos do conjunto de dados os algoritmos forem capazes de estimar os rótulos de qualquer observação $\mathbf{x} \notin \mathbf{X}_u$, então são considerados indutivos.

Exemplos de algoritmos transdutivos e semi-supervisionados são os algoritmos baseados em grafos (ZHU, 2005), como por exemplo o algoritmo de Zhu *et al.* (2003) e o *MinCut* de Blum e Chawla (2001). Nesses algoritmos, cada observação rotulada e não rotulada são representadas como vértices conectados por arestas com pesos proporcionais às (dis)similaridades. Se a aresta estabelecer uma relação de similaridade então a premissa é que as observações conectadas por arestas com pesos elevados tendem a compartilhar o mesmo rótulo de classe. Partindo-se de $\mathbf{X}_l$ o objetivo é definir uma função para rotular as demais observações em $\mathbf{X}_u$.

A principal dificuldade dos algoritmos transdutivos baseados em grafos é que eles são sensíveis à escolha de um tipo particular de grafo, bem como aos parâmetros necessários para a construção do mesmo (SOUSA *et al.*, 2013).

Dentre os algoritmos semi-supervisionados indutivos pode-se destacar algoritmos como o *self-training* e *co-training*. No *self-training* (ZHU, 2005) um classificador é obtido a partir das observações rotuladas e utilizado para inferir as classes das observações em $\mathbf{X}_u$. As observações $\mathbf{x}_i \in \mathbf{X}_u$ com maior confiança em termos de seus rótulos estimados são adicionadas à $\mathbf{X}_l$ e um novo classificador é obtido retomando o processo.

Para o *co-training* (BLUM; MITCHELL, 1998) as observações em $\mathbf{X}_u$ e $\mathbf{X}_l$ têm seus atributos particionados resultando em subconjuntos $\mathbf{X}_{u1}$ e $\mathbf{X}_{u2}$, $\mathbf{X}_{l1}$ e $\mathbf{X}_{l2}$, denominados *views*. Dois classificadores são obtidos, um a partir de $\mathbf{X}_{l1}$ e outro a partir de $\mathbf{X}_{l2}$. Ambos são usados

para inferir as classes das observações em $\mathbf{X}_{u1}$ e $\mathbf{X}_{u2}$ respectivamente. As observações em $\mathbf{X}_{u1}$ e/ou $\mathbf{X}_{u2}$, que tenham alto grau de certeza sobre o rótulo estabelecido, são acrescentadas em $\mathbf{X}_{l1}$ e $\mathbf{X}_{l2}$, retomando o processo.

No *self-training* se observações classificadas com um alto grau de certeza em relação a um determinado rótulo forem na verdade de uma classe diferente e incorporadas em $\mathbf{X}_l$, farão com que o “erro”, seja propagado para as iterações seguintes, podendo comprometer criticamente a qualidade do classificador resultante (ZHU; GOLDBERG, 2009). No *co-training*, as *views* obtidas devem satisfazer a premissa de que sejam condicionalmente independentes. Entretanto, em muitas situações práticas é difícil assegurar tal premissa (ZHU; GOLDBERG, 2009).

A aplicação dos algoritmos descritos, assumem que $\mathbf{X}_l$ seja previamente estabelecido. Entretanto, há cenários em que é possível contar com o especialista de domínio ou qualquer fonte capaz de fornecer a supervisão externa a fim de construir $\mathbf{X}_l$. Nesses cenários, torna-se fundamental a aplicação de algoritmos semi-supervisionados que vão além dos algoritmos convencionais mencionados anteriormente. É dentro desses cenários que se pode destacar os algoritmos de aprendizado ativo.

2.3 Aprendizado Ativo

Algoritmos do aprendizado ativo se diferenciam dos algoritmos semi-supervisionados pois assumem que a supervisão externa mínima não está pré-estabelecida, mas é construída com o auxílio de um oráculo. Segundo Settles (2009), enquanto no aprendizado semi-supervisionado procura-se estabelecer a supervisão externa partindo-se do que se supõe saber a respeito do problema, no aprendizado ativo a supervisão externa é estabelecida buscando explorar os aspectos desconhecidos e promissores para o problema. Com o mesmo propósito de algoritmos de aprendizado semi-supervisionado, algoritmos ativos buscam minimizar as dificuldades que surgem durante o ato de prover as supervisões externas. Para isso, estabelecem como resultado um conjunto mínimo e promissor $\mathbf{X}_l^*$ (SETTLES, 2009).

Para estabelecer $\mathbf{X}_l^*$ representativo o bastante para a tarefa de aprendizado, os algoritmos do aprendizado ativo têm a liberdade de escolher observações $\mathbf{x} \in \mathbf{X}_u$ que julgam serem apropriadas para lidar com o problema e consultam ao oráculo para que o mesmo forneça a supervisão externa correspondente.

Nessa busca pelas observações, surgem diferentes tipos de algoritmos ativos caracterizados pelas diferentes formas de como escolher $\mathbf{x} \in \mathbf{X}_u$, essas formas de escolha são conhecidas como *estratégias de consulta* (SETTLES, 2009).

Tais estratégias desempenham um papel fundamental em um algoritmo ativo. Além de possibilitarem o estabelecimento de $\mathbf{X}_l^*$, devem ser elaboradas a fim de incorporar as informações de consultas realizadas para aprimorar as futuras (TONG; KOLLER, 2002).

Apesar das diferentes estratégias de consultas encontradas na literatura, segundo [Li et al. \(2014\)](#), [Huang et al. \(2010\)](#) e [Dasgupta \(2009\)](#) tais estratégias podem ser categorizadas dentro de três categorias principais: uma relacionada à busca de observações explorando suas incertezas ou potencial de influência a solução para o problema (Amostragem Informativa), outra relacionada à exploração de observações que demonstram ser representativas para as características do problema (Amostragem Representativa), e finalmente aquelas baseadas em pré-processamento dos dados por algoritmos de agrupamento de dados (Amostragem Baseada em Pré-Agrupamento). Suas características e seus principais algoritmos são destacados a seguir.

2.3.1 Amostragem Informativa

Nessa categoria, as observações a serem consultadas, em geral, são aquelas consideradas difíceis de terem sua supervisão externa estabelecida por algum algoritmo de aprendizado ou aquelas que, uma vez consultadas ao especialista, contribuem para mudanças na solução pós aprendizado indutivo. Tais estratégias são as mais populares no aprendizado ativo ([SETTLES, 2009](#); [HUANG et al., 2010](#)) e, dentre as existentes, pode-se destacar as seguintes no contexto de tarefas de classificação: a *amostragem por incerteza*, *amostragem por comitê* e as baseadas em *mudanças no modelo* ([LEWIS; GALE, 1994](#); [SEUNG et al., 1992](#); [ROY; MCCALLUM, 2001](#)).

Proposta por [Lewis e Gale \(1994\)](#), a amostragem por incerteza (*uncertainty sampling*) tem como propósito selecionar uma observação $\mathbf{x}^* \in \mathbf{X}_u$ que possua baixa confiança em termos dos possíveis rótulos de classe.

Uma forma usual de estabelecer valores para quantificar a confiança sobre os rótulos de classe é analisar a resposta de um classificador f sobre as observações não rotuladas, aquelas que são duvidosamente classificadas por f serão fortes candidatas a serem consultadas. No trabalho de [Lewis e Gale \(1994\)](#) assume-se um problema de classificação com duas classes e usa-se o classificador probabilístico *naive* Bayes para selecionar as observações com probabilidade de classe próxima de 0.5.

Para as tarefas de classificação com mais de duas classes e usando-se classificadores probabilísticos a entropia de Shannon é uma forma clássica para medir tal incerteza ([HOLUB et al., 2008](#)). As observações mais incertas são aquelas com um elevado valor de entropia, conforme descreve a Equação (2.1):

$$\operatorname{argmax}_{\mathbf{x}^*} = \sum_i P_f(y_i|\mathbf{x}) \log P_f(y_i|\mathbf{x}) \quad (2.1)$$

,

em que $P_f(y_i|\mathbf{x})$ é a probabilidade estimada pelo classificador f que a observação $\mathbf{x}$ pertence à i -ésima classe y_i .

Na amostragem por comitê, por sua vez, mais de um classificador é considerado para determinar quais observações serão selecionadas ([SEUNG et al., 1992](#)). Nesse tipo de estratégia

de consulta, a incerteza é determinada pela discordância dos rótulos estabelecidos por cada um dos classificadores para uma determinada observação e objetiva-se apresentar ao oráculo aquelas com rótulos discordantes.

Outras pesquisas, que sucederam a partir do trabalho de [Tong e Koller \(2002\)](#), buscam usar *Support Vector Machines* (SVM) para determinar a margem de separação das possíveis classes do problema e estabelecer como observações incertas aquelas próximas à margem.

Uma limitação a que essas estratégias ativas estão sujeitas, devido à forma como as observações são escolhidas e como o processo é guiado para as futuras rotulações, é o problema conhecido como *viés de amostragem* ([DASGUPTA; HSU, 2008](#); [HUANG et al., 2010](#)). A ilustração dessa dificuldade é apresentada pela Figura 1.

Na Figura 1, as observações são provenientes de duas possíveis classes (Classe 1 e Classe 2). As estratégias descritas assumem que uma fração de observações tenham seus rótulos predefinidos para que seja possível obter algum classificador inicial.

Figura 1 – Viés de Amostragem.

Fonte: Adaptada de [Dasgupta e Hsu \(2008\)](#).

Pela distribuição das observações no espaço dos atributos, percebe-se que um conjunto de dados inicialmente rotulado tem maior probabilidade de conter observações das regiões A e B

(Figura 1a). Dessa forma, um possível classificador inicial pode, por exemplo, ser representado por f_0 (Figura 1b).

Uma estratégia baseada em incerteza que faz o uso de f_0 para selecionar as observações não rotuladas guiará a busca em regiões próximas à fronteira de decisão (região de alta incerteza) podendo resultar no classificador f_n (Figura 1c). Considerando que todas as observações acima da fronteira de decisão são rotuladas como Classe 2 e as abaixo são rotuladas como Classe 1, se o classificador f_n fosse usado para realizar a transdução, então 10% das observações seriam incorretamente classificadas. Note, entretanto, que existe um classificador f^* (Figura 1d) capaz de classificar corretamente todas as observações, mas que dificilmente será encontrado em razão do viés de amostragem.

Outras formas de consulta da abordagem informativa buscam por objetos que podem trazer grandes mudanças para o classificador se o seu rótulo de classe fosse conhecido (SETTLES *et al.*, 2008). Em Settles *et al.* (2008) a mudança é medida pela diferença que se observa entre os parâmetros do classificador obtido a partir de $\mathbf{X}_l^*$ em relação aos parâmetros do classificador obtido com $\mathbf{X}_l^* + (\mathbf{x}^*, y_i)$. Se a observação $\mathbf{x}^*$ é aquela que provoca a maior diferença tomando y_i o seu rótulo, então $\mathbf{x}^*$ é candidata a consulta. Essas estratégias podem até contornar o viés de amostragem, mas o custo computacional pode ser elevado se houverem inúmeras observações não rotuladas, isso porque o classificador precisa ser retreinado para cada tupla $(\mathbf{x}^*, y_i)$ considerando todas as observações não rotuladas para todos os possíveis rótulos do problema (FU *et al.*, 2013).

Outra dificuldade das estratégias dentro da abordagem informativa de consulta é a sensibilidade a ruídos nos dados. Em certas situações ruídos são aquelas observações que não pertencem à distribuição que gerou as observações e podem levar ao *overfitting* de um classificador (FACELI *et al.*, 2011). Conforme discutido em Settles (2009) e Fu *et al.* (2013) as estratégias de consultas dentro dessa abordagem deveriam considerar aspectos de como os dados estão distribuídos no espaço de atributos ao invés de tomar decisões de escolhas considerando individualmente as observações. Assim, seria possível evitar que ruídos fossem candidatos a consultas.

A Figura 2 apresenta uma ocasião onde uma consulta pelo rótulo da observação $\mathbf{x}^*$ seria mais conveniente do que consultar pelo rótulo da observação $\mathbf{r}$, que se trata possivelmente de uma observação ruidosa.

Estratégias de consultas que levam em consideração informações sobre a distribuição dos dados no espaço dos atributos para selecionar as observações são encontradas dentro da abordagem de amostragem representativa, que é discutida a seguir.

2.3.2 Amostragem Representativa

Estratégias de busca representativas procuram fazer com que uma “cobertura” a respeito da distribuição dos dados no espaço de atributos possa ser considerada na escolha de uma

Figura 2 – Classificador f_n obtido a partir das observações rotuladas (representadas pelos círculos preenchidos).

Fonte: Adaptada de [Settles et al. \(2008\)](#).

observação a ser consultada ([LI et al., 2014](#)).

Em [Zhang et al. \(2011\)](#) assume-se que as observações podem ser linearmente reconstruídas a partir de observações próximas, as observações consideradas representativas são aquelas que minimizam o erro de se reconstruir todo o conjunto de dados. Em outros trabalhos, como [Basu et al. \(2004\)](#) e [Mallapragada et al. \(2008\)](#), a representatividade está na busca por observações que são distantes uma das outras, possibilitando abranger todo o conjunto de dados. Em [Yu et al. \(2006\)](#) observações representativas são aquelas em que a informação concedida pelo oráculo possa também representar outras observações similares cuja supervisão externa está ausente.

Uma limitação de consulta por representatividade é que a seleção de novas consultas não faz uso do conhecimento adquirido a partir de consultas já realizadas mas apenas o que já se supõe saber a partir da distribuição dos dados. Neste sentido, uma possibilidade das estratégias de consultas que é aprimorar futuras consultas a partir de conhecimento adquirido por consultas anteriores, não é potencialmente explorada.

Outra dificuldade associada a esses métodos é que regiões de difíceis decisões, como a fronteira de decisão entre classes, podem não ser exploradas a fim de melhorar, por exemplo, a capacidade de generalização dos classificadores. Nesse sentido, alguns trabalhos buscam ponderar a representatividade e a informatividade das observações como critérios de escolha. Em [Settles et al. \(2008\)](#) e [Huang et al. \(2010\)](#) define-se estratégias que buscam selecionar observações que tenham alta indecisão a respeito de algum rótulo de classe, mas que estão em regiões onde há numerosas observações similares ainda não rotuladas.

Outra proposta para o desenvolvimento de estratégias de consulta são aquelas que usam informações obtidas a partir de algoritmos de agrupamento de dados para guiar a escolha das observações, possivelmente também com o auxílio de critérios baseados em informatividade

e representatividade. Estratégias de consultas sobre o contexto de classificação que usam essa abordagem são descritas a seguir.

2.3.3 Amostragem Baseada em Pré-Agrupamento

Uma finalidade do uso de algoritmos de agrupamento de dados é buscar por estruturas de grupos que possam auxiliar o entendimento de como os dados estão distribuídos. Dependendo do algoritmo de agrupamento utilizado, informações de interesse podem ser obtidas e usadas para a criação de estratégias de consultas.

Em [Xu et al. \(2003\)](#) um balanceamento entre informatividade e representatividade é usado para selecionar as observações. No trabalho, um classificador *support vector machines* (SVM) é obtido para determinar a fronteira de decisão. Em seguida, as observações próximas da fronteira são selecionadas e algoritmos de agrupamento como o *k-means* e *k-medoid* são utilizados para encontrar os centroides sobre tais observações. Aquelas próximas aos k centroides encontrados são candidatas a consultas e posteriormente usadas para retomar o processo.

Outras estratégias são construídas considerando inicialmente as respostas de algoritmos de agrupamento de dados. [Kang et al. \(2004\)](#), [Zhu et al. \(2008\)](#) e [Hu et al. \(2010\)](#) propõem estratégias baseadas em pré-agrupamento para construir um conjunto inicial de dados rotulados, que dará início às estratégias que se encaixam nas abordagens informativas. Nessas pesquisas, algoritmos como o *k-means* e variantes são usados também com o mesmo propósito: considerar observações próximas aos centroides como aquelas a serem inicialmente rotuladas.

[Kang et al. \(2004\)](#), além de avaliar observações inicialmente rotuladas a partir de algoritmos como o *k-means*, também avaliaram a escolha de observações a partir de algoritmos de agrupamento hierárquico. No trabalho os autores concluem que o conjunto inicial de dados $\mathbf{X}_I$ gerado por *k-means* e suas variantes resultaram em performances instáveis, ao contrário de $\mathbf{X}_I$ provenientes de algoritmos hierárquicos, que foram recomendados.

[Nguyen e Smeulders \(2004\)](#) também estabelecem uma estratégia ativa baseada em agrupamento utilizando o *k-means*. Entretanto, a estratégia não é estabelecida simplesmente como um pré-processamento para as estratégias informativas. Todo o processo é elaborado a partir da resposta do algoritmo de agrupamento, que tem suas soluções refinadas à medida que novas rotulações são realizadas.

Outra estratégia que busca aprimorar as futuras consultas a partir daquelas rotuladas usando respostas de algoritmos de agrupamento de dados é proposta por [Dasgupta e Hsu \(2008\)](#). No trabalho, a estratégia ativa opera sobre uma hierarquia de grupos buscando selecionar grupos representativos e incertos a respeito dos possíveis rótulos de classe.

A principal dificuldade das estratégias dentro dessa categoria de consulta é de que suas respostas são dependentes da qualidade do agrupamento ([HUANG et al., 2010](#); [HU et al., 2010](#)). Algoritmos como o *k-means* e variantes são conhecidos por serem algoritmos não

determinísticos produzindo diferentes soluções a partir de diferentes inicializações e número de grupos. Outros como o DBSCAN (ESTER *et al.*, 1996) capaz de modelar conceitos de ruídos são criticamente sensíveis à escolhas dos parâmetros de entrada. Há diversos algoritmos capazes de produzir hierarquias de grupos (TAN *et al.*, 2005), mas a escolha de uma hierarquia apropriada que possa adequar ao problema de classificação nem sempre é uma escolha trivial, em especial por ser não supervisionada.

2.4 Conclusão

Este Capítulo apresentou uma introdução ao Aprendizado de Máquina e características de suas duas principais abordagens de aprendizado: o aprendizado supervisionado que opera usando supervisões externas definidas antecipadamente e o aprendizado não supervisionado que opera sem qualquer auxílio externo.

Também foram apresentadas outras duas abordagens de aprendizado que vem se destacando em diversas aplicações de interesse: o aprendizado semi-supervisionado e o aprendizado ativo. Ambas se destacam dentro do cenário onde os dados são fáceis e abundantes de se obterem, mas prover algum tipo de auxílio externo como rótulos de classe, por exemplo, se configura como uma etapa difícil.

Aprendizado ativo difere do aprendizado semi-supervisionado por suas estratégias de consultas que permitem, a partir de informações parciais, procurar por observações relevantes para o problema e apresentar ao oráculo para que o mesmo forneça alguma supervisão.

As estratégias de consultas se configuram como importantes componentes para o desenvolvimento de um algoritmo de aprendizado ativo. Na literatura, há diversas dessas estratégias que podem ser caracterizadas dentro de três categorias distintas segundo suas características gerais. Tais categorias foram brevemente discutidas neste capítulo, contrastando a vantagem e desvantagem de cada uma delas.

HIERARQUIAS DE GRUPOS

Este capítulo apresenta os principais algoritmos hierárquicos. Tais algoritmos se encontram dentro da abordagem não supervisionada de aprendizado e produzem como resultados partições de grupos organizadas hierarquicamente. Para o contexto deste trabalho, as hierarquias de grupos se configuram como elementos importantes para o algoritmo de aprendizado ativo que este trabalho investigou.

3.1 Algoritmos Hierárquicos

Algoritmos hierárquicos estão entre as técnicas que se destacam em tarefas de agrupamento de dados (TAN *et al.*, 2005). São algoritmos capazes de gerar partições organizadas hierarquicamente conforme as noções de *partição* e *partições aninhadas* Jain e Dubes (1988).

Definição 1. *Partição Rígida:* Se as observações do conjunto de dados $\mathbf{X}$ são distribuídas em $\mathbf{G}_i$ subconjuntos (ou grupos) para $1 \leq i \leq k$, então o conjunto $\mathbf{P} = \{\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_k\}$ é uma partição de $\mathbf{X}$ se todas as propriedades a seguir são satisfeitas:

- (i) $\mathbf{G}_i \neq \emptyset, \forall i \in \{1, \dots, k\}$;
- (ii) $\mathbf{G}_i \cap \mathbf{G}_j = \emptyset, \forall i, j \in \{1, \dots, k\}$ e $i \neq j$;
- (iii) $\bigcup_{i \in \{1, \dots, k\}} \mathbf{G}_i = \mathbf{X}$.

Definição 2. *Partições Rígidas aninhadas:* Uma partição $\mathbf{P}_i$ com m grupos é aninhada a uma partição $\mathbf{P}_j$ com n grupos se para $i \neq j$ e $\forall \mathbf{G}_v \in \mathbf{P}_i, \exists w \in \{1, \dots, n\} \mid \mathbf{G}_v \subset \mathbf{G}_w, \mathbf{G}_w \in \mathbf{P}_j$.

A Figura 3 ilustra o exemplo de uma hierarquia de grupos. Observa-se que a partição $\mathbf{P}_1$ é aninhada a $\mathbf{P}_2$, mas que o contrário não é verificado pois $\mathbf{G}_3 \in \mathbf{P}_2$ não é subconjunto de nenhum grupo $\mathbf{G}_i \in \mathbf{P}_1$.

Para a construção de uma hierarquia de grupos é fundamental estabelecer alguma relação de proximidade entre os pares de observações do conjunto de dados. Essa relação de proximidade

Figura 3 – Exemplo de uma Hierarquia de Grupos sobre 5 observações

indica algum tipo de similaridade ou dissimilaridade entre as observações e pode ser representada por uma matriz, em que o elemento da linha i e coluna j representa a proximidade da observação i com a observação j .

$$\begin{pmatrix} d(\mathbf{x}_1, \mathbf{x}_1) & d(\mathbf{x}_1, \mathbf{x}_2) & \dots & d(\mathbf{x}_1, \mathbf{x}_j) \\ \vdots & \vdots & \dots & \vdots \\ d(\mathbf{x}_i, \mathbf{x}_1) & d(\mathbf{x}_i, \mathbf{x}_2) & \dots & d(\mathbf{x}_i, \mathbf{x}_j) \end{pmatrix}_{i \times j}$$

Para pré-estabelecer a matriz de proximidade faz-se o uso de alguma métrica $d(.,.)$ garantindo propriedades tais como simetria. Dentre as principais métricas usadas pode-se destacar a distância Euclidiana, distância de Manhattan, distância de Minkowski e distância de Mahalanobis (GAN *et al.*, 2007).

Uma vez estabelecida a medida de proximidade entre os pares de observações, os algoritmos hierárquicos seguem uma abordagem divisiva ou aglomerativa para construir a hierarquia de grupos. Aqueles que se enquadram na abordagem aglomerativa são elaborados para construir uma hierarquia "de baixo para cima", seguindo as descrições gerais do Algoritmo 1 abaixo.

No momento em que os grupos são unidos é comum que os algoritmos hierárquicos reportem alguma informação numérica sobre quando os grupos foram formados. A interpretação deste valor numérico dependerá de como está definida política de união de grupos. Quando a hierarquia e a informação numérica são reportadas tem-se o que se denomina de *dendrograma*, tal como ilustra a Figura 4.

Em certas situações, as informações de escala são úteis para que se possa examinar e definir cortes horizontais a fim de obter uma partição com grupos de interesse, tal como ilustrado nas Figuras 5a e 5b. Em outros casos as informações do dendrograma podem sugerir que algumas

Algoritmo 1: Algoritmo Hierárquico Aglomerativo**Entrada:** Matriz de proximidade (similaridade ou dissimilaridade) entre observações.**1 início:**2 Para cada observação gere um grupo constituído somente daquela observação (*singleton*).**3 repita**

4 | Una o par de grupo mais similar (menos dissimilar) de acordo com uma medida de (dis)similaridade entre grupos pré-estabelecida.

5 **até** um único grupo seja obtido;**6 fim.**

Figura 4 – Exemplo de Dendrograma. Sob a abordagem aglomerativa, a escala indica a união dos grupos.

observações são diferentes das demais sendo útil, por exemplo, para a detecção de *outliers*, como ilustrado nas Figuras 5c e 5d.

O que caracteriza cada um dos algoritmos é a regra usada para determinar quais grupos serão escolhidos e unidos (Linha 4 do Algoritmo 1). Tem-se assim, regras baseadas em minimização de variância, em máxima ligação, baseadas em densidade, entre outras. Dentre os algoritmos hierárquicos aglomerativos existentes pode-se destacar algoritmos como o *Ward's*, *Average-Linkage*, *Complete-Linkage*, *Single-Linkage* e *HDBSCAN** (JAIN; DUBES, 1988; TAN *et al.*, 2005; CAMPELLO *et al.*, 2015).

A fim de compreender os possíveis tipos de hierarquias que podem ser entregues como entrada para algoritmos de classificação ativa como o algoritmo de Dasgupta e Hsu (2008), que constituiu foco de particular interesse do presente trabalho, serão caracterizados nos próximos tópicos os algoritmos hierárquicos destacados acima e as características que possibilitam identificar cenários para suas aplicações.

Figura 5 – Exemplos de aplicações em dendrogramas

3.1.1 Ward's

O *Ward's* (WARD; JOE, 1963) também denominado de método da mínima variância busca encontrar partições de grupos em que a variação dentro dos grupos seja mínima. Para Jain e Dubes (1988), minimizar a variação dentro dos grupos pode ser interpretado como o problema dual em buscar maximizar a variação entre os grupos.

A variação e_i dentro de um grupo G_i é calculada como a soma das distâncias Euclidianas das observações $\mathbf{x}_j \in G_i$ ao centroide $\bar{\mathbf{x}}_i$ do grupo, conforme a Equação (3.1). A interpretação do que se denomina de variação dentro do grupo vem do fato de que o centroide corresponde ao valor esperado das observações do grupo. Assim, medir o quão distantes as observações estão do seu valor esperado é uma forma de medir a variância.

$$e_i = \sum_{\mathbf{x}_j \in G_i} d(\mathbf{x}_j, \bar{\mathbf{x}}_i)^2, \bar{\mathbf{x}}_i = \frac{1}{|G_i|} \sum_{\mathbf{x}_j \in G_i} \mathbf{x}_j, \quad (3.1)$$

Na equação de variância intra-grupo acima, $|G_i|$ é o número de observações do grupo G_i e $d(.,.)$ é a distância Euclidiana.

Considerando uma partição $\mathbf{P}$ de grupos, a variação da partição $E_{\mathbf{P}}$ pode ser calculada

como a soma da variação dentro do grupo de cada grupo na partição, isto é:

$$E_{\mathbf{P}} = \sum_{\mathbf{G}_i \in \mathbf{P}} e_i \quad (3.2)$$

Para determinar quais grupos serão unidos deseja-se saber a mudança que houve na variação da partição após a união de dois grupos quaisquer. Para isso, seja $\mathbf{G}_c$ o grupo formado pela união dos grupos $\mathbf{G}_a$ e $\mathbf{G}_b$. Pode-se escrever a diferença entre $E_{\mathbf{P}}$, da partição que contém $\mathbf{G}_a$ e $\mathbf{G}_b$, e $E_{\mathbf{P}'}$, da partição que contém $\mathbf{G}_c$, da seguinte forma:

$$\begin{aligned} E_{\mathbf{P}} - E_{\mathbf{P}'} &= \left(\sum_{\mathbf{G}_i \in \mathbf{P} - \{\mathbf{G}_a, \mathbf{G}_b\}} e_i + e_a + e_b \right) - \left(\sum_{\mathbf{G}_i \in \mathbf{P}' - \{\mathbf{G}_c\}} e_i + e_c \right) \\ E_{\mathbf{P}} - E_{\mathbf{P}'} &= e_a + e_b - e_c \\ -(E_{\mathbf{P}} - E_{\mathbf{P}'}) &= e_c - e_a - e_b \end{aligned} \quad (3.3)$$

Como a variação na partição aumenta a medida que novas partições são obtidas, espera-se que $(E_{\mathbf{P}} - E_{\mathbf{P}'})$ tenha um valor negativo. Quanto mais negativo o valor então maior será a variação dentro do grupo. Como o objetivo do algoritmo é obter partições de grupos onde a variação dentro dos grupos sejam mínimas, escolhe-se portanto $\mathbf{G}_i \in \mathbf{P}$ e $\mathbf{G}_j \in \mathbf{P}$ que fornecem o maior valor para $(E_{\mathbf{P}} - E_{\mathbf{P}'})$ ou o menor valor para $-(E_{\mathbf{P}} - E_{\mathbf{P}'})$.

Para ilustrar essa ideia, considere o conjunto de dados $\mathbf{X}$ descrito pela Tabela 1, onde cada observação é constituída por dois atributos numéricos. A matriz de proximidade D é formada por elementos d_{ij} que correspondem à distância Euclidiana entre os pares de observações de $\mathbf{X}$.

Tabela 1 – Conjunto de Dados $\mathbf{X}$

Observações	Atributos	
	x_1	x_2
$\mathbf{x}_1$	2.00	2.50
$\mathbf{x}_2$	1.00	2.00
$\mathbf{x}_3$	0.80	2.30
$\mathbf{x}_4$	4.00	3.50

$$D = \begin{pmatrix} 0.00 & & & \\ 1.11 & 0.00 & & \\ 1.21 & 0.36 & 0.00 & \\ 2.23 & 3.35 & 3.41 & 0.00 \end{pmatrix}$$

Conforme a descrição geral do Algoritmo 1 cada observação $\mathbf{x}_i \in \mathbf{X}$ será inicialmente um grupo, logo a partição inicial resultante é dada por $\mathbf{P}_0 = \{\mathbf{G}_1 = \{\mathbf{x}_1\}, \mathbf{G}_2 = \{\mathbf{x}_2\}, \mathbf{G}_3 = \{\mathbf{x}_3\}, \mathbf{G}_4 = \{\mathbf{x}_4\}\}$. Para verificar os dois grupos que serão unidos inicialmente deve-se calcular $-(E_{\mathbf{P}_0} - E_{\mathbf{P}'})$ tomando $E_{\mathbf{P}'}$ a partição formada pelas diferentes alternativas de unir os grupos em $\mathbf{P}_0$.

Considerando, por exemplo, $\mathbf{P}' = \mathbf{P}_1^{\mathbf{G}_2, \mathbf{G}_3}$ a partição resultante da união de $\mathbf{G}_2$ e $\mathbf{G}_3$, como exemplo, pela Equação (3.1) tem-se os respectivos centroides $\bar{\mathbf{x}}_{\mathbf{G}_2} = (1, 2)$ e $\bar{\mathbf{x}}_{\mathbf{G}_3} = (0.8, 2.3)$. Dado o centroide do grupo $\mathbf{G}_2 \cup \mathbf{G}_3 = (0.90, 2.15)$, pelas Equações (3.2) e (3.3) tem-se:

$$\begin{aligned}
-(E_{\mathbf{P}_0} - E_{\mathbf{P}'}) &= e_{\{\mathbf{x}_2\} \cup \{\mathbf{x}_3\}} - e_{\{\mathbf{x}_2\}} - e_{\{\mathbf{x}_3\}} \\
-(E_{\mathbf{P}_0} - E_{\mathbf{P}'}) &= (d(\mathbf{x}_2, \bar{\mathbf{x}}_{\{\mathbf{x}_2, \mathbf{x}_3\}})^2 + d(\mathbf{x}_3, \bar{\mathbf{x}}_{\{\mathbf{x}_2, \mathbf{x}_3\}})^2) - d(\mathbf{x}_2, \bar{\mathbf{x}}_{\{\mathbf{x}_2\}})^2 - d(\mathbf{x}_3, \bar{\mathbf{x}}_{\{\mathbf{x}_3\}})^2 \\
-(E_{\mathbf{P}_0} - E_{\mathbf{P}'}) &= (d(\mathbf{x}_2, \bar{\mathbf{x}}_{\{\mathbf{x}_2, \mathbf{x}_3\}})^2 + d(\mathbf{x}_3, \bar{\mathbf{x}}_{\{\mathbf{x}_2, \mathbf{x}_3\}})^2) - 0 - 0 \\
-(E_{\mathbf{P}_0} - E_{\mathbf{P}'}) &= 0.36
\end{aligned}$$

Cada elemento da matriz de proximidade fornecida como entrada implicitamente define os valores de $-(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{ij}})$ sendo $E_{\mathbf{P}_1^{ij}}$ a partição formada pela união do grupo de índice i com o grupo de índice j pertencentes à partição $\mathbf{P}_0$. Inicialmente:

$$D = \begin{pmatrix} -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{11}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{12}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{13}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{14}}) \\ -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{21}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{22}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{23}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{24}}) \\ -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{31}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{32}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{33}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{34}}) \\ -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{41}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{42}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{43}}) & -(E_{\mathbf{P}_0} - E_{\mathbf{P}_1^{44}}) \end{pmatrix}$$

Com base nos valores da matriz D , os grupos $\mathbf{G}_2$ e $\mathbf{G}_3$ da partição $\mathbf{P}_0$ serão os primeiros a se unirem.

Considerando que $\mathbf{P}_1 = \{\mathbf{G}_1 = \{\mathbf{x}_1\}, \mathbf{G}_2 = \{\mathbf{x}_2, \mathbf{x}_3\}, \mathbf{G}_3 = \{\mathbf{x}_4\}\}$, pode-se ajustar D a fim de manter os cálculos já realizados e verificar que grupos serão unidos na próxima iteração. A partir da Equação (3.3) tem-se que:

$$E_{\mathbf{P}_1} - E_{\mathbf{P}_2^{12}} = 0.76$$

$$E_{\mathbf{P}_1} - E_{\mathbf{P}_2^{23}} = 2.25$$

$$\begin{pmatrix} -(E_{\mathbf{P}_1} - E_{\mathbf{P}_2^{11}}) & -(E_{\mathbf{P}_1} - E_{\mathbf{P}_2^{12}}) & -(E_{\mathbf{P}_1} - E_{\mathbf{P}_2^{13}}) \\ -(E_{\mathbf{P}_1} - E_{\mathbf{P}_2^{21}}) & -(E_{\mathbf{P}_1} - E_{\mathbf{P}_2^{22}}) & -(E_{\mathbf{P}_1} - E_{\mathbf{P}_2^{23}}) \\ -(E_{\mathbf{P}_1} - E_{\mathbf{P}_2^{31}}) & -(E_{\mathbf{P}_1} - E_{\mathbf{P}_2^{32}}) & -(E_{\mathbf{P}_1} - E_{\mathbf{P}_2^{33}}) \end{pmatrix} \Rightarrow \begin{pmatrix} 0.00 & & \\ 0.76 & 0.00 & \\ 2.23 & 2.25 & 0.00 \end{pmatrix}$$

Na próxima etapa $\mathbf{G}_2$ e $\mathbf{G}_1$ da partição $\mathbf{P}_1$ serão unidos. Ao final obtém-se o dendrograma:

As características do *Ward's* fazem com que ele favoreça grupos com formatos globulares no espaço de atributos das observações. A Figura 6 representa a distribuição de observações descritas por dois atributos numéricos, na subfigura 6b está o respectivo dendrograma *Ward's* e a linha em preto indica um corte horizontal capaz de reportar uma partição com 4 grupos distintos

pelas cores. Na subfigura 6c tem-se a representação do corte horizontal em 4 grupos. Percebe-se que a solução apresentada pela partição se alinha aos grupos de interesse indicados na subfigura 6a.

Figura 6 – Hierarquia Ward's para grupos com formatos esféricos

Ao contrário do cenário anterior, o Ward's tem dificuldade de estabelecer grupos com formas arbitrárias. Na subfigura 7b, três grupos são reportados pela partição, conforme indica o corte horizontal em preto. Entretanto, o corte definido não representa os três grupos de interesse ilustrados na subfigura 7a. Na subfigura 7c, tem-se a representação do corte estabelecido.

Figura 7 – Hierarquia Ward's para grupos com formatos arbitrários

3.1.2 Average-Linkage

O *Average-Linkage*, também denominado de *Unweighted Pair Group Method using Arithmetic averages - (UPGMA)*, avalia o valor médio da dissimilaridade entre os pares de observações em diferentes grupos a fim de determinar quais grupos serão escolhidos e unidos. Seja $\mathbf{P}$ a partição e $\mathbf{G}_i$ e $\mathbf{G}_j$ dois grupos em $\mathbf{P}$, a proximidade (dissimilaridade ou similaridade) $P(\mathbf{G}_i, \mathbf{G}_j)$ entre eles é definida pelo valor médio da métrica $d(.,.)$ entre os pares de observações dos dois grupos, conforme a Equação (3.4).

$$P(\mathbf{G}_i, \mathbf{G}_j) = \frac{1}{|\mathbf{G}_i||\mathbf{G}_j|} \sum_{\mathbf{x}_k \in \mathbf{G}_i} \sum_{\mathbf{x}_l \in \mathbf{G}_j} d(\mathbf{x}_k, \mathbf{x}_l) \quad (3.4)$$

Caso a métrica estabeleça uma relação de dissimilaridade entre as observações e se a dissimilaridade de $\mathbf{G}_i$ e $\mathbf{G}_j$ é a menor observada dentre todas as outras calculadas para os demais pares de grupos em $\mathbf{P}$, então $\mathbf{G}_i$ e $\mathbf{G}_j$ são unidos. Uma vez que $\mathbf{G}_i$ e $\mathbf{G}_j$ são unidos, pela Equação (3.4) a dissimilaridade entre o grupo resultante dessa união para qualquer outro grupo $\mathbf{G}_t \in \mathbf{P}$ pode ser estabelecida por:

$$\begin{aligned}
 P(\mathbf{G}_i \cup \mathbf{G}_j, \mathbf{G}_t) &= \frac{1}{|\mathbf{G}_i \cup \mathbf{G}_j| |\mathbf{G}_t|} \sum_{\mathbf{x}_k \in \mathbf{G}_i \cup \mathbf{G}_j} \sum_{\mathbf{x}_l \in \mathbf{G}_t} d(\mathbf{x}_k, \mathbf{x}_l) \\
 P(\mathbf{G}_i \cup \mathbf{G}_j, \mathbf{G}_t) &= \frac{1}{|\mathbf{G}_i \cup \mathbf{G}_j| |\mathbf{G}_t|} \left(\sum_{\mathbf{x}_k \in \mathbf{G}_i} \sum_{\mathbf{x}_l \in \mathbf{G}_t} d(\mathbf{x}_k, \mathbf{x}_l) + \sum_{\mathbf{x}_k \in \mathbf{G}_j} \sum_{\mathbf{x}_l \in \mathbf{G}_t} d(\mathbf{x}_k, \mathbf{x}_l) \right) \\
 P(\mathbf{G}_i \cup \mathbf{G}_j, \mathbf{G}_t) &= \frac{1}{|\mathbf{G}_i \cup \mathbf{G}_j| |\mathbf{G}_t|} \left(\frac{|\mathbf{G}_i|}{|\mathbf{G}_i|} \sum_{\mathbf{x}_k \in \mathbf{G}_i} \sum_{\mathbf{x}_l \in \mathbf{G}_t} d(\mathbf{x}_k, \mathbf{x}_l) + \frac{|\mathbf{G}_j|}{|\mathbf{G}_j|} \sum_{\mathbf{x}_k \in \mathbf{G}_j} \sum_{\mathbf{x}_l \in \mathbf{G}_t} d(\mathbf{x}_k, \mathbf{x}_l) \right) \quad (3.5) \\
 P(\mathbf{G}_i \cup \mathbf{G}_j, \mathbf{G}_t) &= \frac{1}{|\mathbf{G}_i \cup \mathbf{G}_j|} \left(\frac{|\mathbf{G}_i|}{|\mathbf{G}_i| |\mathbf{G}_t|} \sum_{\mathbf{x}_k \in \mathbf{G}_i} \sum_{\mathbf{x}_l \in \mathbf{G}_t} d(\mathbf{x}_k, \mathbf{x}_l) + \frac{|\mathbf{G}_j|}{|\mathbf{G}_j| |\mathbf{G}_t|} \sum_{\mathbf{x}_k \in \mathbf{G}_j} \sum_{\mathbf{x}_l \in \mathbf{G}_t} d(\mathbf{x}_k, \mathbf{x}_l) \right) \\
 P(\mathbf{G}_i \cup \mathbf{G}_j, \mathbf{G}_t) &= \frac{1}{|\mathbf{G}_i \cup \mathbf{G}_j|} (|\mathbf{G}_i| P(\mathbf{G}_i, \mathbf{G}_t) + |\mathbf{G}_j| P(\mathbf{G}_j, \mathbf{G}_t)) \\
 P(\mathbf{G}_i \cup \mathbf{G}_j, \mathbf{G}_t) &= \frac{|\mathbf{G}_i|}{|\mathbf{G}_i \cup \mathbf{G}_j|} P(\mathbf{G}_i, \mathbf{G}_t) + \frac{|\mathbf{G}_j|}{|\mathbf{G}_i \cup \mathbf{G}_j|} P(\mathbf{G}_j, \mathbf{G}_t)
 \end{aligned}$$

A Equação (3.5) fornece uma maneira de atualizar a matriz D entre os grupos a medida que as etapas do algoritmo vão sendo executadas. Para ilustrar, considere-se as mesmas observações usadas no exemplo anterior para o *Ward's* e como métrica considere a distância de Manhattan definida pela Equação (3.6).

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^n |x_{ik} - x_{jk}| \quad (3.6)$$

Tabela 2 – Conjunto de Dados $\mathbf{X}$

Observações	Atributos	
	x_1	x_2
$\mathbf{x}_1$	2.00	2.50
$\mathbf{x}_2$	1.00	2.00
$\mathbf{x}_3$	0.80	2.30
$\mathbf{x}_4$	4.00	3.50

$$D = \begin{pmatrix} 0.00 & & & \\ 1.50 & 0.00 & & \\ 1.40 & 0.50 & 0.00 & \\ 3.00 & 4.50 & 4.40 & 0.00 \end{pmatrix}$$

Temos que D define uma relação de dissimilaridade entre as observações e define a distância entre os grupos da partição inicial dada por $\mathbf{P}_0 = \{\{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}\}$. A distância entre os grupos $\{\mathbf{x}_2\}$ e $\{\mathbf{x}_3\}$ é a menor observada, logo são unidos com $P(\{\mathbf{x}_2\}, \{\mathbf{x}_3\}) = 0.50$. Para atualizar a matriz D parte-se da Equação (3.5).

$$\begin{aligned}
 P(\{\mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_1\}) &= \frac{1.40}{2} + \frac{1.50}{2} = 1.45 \\
 P(\{\mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_4\}) &= \frac{4.50}{2} + \frac{4.40}{2} = 4.45 \Rightarrow D = \begin{pmatrix} 0.00 & & & \\ 1.45 & 0.00 & & \\ 3.00 & 4.45 & 0.00 & \end{pmatrix}
 \end{aligned}$$

Na próxima etapa os grupos $\{x_2, x_3\}$ e $\{x_1\}$ serão unidos com $P(\{x_2, x_3\}, \{x_1\}) = 1.45$. Ao final o seguinte dendrograma será obtido:

Semelhante ao *Ward's*, o *Average-Linkage* também tem a característica de favorecer partições de grupos com formas globulares e dificuldades em encontrar grupos com formas arbitrárias.

3.1.3 Complete-Linkage

Neste método, a dissimilaridade entre dois grupos G_i e G_j será determinada pelo maior valor da métrica, $d(.,.)$ calculada sobre pares de observações entre os grupos, tal como define a Equação (3.7):

$$P(G_i, G_j) = \max_{x_l \in G_i, x_k \in G_j} d(x_l, x_k) \quad (3.7)$$

Seja P uma partição que contém dois grupo G_i e G_j . Se o menor valor da dissimilaridade dentre todos os pares de grupos em P , for aquele correspondente aos grupos G_i e G_j , ou seja, $P(G_i, G_j)$, então G_i e G_j serão unidos.

Uma vez que G_i e G_j foram unidos, a dissimilaridade do grupo resultante para um grupo $G_t \in P$ qualquer pode ser calculada por:

$$\begin{aligned} P(G_i \cup G_j, G_t) &= \max_{x_l \in G_i \cup G_j, x_k \in G_t} d(x_l, x_k) \\ P(G_i \cup G_j, G_t) &= \max\left\{ \max_{x_l \in G_i, x_k \in G_t} d(x_l, x_k), \max_{x_l \in G_j, x_k \in G_t} d(x_l, x_k) \right\} \\ P(G_i \cup G_j, G_t) &= \max\{P(G_i, G_t), P(G_j, G_t)\} \end{aligned} \quad (3.8)$$

Retornando à matriz de dissimilaridade D dos exemplos anteriores:

$$D = \begin{pmatrix} 0.00 & & & \\ 1.11 & 0.00 & & \\ 1.21 & 0.36 & 0.00 & \\ 2.23 & 3.35 & 3.41 & 0.00 \end{pmatrix}$$

a dissimilaridade $P(\{\mathbf{x}_2\}, \{\mathbf{x}_3\}) = 0.36$ indica que $\{\mathbf{x}_2\}$ e $\{\mathbf{x}_3\}$ serão unidos. Para atualizar D usa-se a Equação (3.8):

$$\begin{aligned} P(\{\mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_1\}) &= \max\{1.11, 1.21\} = 1.21 \\ P(\{\mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_4\}) &= \max\{3.35, 3.41\} = 3.41 \end{aligned} \Rightarrow D = \begin{pmatrix} 0.00 & & & \\ 1.21 & 0.00 & & \\ 2.23 & 3.41 & 0.00 & \end{pmatrix}$$

Na próxima iteração, $P(\{\mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_1\}) = 1.21$ é a menor dissimilaridade calculada, logo $\{\mathbf{x}_2, \mathbf{x}_3\}$ e $\{\mathbf{x}_1\}$ são unidos. Ao final das iterações o seguinte dendrograma é obtido:

O *Complete-Linkage* também favorece partições de grupos com formas globulares. Além disso, o algoritmo tem a tendência de quebrar grupos maiores. A Figura 8 ilustra uma situação onde se deseja obter a partição que reporte os dois grupos de interesse. Já a Figura 9 apresenta uma partição sobre o dendrograma obtido pelo *Average-Linkage*.

Figura 8 – Exemplo de partição obtida sobre um corte horizontal na hierarquia *Complete-Linkage*.

Figura 9 – Exemplo de partição obtida sobre um corte horizontal na hierarquia *Average-Linkage*.

3.1.4 *Single-Linkage*.

O *Single-Linkage* ao contrário do *Complete-Linkage* define que a dissimilaridade entre dois grupos G_i e G_j de uma determinada partição P , Equação (3.9), é determinada pelo menor valor da métrica $d(.,.)$ entre observações dos dois grupos em questão, se a mesma estabelecer uma relação de dissimilaridade observações.

$$P(G_i, G_j) = \min_{x_l \in G_i, x_k \in G_j} d(x_l, x_k) \quad (3.9)$$

Os grupos G_i e G_j serão unidos se fornecerem o menor valor de dissimilaridade $P(G_i, G_j)$ dentre todos os pares de grupo em P . A dissimilaridade do grupo resultante da união de G_i e G_j para um terceiro grupo $G_t \in P$ qualquer pode ser estabelecida por:

$$\begin{aligned} P(G_i \cup G_j, G_t) &= \min_{x_l \in G_i \cup G_j, x_k \in G_t} d(x_l, x_k) \\ P(G_i \cup G_j, G_t) &= \min \left\{ \min_{x_l \in G_i, x_k \in G_t} d(x_l, x_k), \min_{x_l \in G_j, x_k \in G_t} d(x_l, x_k) \right\} \\ P(G_i \cup G_j, G_t) &= \min \{ P(G_i, G_t), P(G_j, G_t) \} \end{aligned} \quad (3.10)$$

Considerando a mesma matriz de dissimilaridade dos exemplos anteriores:

$$D = \begin{pmatrix} 0.00 & & & \\ 1.11 & 0.00 & & \\ 1.21 & 0.36 & 0.00 & \\ 2.23 & 3.35 & 3.41 & 0.00 \end{pmatrix}$$

tem-se que os grupos $\{x_2\}$ e $\{x_3\}$ serão mais uma vez os primeiros a se unirem com $P(\{x_2\}, \{x_3\}) = 0.36$. Para atualizar a dissimilaridade do grupo resultante com os demais grupos pode-se aplicar a Equação (3.10):

$$\begin{aligned} P(\{x_2, x_3\}, \{x_1\}) &= \min\{1.11, 1.21\} = 1.11 \\ P(\{x_2, x_3\}, \{x_4\}) &= \min\{3.35, 3.41\} = 3.35 \end{aligned} \Rightarrow D = \begin{pmatrix} 0.00 & & & \\ 1.11 & 0.00 & & \\ 2.23 & 3.35 & 0.00 & \end{pmatrix}$$

A iteração seguinte indica que $\{x_2, x_3\}$ e $\{x_1\}$ são unidos com $P(\{x_2, x_3\}, \{x_1\}) = 1.21$. Procedendo com as demais iterações o dendrograma resultante é dado por:

Embora tenha a vantagem de ser capaz de gerar partições de grupos com formas arbitrárias, o *Single-Linkage* é comumente conhecido por seu "*chaining effect*", que o torna muito sensível à presença de ruído nos dados. A Figura 10 mostra como este efeito pode afetar a construção do dendrograma e produzir partições que podem não ser adequadas.

Figura 10 – *Single-Linkage* e o conhecido "*chaining effect*". Em (c) as poucas observações encadeadas destacadas conectam regiões distantes produzindo soluções de agrupamento inesperadas.

Uma forma alternativa e eficiente de obter uma hierarquia *Single-Linkage* é através de uma Árvore Geradora Mínima (AGM) (JAIN; DUBES, 1988). Nessa perspectiva, representa-se cada observação como um vértice e cada aresta conectando um dado par de vértice é ponderada pela dissimilaridade entre as respectivas observações.

Desta forma pode-se construir um grafo completo e então aplicar algoritmos como os de Prim ou Kruskal para encontrar a AGM e aplicar o Algoritmo 2 para construir o dendrograma correspondente ao algoritmo *Single-Linkage*. O dendrograma é exatamente o mesmo que seria obtido executando o algoritmo tradicional porém nesse caso ele é construído de forma divisiva (de "cima para baixo").

Para ilustrar considere as observações descritas por dois atributos numéricos ilustrados pela subfigura 11a. Tomando como métrica a distância Euclidiana e determinando as dissimilari-

Algoritmo 2: Algoritmo *Single-Linkage* a partir de uma AGM**Entrada:** AGM.**1 início:****2** Coloque todas as observações em um único grupo**3 repita****4** Remova a aresta de maior peso (arestas com mesmo peso são removidas simultaneamente).**5** Configure o valor de escala no dendrograma ao peso da aresta removida.**6** Defina novos grupos colocando em um mesmo grupo as observações de cada componente (subgrafo) conexo resultante.**7 até** Não há arestas a serem removidas;**8 fim.**

dades entre todos os pares de observações obtém-se um grafo completo e então a AGM resultante mostrada pela subfigura 11b.

Figura 11 – Exemplo para ilustrar o *Single-Linkage* sob a perspectiva de algoritmos em grafos.

Aplicando o Algoritmo 2 sobre a AGM resultante obtém-se o seguinte dendrograma:

3.1.5 HDBSCAN*

O HDBSCAN* (*Hierarchical DBSCAN**) (CAMPELLO *et al.*, 2013a; CAMPELLO *et al.*, 2015) é um algoritmo hierárquico baseado em densidade. A ideia é que os grupos nas partições sejam regiões de alta concentração de observações (regiões de alta densidade) separados por regiões de baixa concentração de observações (regiões de baixa densidade). As noções de densidades se dão pelas seguintes definições (ESTER *et al.*, 1996; CAMPELLO *et al.*, 2015):

Definição 3. (Observações *Core* e *Noise*): Uma observação $\mathbf{x}_p \in \mathbf{X}$ é denominada *core* se para um valor de $\varepsilon \in \mathbb{R}$ positivo e $m_{pts} \in \mathbb{N}$ positivo, o conjunto $\mathbf{N}_\varepsilon(\mathbf{x}_p) = \{\mathbf{x} \in \mathbf{X} \mid d(\mathbf{x}_p, \mathbf{x}) \leq \varepsilon\}$ contém pelo menos m_{pts} elementos, incluindo $\mathbf{x}_p$ como elemento, sendo $d(.,.)$ a métrica pré-definida que estabelece uma relação de dissimilaridade entre os pares de observação. Uma observação $\mathbf{x}_p \in \mathbf{X}$ é denominada *noise* se $\mathbf{x}_p$ não é uma observação *core*.

Definição 4. (ε -alcançáveis): Duas observações *core* $\mathbf{x}_p$ e $\mathbf{x}_q$, $\mathbf{x}_p, \mathbf{x}_q \in \mathbf{X}$, são ε -alcançáveis para um valor de $\varepsilon \in \mathbb{R}$ positivo e $m_{pts} \in \mathbb{N}$ positivo se $\mathbf{x}_q \in \mathbf{N}_\varepsilon(\mathbf{x}_p)$ e $\mathbf{x}_p \in \mathbf{N}_\varepsilon(\mathbf{x}_q)$.

Definição 5. (Densamente Conectados): Duas observações *core* $\mathbf{x}_p$ e $\mathbf{x}_q$, $\mathbf{x}_p, \mathbf{x}_q \in \mathbf{X}$, são densamente conectados para um valor de $\varepsilon \in \mathbb{R}$ positivo e $m_{pts} \in \mathbb{N}$ positivo, se são diretamente ou transitivamente ε -alcançáveis.

Definição 6. (Grupo): Um grupo $\mathbf{G}$ para um valor de $\varepsilon \in \mathbb{R}$ positivo e $m_{pts} \in \mathbb{N}$ positivo é um conjunto máximo não-vazio de observações tal que todo par de observações em $\mathbf{G}$ são densamente conectados.

Definição 7. (Distância *core*): A distância *core* de uma observação $\mathbf{x}_p \in \mathbf{X}$, dada por $d_{core}(\mathbf{x}_p)$, para um dado valor de $m_{pts} \in \mathbb{N}$ positivo, é o menor $\varepsilon \in \mathbb{R}$ positivo tal que $\mathbf{x}_p$ é uma observação *core*.

Definição 8. (Distância de Alcançabilidade Mútua): A distância de alcançabilidade mútua entre dois objetos $\mathbf{x}_p$ e $\mathbf{x}_q$, $\mathbf{x}_p, \mathbf{x}_q \in \mathbf{X}$, para um dado valor de $m_{pts} \in \mathbb{N}$ positivo, é definida por $d_{alcm}(\mathbf{x}_p, \mathbf{x}_q) = \max\{d_{core}(\mathbf{x}_p), d_{core}(\mathbf{x}_q), d(\mathbf{x}_q, \mathbf{x}_p)\}$, isto é, $d_{alcm}(\mathbf{x}_p, \mathbf{x}_q)$ é o menor $\varepsilon \in \mathbb{R}$ que faz com que $\mathbf{x}_p$ e $\mathbf{x}_q$ sejam ε -alcançáveis.

Figura 12 – Em (a) as observações $\mathbf{x}_p, \mathbf{x}_q$ e $\mathbf{x}_r$ são *cores*, enquanto que $\mathbf{x}_s$ é *noise*; $\mathbf{x}_q$ e $\mathbf{x}_r$ são ε -alcançáveis e $\mathbf{x}_p$ e $\mathbf{x}_q$ são densamente conectados. Em (b) são apresentadas as noções de Distância *core* e de Distância de Alcançabilidade Mútua.

Fonte: Adaptada de [Ankerst et al. \(1999\)](#).

A hierarquia de grupos HDBSCAN* também pode ser obtida a partir do Algoritmo 2, pois é possível demonstrar que HDBSCAN*, exceto pela modelagem de ruído (noise), é equivalente a executar o *Single-Linkage* em um espaço transformado de distâncias de alcançabilidade mútua, $d_{alcm}(\cdot, \cdot)$. Em particular, dada a matriz de dissimilaridade formada pelas $d_{alcm}(\cdot, \cdot)$ entre os pares de observações, obtém-se um grafo completo representando cada observação como um vértice e cada aresta conectando um dado par de vértice ponderada pela respectiva $d_{alcm}(\cdot, \cdot)$. Calcula-se então uma árvore geradora mínima (AGM) deste grafo. Adiciona-se então, para cada vértice, uma *self*-aresta que conecta o vértice a ele mesmo, ponderada pelo valor $d_{core}(\mathbf{x})$ da observação, para que o conceito de *noise* (ruído) possa ser modelado. Assim, conforme a Definição 7, enquanto a *self*-aresta não for removida indicará que a observação é um *core* e quando removida indicará que a observação tornou-se *noise*.

Para ilustrar, considere a AGM representada pela Figura 13 obtida sobre o espaço transformado de $d_{alcm}(\cdot, \cdot)$ dos dados anteriormente apresentados pela subfigura 11a, assumindo como métrica $d(\cdot, \cdot)$ a distância Euclidiana e $m_{pts} = 3$.

Pela figura, tomando-se qualquer $\varepsilon \geq 0.6742$ todos os pares de observações serão densamente conectados conforme a Definição 6 e formarão um único grupo. Por outro lado, um valor de $\Delta < 0.6742$ faz da observação $\mathbf{x}_5$ um *noise* pela Definição 7 e gera também dois subgrafos conexos e distintos com pares de observação densamente conectados, isto é, dois grupos diferentes. Assim, pelo Algoritmo 2, ao remover as arestas de valor 0.6742 reporta-se 2 grupos e a

observação $\mathbf{x}_5$ como *noise*.

Figura 13 – AGM no espaço de $d_{alcm}(\cdot, \cdot)$ com as *self*-arestas, d_{core} , de cada observação.

Espaço Transformado

Em seguida, quando as arestas de valor 0.5239 são removidas o grupo que antes reportava como observações $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$ perde a observação $\mathbf{x}_4$ que se torna *noise* para valores inferiores ao valor 0.5239. Nesse sentido, para o valores de escala menores que 0.5239 e maiores e iguais a 0.4561 reporta-se o grupo $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, o grupo $\{\mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8\}$ e como *noise* as observações $\mathbf{x}_4$

e $\mathbf{x}_5$. Conforme as próximas iterações seguem o dendrograma resultante para este exemplo é representado pela Figura 14, as linhas pontilhadas a partir de um certo valor de escala indica quando as observações se tornam *noise*.

Figura 14 – Dendrograma correspondente à hierarquia HDBSCAN* para $m_{pts} = 3$ aplicando o Algoritmo 2 sobre a AGM estendida da Figura 13.

A distância de alcançabilidade mútua pode ser vista como uma transformação espacial do espaço original das observações. Essa transformação permite que em regiões onde a densidade é baixa, valores de $m_{pts} > 2$ fazem com que a dissimilaridade entre observações nesta região sejam alongadas. Quando $m_{pts} = 1$ ou $m_{pts} = 2$ a $d_{alcm}(\cdot, \cdot)$ na Definição 8 se reduz a $d(\mathbf{x}_p, \mathbf{x}_q)$, e, portanto, a AGM acaba sendo estabelecida no espaço original e o resultando o mesmo que o *Single-Linkage*. A vantagem de se usar valores de $m_{pts} > 2$ é que o *chaining effect* do *Single-Linkage* pode ser contornado alongando as distâncias em regiões de baixa densidade evitando soluções de agrupamentos inesperadas como aquela na Figura 10.

3.1.6 Hierarquias Simplificadas

Como se pode observar para a hierarquia HDBSCAN*, determinados valores de escala podem indicar a divisão de um grupo em outros, ou apenas indicar que um grupo diminui de tamanho pois uma ou mais observações tornaram-se *noise* a partir de um certo limiar. No exemplo da Figura 14, o grupo G_1 primeiro reduz de tamanho ao perder a observação $\mathbf{x}_5$, então se subdivide nos grupos G_2 e G_3 que nunca de fato se subdividem, mas apenas diminuem de tamanho até desaparecerem por completo. Trata-se portanto que apenas 3 grupos precisam de fato ser reportados, em contraste a 12 em um dendrograma tradicional. Embora essa característica não

esteja presente para os demais algoritmos hierárquicos descritos, uma forma de estendê-la para quaisquer tipo de dendrograma é através de um parâmetro que indica o valor mínimo do tamanho do grupo (*minimum cluster size* - m_{CLSize}) (CAMPELLO *et al.*, 2013b). A ideia consiste em reportar apenas grupos ao longo do dendrograma que contenham pelo menos m_{CLSize} quantidades de observações e indicar limiares para observações que tornam-se *noise* por pertencerem a grupos espúrios com tamanho inferior a m_{CLSize} .

Para ilustrar considere, por exemplo, as observações e o dendrograma *Ward's* representado pela Figura 15.

Figura 15 – Exemplo de observações e seu respectivo dendrograma *Ward's*.

Suponha que é desejado obter um dendrograma simplificado reportando apenas grupos com pelo menos 3 observações ($m_{CLSize} = 3$). Partindo-se do início do dendrograma da subfigura 15b em direção aos níveis inferiores, pode-se seguir o seguinte raciocínio: G_1 (o grupo com todas as observações) é formado no limiar 3.32. Abaixo desse limiar, G_2 e G_3 passam a existir com quantidade de observações maior que o m_{CLSize} desejado, portanto devem ser reportados no dendrograma final.

Originalmente, o grupo G_3 deixaria de existir para valores menores que o limiar 0.54 dividindo-se nos grupos $G_{21} = \{x_4\}$ e G_6 . Entretanto, G_{21} não deverá ser reportado como um grupo para o novo dendrograma simplificado pois possui um número de observações menor que m_{CLSize} . Nesse caso, considera-se que x_4 é *noise* para valores menores que o limiar 0.54 e que G_3 ainda continua sendo um mesmo grupo, só que com apenas 3 observações.

Quando o limiar 0.28 (instante em que G_6 seria formado no dendrograma original) é atingido, G_3 passa a ter um número menor do que m_{CLSize} observações. A partir desse limiar considera-se que as observações em G_3 tenham se tornado ruído e G_3 deixa de existir, isto é, a partir desse limiar já não é possível reportar G_3 com mais de três observações. Essa ideia segue com o grupo G_2 até que não seja mais possível reportar por grupos. Ao final o seguinte dendrograma é obtido:

3.2 Conclusão

Este capítulo apresentou os principais algoritmos de agrupamentos de dados capazes de produzirem hierarquias de grupos, em particular algoritmos que constroem uma hierarquia de grupos seguindo uma abordagem aglomerativa ou divisiva.

As características e possíveis cenários de aplicações desses algoritmos foram discutidas. O capítulo a seguir discutirá o algoritmo proposto por [Dasgupta e Hsu \(2008\)](#) que opera sobre tais hierarquias de grupos.

CLASSIFICAÇÃO ATIVA COM HIERARQUIAS DE GRUPOS

Este capítulo aborda o algoritmo ativo aplicado em problemas de classificação considerado por esta pesquisa. O algoritmo foi originalmente proposto por [Dasgupta e Hsu \(2008\)](#) e é um algoritmo ativo capaz de operar em quaisquer hierarquias de grupos. O algoritmo não utiliza por completo um dendrograma pois os níveis de escala não se fazem necessários, apenas a estrutura hierárquica de grupos.

Este capítulo inicia introduzindo o algoritmo originalmente proposto pelos autores e, em seguida, aborda suas principais características. No final, alguns trabalhos relacionados e propostas de melhorias já estabelecidas na literatura são apresentadas.

4.1 Amostragem Hierárquica

O algoritmo ativo proposto por [Dasgupta e Hsu \(2008\)](#) - denominado neste trabalho de *Hierarchical Sampling* (HS) - tem como propósito determinar uma partição de grupos na hierarquia fornecida como entrada, associando cada grupo da partição aos possíveis rótulos de classe do problema de classificação.

A ideia é que o rótulo de classe definido para cada um dos grupos seja representativo para que ao final do algoritmo seja possível estabelecer com alta precisão a classe para as demais observações não consultadas. A Figura 16 ilustra uma solução geral do algoritmo.

A cada informação introduzida após a consulta ao oráculo, o HS determina estatísticas a respeito das proporções de observações por classes e estimativas para os erros de rotulações para as demais observações não consultadas. Tais estatísticas permitem estabelecer formas para encontrar os rótulos representativos, a partição de grupos e a elaboração de uma estratégia de consulta.

Figura 16 – Exemplo de solução do algoritmo HS. Em (a) a distribuição das observações considerando um problema de classificação com 5 possíveis classes; em (b) a partição encontrada pelo HS, as cores diferenciam os rótulos de classes definidos para cada grupo pelo algoritmo; em (c) a solução final do HS, as observações em preto foram consultadas ao oráculo, as demais têm suas saídas definidas pelos rótulos de seus grupos correspondentes.

4.1.1 Determinando Rótulos Representativos

Definida a hierarquia de grupos para o algoritmo e sabendo que o problema de classificação trata-se de $\mathbf{L} = \{1, \dots, k\}$ possíveis classes, tem-se que a proporção $p_{i,l}(t)$ de observações rotuladas para cada rótulo $l \in \mathbf{L}$, dado que t observações foram consultadas é dada por:

$$p_{i,l}(t) = \frac{N_{i,l}(t)}{\sum_{j=1}^k N_{i,j}(t)} \quad (4.1)$$

em que $N_{i,l}(t)$ é a quantidade de observações consultadas e rotuladas com o rótulo l a partir do grupo i .

Tal proporção trata-se de uma proporção relacionada ao número de observações consultadas e pode não representar a proporção real $p_{i,l}$, que não é conhecida. A medida que as consultas são realizadas $p_{i,l}(t)$ se aproxima de $p_{i,l}$ e serão iguais quando todas as observações do grupo forem consultadas. Por operar em um cenário ativo é de interesse que com poucas consultas o algoritmo possa estabelecer uma aproximação dado de $p_{i,l}$ tal que decisões a respeito da partição de grupos e dos rótulos representativos possam ser tomadas com alta precisão.

Há ferramentas estatísticas que oferecem formas de se estimar valores para proporções reais a partir de uma amostra realizada sobre os dados, uma delas é a estimação intervalar. Entretanto, a construção tradicional de intervalos de confiança para proporções assume um tamanho mínimo para a amostra (por exemplo, para a construção de um intervalo de confiança é comum assumir que o tamanho da amostra retirada da população em estudo seja maior ou igual a 30) (SCHEFLER, 1988), mas no cenário em que HS está configurado nenhuma informação ou apenas uma fração de rótulos está inicialmente disponível. Portanto, outras maneiras de se estabelecer um intervalo para proporções devem ser consideradas.

Para cada grupo i e rótulo l um intervalo de confiança $[p_{i,l}^{LI}(t), p_{i,l}^{LS}(t)]$ (veja o Apêndice A para detalhes da estimação intervalar) é mantido, no qual espera-se conter, com certa confiança, a probabilidade de que um objeto aleatoriamente consultado a partir do grupo i tenha rótulo l :

$$[p_{i,l}^{LI}(t), p_{i,l}^{LS}(t)] = [\max(p_{i,l}(t) - \Delta(t), 0), \min(p_{i,l}(t) + \Delta(t), 1)], \quad (4.2)$$

A partir do intervalo, é possível definir uma estimação para o mínimo $\varepsilon_{i,l}^{LI}(t)$ e máxima $\varepsilon_{i,l}^{LS}(t)$ proporção de observações da base de dados que teriam um rótulo diferente de l :

$$[1 - p_{i,l}^{LS}(t), 1 - p_{i,l}^{LI}(t)] = [\varepsilon_{i,l}^{LI}(t), \varepsilon_{i,l}^{LS}(t)] \quad (4.3)$$

Quando para algum rótulo l^* sua estimativa máxima $\varepsilon_{i,l^*}^{LS}(t)$ se torna menor do que β vezes $\varepsilon_{i,l}^{LI}(t)$ para todo $l \neq l^*$, então l^* é considerado o rótulo representativo para o grupo i :

$$l^* = l' \in \{1, \dots, k\} \Rightarrow \varepsilon_{i,l'}^{LS}(t) < \beta \varepsilon_{i,l}^{LI}(t), \forall l \neq l' \quad (4.4)$$

Um conjunto $\mathcal{A}(t)$ de pares (i, l^*) para o qual a inequação (4.4) é ou foi verdadeira para algum $t' \leq t$ é mantido. Qualquer grupo i listado em $\mathcal{A}(t)$ é dito ter um rótulo representativo. Um rótulo estritamente representativo é aquele que satisfaz a inequação (4.4) na iteração t atual. Quando o grupo i está em mais de um par pertencente a $\mathcal{A}(t)$ com diferentes l^* , então o rótulo majoritário do grupo i é considerado o rótulo representativo.

4.1.2 Determinando uma Partição na Hierarquia

A cada iteração do algoritmo uma partição $\mathbf{P}(t)$ de grupos é mantida. É a partir dessa partição que um grupo é escolhido (conforme é discutido na Seção 4.1.3) e uma observação é aleatoriamente selecionada deste grupo para ser consultada ao oráculo.

Após t consultas, $\mathbf{P}(t)$ segue a partir de $\mathcal{A}(t)$ substituindo os grupos $i \in \mathbf{P}$ se i possui rótulo representativo l^* e se a estimativa dos erros dos subgrupos de i são menores do que a estimativa de erro que i mantém sobre l^* . Considerando i e seu rótulo representativo l^* , uma estimativa conservadora para o erro de classificação ao se classificar as observações em i com o rótulo l^* é dada por $\varepsilon_{i,l^*}^{LS}(t)$. Quando um grupo i não possui um rótulo representativo então uma estimativa ainda mais conservadora para o erro de i é utilizada, dada por 1¹:

$$\varepsilon_i(t) = \begin{cases} \varepsilon_{i,l^*}^{LS}(t) & \text{se } i \text{ tem um rótulo representativo } l^* \\ 1 & \text{caso contrário} \end{cases} \quad (4.5)$$

¹ Em Dasgupta e Hsu (2008) o erro é definido como $1 - p_{i,l^*}(t)$ quando $(i, l^*) \in \mathcal{A}(t)$. A descrição em (4.5) segue conforme está na implementação do algoritmo disponibilizada pelos autores.

A partir de (4.5), para um grupo i pode-se, recursivamente, definir o erro mínimo para a sub-árvore T_i que tem como raiz o grupo i seguindo: (i) se i é um grupo folha, então $\varepsilon_i^{T_i}(t) = \varepsilon_i(t)$; (ii) se i é qualquer outro grupo que não seja folha, então:

$$\varepsilon_i^{T_i}(t) = \min \left\{ \varepsilon_i(t), \sum_j \frac{N_j}{N_i} \varepsilon_j^{T_j}(t) \right\}, \quad (4.6)$$

em que j são os grupos filhos do grupo i . Seguindo a relação dada pela Equação (4.6) qualquer grupo $i \in \mathbf{P}$ que tem um rótulo representativo l^* será recursivamente substituído por seus filhos se a estimativa do erro $\varepsilon_i(t)$ é maior que a estimativa do erro ponderada ($\sum_j \frac{N_j}{N_i} \varepsilon_j^{T_j}(t)$) das sub-árvores que tem como raiz seus filhos. Esse procedimento é análogo ao método de poda baseado em erro em árvores de decisão, e pode ser realizado de forma eficiente com uma passagem “de baixo para cima” na hierarquia.

Após cada nova observação consultada as estimativas para as proporções para cada rótulo, as estimativas para os erros, o conjunto $\mathcal{A}(t)$ e a atualização de $\mathbf{P}(t)$ podem ser eficientemente atualizados.

4.1.3 Definindo a Estratégia Ativa de Consulta

A escolha de qual observação será apresentada ao oráculo é realizada de maneira aleatória. A estratégia ativa cabe somente na escolha de qual grupo $i \in \mathbf{P}(t)$ a partir do qual essa observação será escolhida.

Para isso a estratégia estabelecida foi selecionar o grupo com probabilidade proporcional à estimativa de $\varepsilon_i(t)$ multiplicada pela fração de observações do subconjunto de dados que o grupo mantém, $w_i(t) = \frac{N_i}{N}$, em que N é o número de observações do conjunto de dados e N_i é a quantidade de observações do grupo i . A ideia é atribuir maiores chances de seleção aos grupos maiores e com maior incerteza sobre seu rótulo representativo, ou seja, uma combinação de representatividade e informatividade.

4.1.4 Transdução dos Rótulos Representativos

Uma vez que o processo ativo foi finalizado, para cada grupo na partição $\mathbf{P}(t)$ os respectivos rótulos representativos são usados como rótulos de classe para todas as observações do grupo, incluindo aquelas consultadas.

Se um determinado grupo $i \in \mathbf{P}(t)$ não possui rótulo representativo, então o rótulo que será atribuído para suas observações será o rótulo representativo de seu pai, que pela descrição dada pela Seção 4.1.2 terá necessariamente um rótulo representativo definido.

O Algoritmo 3 descreve de modo geral os passos do HS.

Algoritmo 3: Algoritmo HS

Entrada: Hierarquia de Grupos: $\mathbf{H}$.

```

1 início:
2    $t = 0$ 
3   Coloque na partição inicial  $\mathbf{P}(t)$  a raiz de  $\mathbf{H}$ ;
4   enquanto o oráculo está disposto a rotular faça:
5     Selecciona um grupo  $i$  em  $\mathbf{P}(t)$ ;
6     Selecione aleatoriamente uma observação  $o \in i$ ;
7     Consulte  $o$  ao especialista;
8     Partindo-se do grupo folha de que contém  $o$  a caminho de  $i$ : estabeleça os limites
      para as proporções de classe (Equação (4.2)), as estimativas para os erros (Equação
      (4.3)), os rótulos representativos (Inequação (4.4)), o conjunto de pares  $\mathcal{A}(t)$  e os
      cálculos para  $\varepsilon_i(t)$  e  $\varepsilon_i^{T_i}(t)$  para cada grupo ao longo desse caminho;
9     Atualize os grupos em  $\mathbf{P}(t)$ ;
10     $t = t + 1$ ;
11  fim-enquanto
12  para  $i \in \mathbf{P}(t)$  faça:
13    se  $i$  tem um rótulo representativo  $l^*$  então:
14      classifique as observações em  $i$  com rótulo  $l^*$ ;
15    fim-se
16    se-não
17      classifique as observações em  $i$  com rótulo  $l^*$  de seu pai;
18    fim-se
19  fim-para
20 fim.

```

4.2 Trabalhos Relacionados

O HS apresenta aspectos positivos que o coloca a frente de outros algoritmos ativos, tanto aqueles que descrevem estratégias informativas quanto representativas.

O algoritmo, como discutido por Dasgupta e Hsu (2008), é capaz de contornar o problema do viés de amostragem que estratégias baseadas em incertezas estão sujeitos. A Figura 17 ilustra como uma solução obtida pelo HS é capaz de contornar o problema do viés de amostragem.

Todo processo ativo do algoritmo é construído a partir do que já foi consultado ao especialista, possibilitando que suas escolhas futuras possam ser aprimoradas, diferente do ocorre com algoritmos ativos dentro da categoria de abordagem representativa discutidos na Seção 2.3.2. Além disso, a estratégia ativa para a seleção de grupos é estabelecida ponderando um fator de densidade a um fator de incerteza, estabelecendo assim uma combinação entre representatividade e informatividade que são aspectos importantes para o desenvolvimento de estratégias de consultas (SETTLES *et al.*, 2008; AODHA *et al.*, 2014; HUANG *et al.*, 2010).

Outro aspecto positivo do HS é a partição $\mathbf{P}(t)$ que não se trata necessariamente de um corte horizontal na hierarquia. Um corte não horizontal em determinados cenários é fundamental para estabelecer uma boa solução de agrupamento. A Figura 18 apresenta um cenário onde um

Figura 17 – Exemplo de solução do algoritmo HS. Em (a) a distribuição das observações conforme apresentadas anteriormente pela Figura 1, em (b) a partição encontrada pelo HS que permite contornar o problema do viés de amostragem.

corde não horizontal representa uma solução para o problema que não poderia ser capturada por um corte horizontal.

Figura 18 – Exemplo de solução do algoritmo HS que se trata de um corte não horizontal na hierarquia de grupos.

Um algoritmo semi-supervisionado que opera sobre cortes horizontais em dendrogramas é o *cluster-then-label (CTL)* (ZHU; GOLDBERG, 2009). Nesse algoritmo, sabendo que o problema trata-se de k possíveis classes, obtém-se um dendrograma usando os dados $\mathbf{X}_l$ e $\mathbf{X}_u$ e um corte horizontal que reporte por k grupos. A partir de $\mathbf{X}_l$ determina-se para cada um dos grupos o rótulo majoritário e o define como classe para todas as observações do grupo. A Figura 19 ilustra uma solução dada pelo CTL, observe que o corte define 5 grupos e espera-se que tais grupos possam estar alinhados as classes do problema e reportar a solução esperada conforme a subfigura 18a, entretanto o resultado (subfigura 19b) após a aplicação do CTL não corresponde a solução esperada.

Figura 19 – Exemplo de solução do algoritmo CLT.

Yang *et al.* (2010), seguindo o paradigma do algoritmo HS, exploram hierarquias de grupos para produzir um conjunto inicial de observações rotuladas para que, em seguida, um classificador SVM seja utilizado para selecionar observações próximas à margem de separação entre as classes. Embora os autores discutam que o propósito da hierarquia de grupos é selecionar observações representativas, a seleção das observações é realizada de forma aleatória e, diferente do algoritmo HS, a homogeneidade de um grupo é avaliada pela Equação (4.7) restringindo sua aplicação para problemas de classificação de duas classes.

$$Pur(\mathbf{G}_i) = \max \left(\frac{|\mathbf{G}_i \cap L^+| + \delta}{|\mathbf{G}_i \cap L| + \delta}, \frac{|\mathbf{G}_i \cap L^-| + \delta}{|\mathbf{G}_i \cap L| + \delta} \right), \quad (4.7)$$

onde $|\mathbf{G}_i \cap L^+|$ e $|\mathbf{G}_i \cap L^-|$ é o número de observações rotuladas com o rótulo $l+$ e $l-$ respectivamente e $|\mathbf{G}_i \cap L|$ é o número total de observações rotuladas no grupo $\mathbf{G}_i$. Se $Pur(\mathbf{G}_i) < \alpha$ então $\mathbf{G}_i$ é considerado homogêneo e não será substituído por seus filhos.

Em Tuia *et al.* (2010) o algoritmo HS é aplicado para a rotulação de pixels em imagens. Além de investigar a seleção de grupos, segundo a estratégia discutida na Seção 4.1.3, os autores investigam a aplicação do algoritmo quando os grupos, no caminho do grupo na partição até o grupo folha que contém a observação, são selecionados com probabilidade proporcional à sua incerteza, conforme a Equação (4.5). No trabalho a estratégia foi propensa à seleção de ruídos enquanto a estratégia original pôde contornar essa situação obtendo melhores resultados, implicando a importância do fator w_i para o critério de escolha das observações.

Em outros trabalhos aplicados para a rotulação de pixel em imagens (TUIA *et al.*, 2012; MUÑOZ-MARÍ *et al.*, 2012) a estratégia conforme discutida na Seção 4.1.3 é estendida para selecionar os subgrupos recursivamente até que uma observação seja selecionada. Em ambos os trabalhos essa estratégia possibilitou uma melhoria do algoritmo para a resposta ao problema de

classificação em comparação com a escolha aleatória das observações.

Bolaños *et al.* (2013) utilizou o HS como o centro de uma aplicação visual para a rotulação de imagens. Iterativamente, imagens são apresentadas ao usuário da aplicação conforme o procedimento de escolha discutido na Seção 4.1.3. Além disso, apresentam-se ao usuário algumas imagens dos grupos em $\mathbf{P}(t)$ com suas classes definidas pelos rótulos representativos, possibilitando que o usuário tenha uma visão antecipada das soluções do HS podendo atribuir rótulos para as imagens que não estão de acordo com a solução.

Sterckx *et al.* (2014) explora técnicas de redução de dimensionalidade afim de construir hierarquias de grupos que possam aprimorar a resposta do algoritmo HS. Nesse trabalho as diferentes técnicas avaliadas resultam em diferentes tipos de hierarquias, levando a diferentes performances de resposta do HS.

ESTRATÉGIAS DE MELHORIAS PARA O HS

As seções a seguir discutem as propostas de melhorias deste trabalho para o HS. Com base em alguns dos trabalhos mencionados no capítulo anterior e a partir das características do HS, discute-se e aponta-se dificuldades do algoritmo e como foram estabelecidas estratégias para contorná-las e melhorar a qualidade de solução.

5.1 Transdução *Bottom-Up*

Uma etapa fundamental do algoritmo é o passo transdutivo após o processo ativo. Como já descrito no Capítulo 4 Seção 4.1.4, cada grupo na partição resultante do algoritmo atribui suas observações ao seu respectivo rótulo representativo l^* . Uma dificuldade dessa estratégia é quando algum grupo $i \in \mathbf{P}(t)$ não possui rótulo l^* estabelecido. Nessa condição, o rótulo definido para as observações em i é dado pelo rótulo representativo de seu pai na hierarquia de grupos.

A Figura 20 ilustra uma situação ideal para o HS, onde há grupos que se alinham perfeitamente às classes do problema, mas a transdução dos rótulos representativos dos grupos na partição final, após 8 consultas, resultaria em uma solução inesperada. Na figura, a hierarquia é obtida a partir de 35 observações e o HS foi aplicado (para $\beta = 2$, o mesmo valor configurado por Dasgupta e Hsu (2008)). Após a oitava consulta a partição final é dada pelos grupos com identificadores 67, 66 e 64. A rotulação a partir deles é capaz de rotular corretamente as observações dos grupos 66 e 64. Para o grupo 67 nenhum rótulo representativo l^* foi estabelecido e portanto as observações do grupo serão rotuladas com o rótulo $l^* = b$ do grupo 69 ((69, b) $\in \mathcal{A}(t)$) e, conseqüentemente, todas as observações de 67 serão incorretamente classificadas.

Esta situação ocorre porque, durante a fase de consulta do algoritmo, é desejável conservar e manter uma partição em níveis superiores da hierarquia. Entretanto, quando o processo ativo é finalizado, propagar os rótulos a partir dos grupos na partição, que mantém grupos candidatos a consulta, implica propagar rótulos para observações $\mathbf{x}$ baseando-se em observações consultadas

Figura 21 – Desempenho do HS sobre os mesmos conjuntos de dados, mas a partir de diferentes hierarquias.

Veja que para os dados artificiais da subfigura 21a a qualidade de resposta do HS sobre uma hierarquia *Single-Linkage* é melhor comparada a resposta dada sobre uma hierarquia *Ward's*, isto é, o erro com o uso da hierarquia *Single-Linkage* é menor quando as mesmas observações são consultadas sobre a hierarquia *Ward's*. Isso se deve as características de ambos algoritmos, nesse caso, o *Single-Linkage* é a hierarquia mais apropriada pois é capaz de reportar por grupos com formas arbitrárias no espaço dos atributos, enquanto que o *Ward's* favorece grupos globulares.

Definir antecipadamente qual hierarquia é mais apropriada a ser fornecida como entrada em um dado problema torna-se importante para a aplicação bem sucedida do algoritmo. Entretanto, esta tarefa está longe de ser trivial.

A escolha da hierarquia poderia ser realizada manualmente pelo especialista observando antecipadamente o dendrograma e identificando se há grupos e se os mesmos podem estar alinhados às classes a partir de frações iniciais de dados rotulados, caso esses estejam disponíveis. Todavia, é possível que nenhum ou poucos rótulos estejam disponíveis no início da execução do algoritmo. Além disso, para bases de dados com numerosa quantidade de observações, analisar grupos ao longo de dendrogramas nem sempre é raramente uma tarefa simples (SANDER *et al.*, 2003; ANKERST *et al.*, 1999).

Afim de evitar essa dependência de entrada do algoritmo, esse trabalho propõe uma estratégia para que o algoritmo opere sobre mais de uma hierarquia de grupos, permitindo que a solução final possa estar de acordo com a solução dada pela hierarquia de grupos que melhor representaria o problema.

Múltiplas hierarquias $H_1, \dots, H_n$ são entregues ao algoritmo, que escolherá uma delas para decidir a próxima consulta. Uma vez realizada a consulta todas as hierarquias são processadas (atualizadas) pelo algoritmo de forma independente. Dadas suas respectivas partições $\mathbf{P}_{H_1}(t), \dots, \mathbf{P}_{H_n}(t)$ após t consultas, definiu-se como estimativa conservadora para o erro de uma hierarquia uma média ponderada das estimativas dos erros dos grupos na partição daquela hierarquia, ou seja:

$$\xi_{H_i}(t) = \sum_{j \in \mathbf{P}_{H_i}(t)} \frac{N_j}{N} \varepsilon_j(t) \quad (5.1)$$

Com base para a escolha da próxima consulta, optou-se por sempre selecionar a hierarquia que fornece a menor estimativa do erro, isto é, $\text{argmin}_{H_i} \{\xi_{H_i}(t)\}$. Uma vez realizada a próxima consulta a partir da hierarquia escolhida, pode-se simultaneamente realizar os passos descritos nas Seções 4.1.1, 4.1.2 e 4.1.3 de forma paralela e independente para todas as hierarquias.

Para a rotulação das observações ao finalizar o processo ativo, optou-se por explorar simultaneamente os rótulos de classe que são dados por todas as hierarquias, seguindo a ideia da Transdução *Bottom-Up*. Ao aplicar a estratégia de Transdução *Bottom-Up* independentemente para todas as hierarquias, cada observação $\mathbf{x}$ terá a ela atribuído um rótulo l_i em cada hierarquia H_i , rótulo este obtido a partir de um determinado grupo $c_i \in H_i$. O grupo c_i , cujo rótulo majoritário é l_i tem como estimativa máxima para o erro, conforme discutido na Seção 4.1.1, o valor $\varepsilon_{c_i}(t) = 1 - p_{c_i, l_i}^{LI}(t)$. Seja $\mathbf{C} = \{c_1, \dots, c_n\}$, $c_i \in H_i$, o conjunto de grupos responsáveis pela rotulação da observação $\mathbf{x}$ nas múltiplas hierarquias. O rótulo final para $\mathbf{x}$ será aquele dado pelo grupo com a menor estimativa de erro, isto é, $\text{argmin}_{c_i \in \mathbf{C}} \{\varepsilon_{c_i}(t)\}$.

5.3 Representatividade com Hierarquias de Densidade e Simplificadas

O Capítulo 3 Seções 3.1.5 e 3.1.6, introduziu hierarquias que são capazes de modelar o conceito de observações *noise* a partir de certos limiares de escala do dendrograma. Essa modelagem pode ser útil quando o objetivo é procurar por representatividade nos dados. Sendo assim, hierarquias de densidade, tais como HDBSCAN* e hierarquias simplificadas, podem ser usadas para guiar a seleção de observações em que os rótulos de classe possam ser representativos para as demais observações similares.

Apesar da própria estratégia ativa de consulta do algoritmo, especialmente a extensão

proposta por [Tuia et al. \(2012\)](#) e [Muñoz-Marí et al. \(2012\)](#), se configurar como uma estratégia que busca alcançar representatividade a partir do termo w_i , o processo é guiado probabilisticamente. No uso das hierarquias simplificadas e de densidade o objetivo é focar inicialmente nas observações que são representativas a fim de estabelecer com poucas consultas uma cobertura abrangente e de qualidade para o problema, sem possibilitar a seleção de observações em regiões de alta indecisão como a fronteira entre classes.

Embora o algoritmo HS não utilize os níveis de escala como informação, é possível, pela própria hierarquia de grupos, representar quando determinadas observações se tornam *noise*. Considere, por exemplo, o dendrograma da subfigura 22a. Na subfigura 22b tem-se o dendrograma para um valor de $m_{ClSize} = 2$ e os valores de escalas (linha pontilhada) quando as observações se tornam ruídos (*noise*). Na sub-figura 22c, tem-se a representação da hierarquia correspondente.

Figura 22 – Exemplo de hierarquia de grupos simplificada.

Para a observação x_{13} no nível em que os grupos G_2 e G_3 passam a existir a observação se torna *noise* e por isso, na hierarquia 22c, a observação não é apresentada como um grupo. O mesmo ocorre para a observação x_{10} quando G_7 e G_6 passam a existir.

Para modelar o conceito de *noise* na hierarquia de grupos, associam-se as observações que se tornam *noise* a um grupo “virtual” $G_i^\emptyset$ como ilustra a Figura 23, tal como proposto em Campello *et al.* (2013b).

Para possibilitar que o algoritmo foque nas observações representativas, quando a hierarquia de grupos simplificada é dada como entrada para o algoritmo, todas as observações, exceto aquelas em grupos virtuais, serão candidatas às consultas. A Figura 24 ilustra as observações candidatas para as hierarquias simplificadas HDBSCAN*.

Figura 23 – Hierarquia simplificada representando os grupos virtuais.

Figura 24 – Exemplo de observações candidatas a seleção a partir de hierarquias simplificadas HDBSCAN*.

Percebe-se que observações em regiões de difíceis decisão, tais como aquelas em fronteiras de classes, não serão permitidas de serem consultadas. Nesse sentido, espera-se que as consultas iniciais feitas ao oráculo possam contribuir (sem impor dificuldades quando se permite a consulta daquelas que possam trazer indecisões) para a definição de rótulos representativos ao longo dos grupos na hierarquia.

Para permitir que as regiões em fronteiras de decisão sejam posteriormente consideradas, deve-se possibilitar que as observações nessas regiões sejam consultadas. Rótulos representativos

para grupos virtuais podem ser definidos, entretanto espera-se que as observações nesses grupos sejam de diferentes classe e, portanto, a estimativa do erro (indecisão) a respeito dos rótulos representativos serão altas. Perceba que os grupos folhas também estão sujeitos a esse aspecto. Nesse sentido, quando a partição $\mathbf{P}(t)$ do HS começar a manter tais grupos e tais estimativas, não será possível melhorar a qualidade do algoritmo pois não haverá mais subgrupos e assim a qualidade do algoritmo sobre essas hierarquias estará comprometida. Deve-se portanto recorrer à hierarquia completa para evitar tal situação. Nesse sentido, a hierarquia completa será considerada pelo algoritmo quando uma cobertura abrangente dos dados é encontrada.

Definiu-se como cobertura abrangente o momento que a partição do HS $\mathbf{P}(t)$ na hierarquia completa no instante t define grupos e rótulos representativos, tal que, tenha-se pelo menos um rótulo representativo para cada possível classe do problema. Assim como na estratégia de múltiplas hierarquias, após cada consulta, os passos do HS podem ser feitos independentes para a hierarquia simplificada e completa. No final do processo ativo, a Transdução *Bottom-Up* é feita a partir da hierarquia completa.

Para hierarquias *Ward's*, *Complete-Linkage* e *Average-Linkage*, embora a simplificação estabeleça o conceito de *noise* nesses algoritmos, o que se observa com a simplificação nessas hierarquias é que muitas das observações acabam se tornando *noise* em grupos folhas da hierarquia.

Figura 25 – Observações e seus rótulos de classe definidos pelas cores.

Esse efeito é ilustrado pelas hierarquias da Figura 27. Tais hierarquias são obtidas a partir do dendrograma *Ward's* apresentado pela Figura 26, que por sua vez é obtido a partir das observações da Figura 25.

Na Figura 25 mostra-se interessante que algumas observações, tais como aquelas presentes na região mais escura, fossem reportadas como *noise* pelas hierarquias simplificadas *Ward's*. Entretanto, isso não é o que acontece. Considerando valores para $m_{CLSize} = 2, 3, 4, 5$ e superiores a esses, não seria possível reportar por grupos virtuais que possibilitem guiar a seleção de HS em observações representativas.

Figura 26 – Hierarquia *Ward's*. Os grupos são identificados pelos valores numéricos.

Como se pode observar pela Figura 27, as respectivas hierarquias de grupos simplificadas *Ward's* não reportam por quaisquer grupos virtuais. Nesse sentido, espera-se que a operação do HS sobre tais hierarquias simplificadas tenham um desempenho semelhante a execução do HS sobre a hierarquia completa correspondente.

Figura 27 – Hierarquias Simplificadas *Ward's* obtidas a partir do dendrograma da Figura 26.

A Figura 28 apresenta as observações candidatas para hierarquias simplificadas do *Ward's*, *Complete-Linkage* e *Average-Linkage* para $m_{CLSize} = 10$ sobre os dados da Figura 24a. Para valores menores de m_{CLSize} as observações em nós virtuais são ainda menores. Pode-se

observar que para tais algoritmos a simplificação tem dificuldade de reportar por grupos virtuais com observações em regiões de difíceis decisão.

Figura 28 – Observações candidatas em branco e observações em nós virtuais destacadas em vermelho.

5.4 Complexidade Computacional

Para compreender a complexidade computacional do HS com as melhorias estabelecidas neste trabalho observa-se que o algoritmo possui as seguintes fases: (i) obtenção da hierarquia de grupos, (ii) processo ativo para realização das consultas e (iii) propagação transdutiva dos rótulos.

A complexidade computacional para obter uma hierarquias de grupos é dada por $\mathcal{O}(N^2 \log N)$ ou $\mathcal{O}(N^2)$ para a maioria dos algoritmos de agrupamento de dados (TAN *et al.*, 2005).

O processo ativo de consultas usando uma única hierarquia não pode ter complexidade maior que a geração da hierarquia, pois o número de consultas é limitado a $t = N$ e cada consulta envolve escolher um grupo da partição (que contém no pior caso N grupos) e atualizar as estatísticas dos grupos no caminho partindo da observação consultada até a raiz da hierarquia, que contém no máximo altura igual a $\mathcal{O}(N)$, resultando em uma complexidade de $\mathcal{O}(N^2)$ no pior caso.

A propagação final dos rótulos (após o processo ativo) usando uma única hierarquia tem complexidade igual a $\mathcal{O}(N^2)$, uma vez que há N observações que percorrem, no pior caso, o caminho da observação até a raiz da hierarquia, que tem altura no máximo $\mathcal{O}(N)$. É importante observar que na prática, a altura da hierarquia está longe do pior caso, além disso, o caminho para atualizar as estatísticas e rotular as observações são pequenos comparados à altura da hierarquia, pois interrompe-se o processo ao encontrar o primeiro grupo com rótulo majoritário.

É importante observar também que em um cenário ativo, o gargalo usualmente está no tempo que um especialista está sujeito para atribuir manualmente ou via análise laboratorial os rótulos para as observações, ao invés do tempo relacionado aos cálculos computacionais.

De maneira geral, a complexidade assintótica do método com as melhorias propostas por neste capítulo é limitada pela complexidade de se construir a hierarquia de grupos (que pode ser feito antecipadamente). Uma vez que o HS tal como proposto por [Dasgupta e Hsu \(2008\)](#) também exige a hierarquia de grupos, a complexidade do método aqui proposto quando este faz uso de uma única hierarquia não é maior que a complexidade do HS original. O uso de múltiplas hierarquias afetará a complexidade apenas por um fator constante: a construção da hierarquia, o processo ativo e a fase de propagação dos rótulos são apenas repetidas através das diferentes hierarquias. Mesmo tal constante pode ser removida, se as múltiplas hierarquias forem processadas em um ambiente em paralelo.

EXPERIMENTOS E DISCUSSÕES

Este capítulo apresenta os resultados obtidos pelas estratégias definidas no Capítulo 5. São apresentados os resultados do algoritmo a partir: (i) da estratégia de melhoria de Transdução *Bottom-Up* sobre diferentes hierarquias de grupo; (ii) da estratégia de múltiplas hierarquias e (iii) de hierarquias simplificadas HDBSCAN*. Por último, são apresentados os resultados do algoritmo com as melhorias em relação a outros algoritmos ativos encontrados na literatura.

6.1 Bases de Dados

As bases de dados utilizadas nos experimentos a seguir estão descritas na Tabela 3.

Tabela 3 – Bases de dados.

Base	Número		
	Observações	Atributos	Classes
Banknote	1372	5	2
Breast Cancer Wisconsin (Original)	569	13	3
Dermatology	366	33	6
Detect Malicious	373	513	2
Ionosphere	351	34	2
Iris	150	4	3
Landsat	6435	36	7
Multiple Features	2000	649	10
Optical	5620	64	10
Seeds	210	7	3
Texture	5500	40	11
Wine	178	13	3

As bases foram retiradas do repositório UCI ([BACHE; LICHMAN, 2013](#)) e todas contêm atributos numéricos. Foram usadas as hierarquias de grupos discutidas no Capítulo 3 e para suas construções usou-se como métrica a distância Euclidiana.

6.2 Resultados com Transdução *Bottom-Up*

Para avaliar a estratégia de Transdução *Bottom-Up* descrita no Capítulo 5, Seção 5.1, verificou-se a qualidade de resposta dada pelo algoritmo ao encerrar o processo ativo.

Três observações aleatoriamente escolhidas são estabelecidas como *seeds* para dar início ao processo ativo. A partir de um *10-fold cross-validation*, no qual são permitidas consultas de observações em 9 *folds*, avaliou-se o número de observações que são classificadas incorretamente a partir do *fold* restante estabelecido como teste (que garante não haver nenhuma observação rotulada). A soma de observações incorretamente classificadas sobre todos os 10 *folds* de teste, dividido pelo número total de observações, é reportado como a taxa de erro. Repetiu-se esse processo por 30 vezes¹ variando as observações pré-estabelecidas como *seeds* e reportando a média sobre todas as execuções como o Erro de Transdução².

Comparou-se a estratégia de Transdução *Bottom-Up* com o algoritmo tal como proposto por Dasgupta e Hsu (2008) e descrito no Capítulo 4, Seção 4.1, e também com a variante que utiliza a estratégia modificada de consulta proposta por Tuia *et al.* (2012) e Muñoz-Marí *et al.* (2012). O valor configurado para β na Inequação (4.4) para a admissibilidade de rótulos representativos foi igual a 2, o mesmo sugerido e utilizado nos trabalhos anteriormente citados. Nesses mesmos trabalhos, apenas a hierarquia *Ward's* foi usada como entrada. Os resultados a seguir também são avaliados a partir de hierarquias *Single-Linkage*, *Average-Linkage* e *Complete-Linkage*.

A Figura 29 apresenta o Erro de Transdução sobre hierarquias *Ward's*. O algoritmo HS proposto por Dasgupta e Hsu (2008) é denotado por "HS(Ward)", a versão com modificação da estratégia de consulta proposta por Tuia *et al.* (2012) e Muñoz-Marí *et al.* (2012), descrita na Seção 4.2, é denotada por "HS*(Ward)". A estratégia de Transdução *Bottom-Up* proposta neste trabalho é denotada por "BUHS(Ward)", tal estratégia combinada com a modificação da estratégia de consulta proposta por Tuia *et al.* (2012) e Muñoz-Marí *et al.* (2012) é denotada por "BUHS*(Ward)".

Para as hierarquias *Single-Linkage* (SL), *Complete-Linkage* (CL) e *Average-Linkage* (AL) os resultados são dados pelas Figuras 30, 31 e 32 respectivamente.

Independente da hierarquia de grupos fornecida como entrada é possível verificar que a modificação da estratégia de seleção das observações proposta por Tuia *et al.* (2012) e Muñoz-Marí *et al.* (2012) produz uma melhoria para o algoritmo HS. Das Figuras 29, 30, 31 e 32 fica

¹ permutou-se aleatoriamente a ordem em que as classes aparecem na base de dados antes de cada execução devido à sensibilidade a esta ordem identificada no código do HS, disponível em: <http://www.cs.columbia.edu/~djhsu/>

² Veja que aplicando a fase de transdução, após o processo ativo, sem a utilização dos *folds* sobre toda a base de dados, poderia-se reportar a taxa de erro (a) em relação a todas as observações (incluindo aquelas consultadas), ou (b) reportar a taxa de erro somente sobre aquelas observações que tem seus rótulos definidos na fase transdutiva. Como o número de consultas aumentam, o erro reportado em (a) estaria sujeito e dominado pelas observações consultadas que são observações corretamente rotuladas; no caso (b), o número de observações não rotuladas diminui e, consequentemente, a confiança sobre a taxa do erro reportada.

Figura 29 – Erro de Transdução após cada consulta sobre a hierarquia Ward's

evidente que, de maneira geral, HS* é superior a HS, BUHS* é superior a BUHS e BUHS* são superiores a HS*.

6.3 Resultados com Múltiplas Hierarquias

Os resultados apresentados pela Figura 33 foram obtidos a partir da estratégia de Múltiplas Hierarquias discutida no Capítulo 5 Seção 5.2.

Figura 30 – Erro de Transdução após cada consulta sobre a hierarquia *Single Linkage* (SL)

Foram dadas como entradas as hierarquias *Ward's* (Ward), *Complete-Linkage* (CL), *Average-Linkage* (AL) e *Single-Linkage* (SL). O algoritmo proposto com uso de múltiplas hierarquias é denotado por "M-BUHS*(Ward,SL,CL,AL)". Também são pareados os resultados do BUHS* individualmente para cada hierarquia, a fim de perceber as diferenças entre soluções do mesmo algoritmo utilizando apenas uma ou múltiplas hierarquias.

Pode-se perceber que não há uma hierarquia única que garante melhores resultados em todas as bases de dados e que o desempenho individual de cada hierarquia pode variar

Figura 31 – Erro de Transdução após cada consulta sobre a hierarquia *Complete Linkage* (CL)

significativamente para diferentes bases. Por exemplo, o *Single-Linkage* tem uma qualidade de resposta destacada para as bases Banknote, Dermatology e Multiple Features, porém apresenta desempenho dentre os piores para diversas outras (Breast Cancer, Ionosphere e Landsat). *Average-Linkage* apresenta soluções superiores para as bases Landsat e Optical, porém soluções inferiores em bases como Ionosphere e Detect Malicious. *Ward's* quase nunca está entre as piores soluções, exceto possivelmente para Dermatology e Banknote sendo que seu desempenho se destaca para as bases Detect Malicious e Ionosphere.

Figura 32 – Erro de Transdução após cada consulta sobre a hierarquia *Average Linkage* (AL)

Com exceção das bases Banknote e Ionosphere, as soluções da estratégia M-BUHS* de certa forma seguem as soluções da hierarquia de grupos que fornece melhores soluções para o problema. Em especial, para a base Texture e Dermatology a partir de um certo número de consultas a estratégia inclusiva apresenta desempenho superior em relação ao uso de uma única hierarquia. Mesmo para as bases Banknote e Ionosphere, M-BUHS* se configura ainda como a segunda melhor estratégia. Em nenhuma das bases esta estratégia produz a pior solução, ao contrário do que se pode observar, por exemplo, para as bases Landsat com *Single-Linkage*,

Figura 33 – Erros de Transdução para diferentes hierarquias

Multiple Features e Optical com *Complete-Linkage*.

M-BUHS* se configura como uma estratégia recomendada nos cenários práticos onde o especialista não tem certeza sobre qual hierarquia deve ser entregue como entrada para o algoritmo.

A Seção a seguir incorpora na estratégia hierarquias simplificadas e de densidade para a exploração de representatividade.

6.4 Resultados com Hierarquias de Densidade e Simplificadas

Para explorar a representatividade com hierarquias de densidade e simplificadas utilizou-se a estratégia de múltiplas hierarquias com a estratégia discutida na Seção 5.3, denotada de "Mr-BUHS*". Foram entregues duas novas hierarquias, uma hierarquia simplificada obtida pelo HDBSCAN* com valor de $m_{pts} = m_{ClSize} = 4$ conforme sugerido nos trabalhos Ankerst *et al.* (1999), Campello *et al.* (2015) e por se tratar de um parâmetro não crítico, e a hierarquia completa correspondente que é a execução do *Single-Linkage* no espaço das distâncias de alcançabilidade mútua, conforme discutido na Seção 3.1.5.

Conforme a estratégia discutida na Seção 5.3, permitiu-se a consulta de observações que não estão em nós virtuais sobre a hierarquia simplificada. Com múltiplas hierarquias, quando a cobertura abrangente é encontrada para alguma das demais hierarquias fornecidas como entrada, não apenas sobre a perspectiva da hierarquia completa correspondente, os procedimentos para a escolha das observações de M-BUHS* são retomados, isto é, a hierarquia a ser escolhida é aquela com a menor estimativa de erro. Os procedimentos para a transdução dos rótulos são os mesmo quando executados sobre M-BUHS*.

A Figura 34 mostra o resultado quando a hierarquia de densidade e simplificada HDBSCAN* para $m_{pts} = m_{ClSize} = 4$ (HDBSCAN*(4)) e sua hierarquia completa correspondente *Single-Linkage* no espaço transformado (SLT(4)) são entregues a estratégia de M-BUHS e a exploração de representatividade é considerada. Os resultados são comparados aos resultados mostrados na Seção 6.3.

Exceto para a base Ionosphere, os resultados explorando representatividade possibilitou melhorias consideráveis sobre as bases Landsat, Multiple Features, Optical e Texture, bases em que o número de observações são os maiores.

6.5 Comparação com outros Algoritmos Ativos

Nesta seção estão os resultados do algoritmo com as melhorias propostas em comparação com outros algoritmos ativos para a tarefa de classificação indutiva.

Os algoritmos ativos usados para comparar com o algoritmos proposto neste trabalho foram: o conhecido *Margin Sampling* (TONG; KOLLER, 2002), algoritmo ativo que se enquadra dentro da abordagem de amostragem informativa, o *Hierarchical Subquery on Graph* (AODHA *et al.*, 2014), que busca agregar ambas abordagens de seleção informativa e representativa, o HS originalmente proposto por Dasgupta e Hsu (2008) e o HS* proposto por Tuia *et al.* (2012) e Muñoz-Marí *et al.* (2012), além do *baseline Random Sampling* que escolhe aleatoriamente qualquer observação para ser consultada.

Figura 34 – Erros de Transdução para diferentes hierarquias com hierarquia de densidade e simplificada

No trabalho de [Tong e Koller \(2002\)](#), um classificador SVM é treinado a partir de um conjunto de observações inicialmente rotuladas. Em seguida, o classificador é usado para selecionar as observações em regiões de alta incerteza (observações próximas à fronteira de decisão entre as classes). As observações com maior incerteza são selecionadas, rotuladas e acrescentadas ao conjunto de observações rotuladas. O classificador é retreinado e novas observações são escolhidas.

O algoritmo ativo de [Aodha et al. \(2014\)³](#) opera sobre um grafo onde os vértices são

³ Disponível em: <http://visual.cs.ucl.ac.uk/pubs/graphActiveLearning/>

dados pelas observações e às arestas possuem pesos que correspondem as similaridades entre os vértices. Semelhante ao HS, tal algoritmo é capaz de rotular todas as observações da base de dados, ao final do processo ativo. Entretanto, enquanto o HS faz esta operação sobre uma hierarquia de grupos, a transdução realizada pelo algoritmo ativo de Aodha *et al.* (2014) usa o algoritmo semi-supervisionado de Zhu *et al.* (2003).

Para a comparação, fez-se 30 execuções de um 10-fold *cross-validation* provendo para cada um dos métodos 3 observações aleatoriamente escolhidas e rotuladas como *seeds* para dar início ao procedimento ativo. Ao final aplica-se a transdução sobre 9 dos 10 folds (exceto para o *Random Sampling*). Em seguida, um classificador SVM (usando o *toolbox* LIBSVM (CHANG; LIN, 2011) com kernel RBF e parâmetros *default*) é treinado a partir do resultado de cada método e a acurácia é avaliada usando o *fold* restante configurado como teste.

A Figura 35 apresenta os resultados dos algoritmos. Para a estratégia M-BUHS* foram entregues como hierarquias de grupos o *Ward's*, *Complete-Linkage*, *Average-Linkage* e *Single-Linkage*, enquanto para o HS e HS* a hierarquia de grupos usada foi o *Ward's*, tal como no trabalho dos autores. Esta também é a hierarquia que apresentou os melhores resultados individuais nos experimentos da Seção 6.3.

Pode-se perceber que M-BUHS* está entre os 2 melhores desempenho em todas as bases de dados, melhorando (muitas vezes de forma significativa) os métodos HS, HS*, *Margin Sampling SVM* e *Random Sampling*. Em comparação com Aodha2014: somente para a base Optical Aodha2014 tem uma vantagem significativa sobre M-BUHS* e uma leve vantagem nas bases Banknote, Iris e Seeds e desempenho semelhante nas bases Brest Cancer e Dermatology. M-BUHS* tem uma vantagem significativa sobre Aodha2014 nas bases Landsat, Multiple Features, Texture e Wine. Perceba que quando M-BUHS* é superado por outro método, o mesmo se apresenta como a segunda melhor estratégia ativa com uma pequena diferença de desempenho, enquanto que Aodha2014 pode ter seu desempenho comprometido em relação a M-BUHS* inclusive com os demais métodos, tal como se pode observar para as bases Landsat e Texture e totalmente degradado como na base Multiple Features.

A Figura 36 apresenta os resultados de indução quando é explorado a representatividade com hierarquia de densidade e simplificada. Foi possível estabelecer melhorias significativa para as bases Landsat, Multiple Features e Optical. O Apêndice B apresenta uma tabela com a média de classificação para algumas das bases cujos resultados estão próximos.

No Apêndice C apresenta uma tabela com o tempo médio após cada consulta dos algoritmos HS e M-BUHS*. No aprendizado ativo como o usuário faz parte do processo de aprendizado a complexidade de tempo na busca por uma observação acaba se tornando um aspecto essencial. Na Seção 5.4 foi mostrado que a complexidade de M-BUHS* é de $\mathcal{O}(N)$ multiplicada por uma constante que se trata no número de hierarquias usadas e que quando executado em paralelo possui a mesma do algoritmo HS.

Figura 35 – Resultados de Classificação Indutiva

Figura 36 – Resultados de Classificação Indutiva com hierarquia de densidade e simplificada

CONCLUSÃO

Este trabalho de pesquisa partiu do algoritmo ativo HS, proposto por [Dasgupta e Hsu \(2008\)](#) e identificou suas principais limitações. Melhorias foram propostas e confirmadas pelos resultados, permitindo contornar tais limitações e apresentar soluções eficientes dentro do cenário ativo.

[Tuia et al. \(2012\)](#) estendeu a estratégia de seleção ativa dos grupos do HS na sua forma original para selecionar também subgrupos e conseqüentemente as observações a serem consultadas. Nesta dissertação tal melhoria foi combinada com uma nova estratégia para a rotulação das observações denominada Transdução *Bottom-Up*. Os resultados estabelecidos pelas características de ambas estratégias combinadas permitiram aprimorar os resultados do algoritmo em comparação com aqueles obtidos pelo algoritmo na sua forma original e também quando cada uma das estratégias é considerada separadamente.

Uma das principais limitações do algoritmo HS está na sua entrada, que é dada por uma hierarquia de grupos. Tal hierarquia, pode ser obtida por algoritmos como *Ward's*, *Complete-Linkage* ou por algoritmo recentes baseados em densidades como o HDBSCAN*. Há cenários em que determinadas hierarquias relacionam de forma mais (ou menos) apropriada os grupos com as possíveis classes do problema, se configurando como uma entrada mais (ou menos) adequada para o algoritmo. Entretanto, identificar o tipo de hierarquia de grupos mais apropriada em um dado problema não é trivial.

O trabalho investigou uma forma de contornar este problema permitindo que o algoritmo trabalhe com mais de uma hierarquia de grupos ao mesmo tempo. Os resultados obtidos permitiram concluir que tal estratégia se caracteriza como uma solução eficiente e robusta em cenários práticos em geral quando não se sabe antecipadamente que tipo de hierarquia de grupos levaria aos melhores resultados.

Tais estratégias se demonstraram eficientes para todos os tipos de estruturas de grupos usadas. A exploração de representatividade usando hierarquias de densidade e simplificada

também se mostrou eficiente dentro do cenário ativo. A exploração de representatividade quando inicialmente considerada se mostra como uma alternativa eficiente para o desenvolvimento de uma estratégias de consulta. Este aspecto também é discutido em [Aodha et al. \(2014\)](#).

Os resultados do HS com todas as estratégias definidas neste trabalho em comparação com outros algoritmo ativos, como por exemplo o *Margin Sampling*, um dos principais algoritmos investigados dentro do aprendizado ativo, especialmente dentro da abordagem de informatividade, se mostram eficientes para a indução de classificadores dada poucas observações rotuladas. Também é possível perceber que os resultados do algoritmo proposto por este trabalho está acima do HS proposto por [Dasgupta e Hsu \(2008\)](#) e pela variante de [Muñoz-Marí et al. \(2012\)](#), [Tuia et al. \(2012\)](#).

Nos experimentos, *seeds* foram usadas para dar início ao processo ativo. Entretanto, vale ressaltar que o HS, bem como a versão estendida neste trabalho, podem ser aplicados sem qualquer informação inicial de rótulo. Perceba que um número maior de observações nas *seeds* possibilita que o algoritmo obtenha mais informações sobre o problema e consequentemente pode vir a dificultar a investigação das definições de conjuntos mínimos de rotulações promissoras. Nesse sentido, poucas observações rotuladas foram usadas. Possibilitando assim, resultados a partir de frações mínimas de rotulações promissoras em cenários onde quase não há informações prévias sobre o problema de classificação.

Como investigações futuras pode-se agregar as informações de escala para guiar por consultas, buscando explorar eficientemente representatividade e informatividade. As informações de densidade produzidas pelo HDBSCAN* podem ser úteis para essa finalidade. Princípios como a Minimização do Risco Empírico e de Limites de Chernoff podem ser estudados para estabelecer limites para o mínimo de observações a serem rotuladas dentro de um cenário onde o algoritmo parte de nenhuma observação rotulada. Trabalhos como o de [Carlsson e Măşmoli \(2010\)](#) discute questões de estabilidade em hierarquias de grupos e pode vir a ser investigadas para a decisão de que hierarquia de grupos pode ser usada pelo algoritmo.

REFERÊNCIAS

- ALPAYDIN, E. **Introduction to Machine Learning**. [S.l.]: MIT Press, 2004. Citado na página 23.
- ANKERST, M.; BREUNIG, M. M.; KRIEGEL, H.-P.; SANDER, J. Optics: ordering points to identify the clustering structure. In: **ACM Sigmod Record**. [S.l.: s.n.], 1999. Citado 3 vezes nas páginas 49, 65 e 80.
- AODHA, O. M.; CAMPBELL, N.; KAUTZ, J.; BROSTOW, G. Hierarchical subquery evaluation for active learning on a graph. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2014. Citado 5 vezes nas páginas 59, 80, 81, 82 e 86.
- BACHE, K.; LICHMAN, M. **UCI Machine Learning Repository**. 2013. Citado na página 73.
- BASU, S.; BANERJEE, A.; MOONEY, R. Semi-supervised clustering by seeding. In **Proceedings of 19th International Conference on Machine Learning**, 2002. Citado na página 26.
- BASU, S.; BANERJEE, A.; MOONEY, R. J. Active semi-supervision for pairwise constrained clustering. In: **Proceedings of the Fourth SIAM International Conference on Data Mining**. [S.l.: s.n.], 2004. Citado na página 31.
- BLUM, A.; CHAWLA, S. Learning from labeled and unlabeled data using graph mincuts. In: **International Conference on Machine Learning**. [S.l.: s.n.], 2001. p. 19–26. Citado na página 26.
- BLUM, A.; MITCHELL, Y. Combining labeled and unlabeled data with co-training. **Proceedings of the eleventh annual conference on Computational learning theory**, 1998. Citado na página 26.
- BOLAÑOS, M.; GAROLERA, M.; RADEVA, P. Active labeling application applied to food-related object recognition. In: **Proceedings of the 5th international workshop on Multimedia for cooking & eating activities**. [S.l.: s.n.], 2013. Citado na página 62.
- BOLTON, R. J.; HAND, D. J. Statistical fraud detection: A review. **Statistical science**, 2002. Citado na página 24.
- CAMPELLO, R.; MOULAVI, D.; ZIMEK, A.; SANDER, J. Hierarchical density estimates for data clustering, visualization, and outlier detection. **ACM Transactions on Knowledge Discovery from Data**, 2015. Citado 4 vezes nas páginas 21, 37, 48 e 80.
- 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.]: Springer, 2013. Citado na página 48.
- 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**, 2013. Citado 2 vezes nas páginas 52 e 68.

- CARLSSON, G.; MÄŠMOLI, F. Characterization, stability and convergence of hierarchical clustering methods. **Journal of machine learning research**, p. 1425–1470, 2010. Citado na página 86.
- CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. **ACM Transactions on Intelligent Systems and Technology**, 2011. Citado na página 82.
- CHAPELLE, O.; SCHLKOPF, B.; ZIEN, A. **Semi-Supervised Learning**. [S.l.]: The MIT Press, 2010. Citado 3 vezes nas páginas 20, 25 e 26.
- CORTEZ, P.; PORTELINHA, M.; RODRIGUES, S.; CADAVEZ, V.; TEIXEIRA, A. Lamb meat quality assessment by support vector machines. **Neural Processing Letters**, v. 24, n. 1, p. 41–51, 2006. Citado na página 24.
- DASGUPTA, S. The two faces of active learning. In: **Algorithmic Learning Theory**. [S.l.: s.n.], 2009. Citado 2 vezes nas páginas 20 e 28.
- DASGUPTA, S.; HSU, D. Hierarchical sampling for active learning. In: **Proceedings of the 25th international conference on Machine learning**. [S.l.: s.n.], 2008. Citado 18 vezes nas páginas 20, 21, 22, 29, 32, 37, 53, 55, 57, 59, 63, 72, 74, 80, 85, 86, 95 e 96.
- DENG, L.; LI, X. Machine learning paradigms for speech recognition: An overview. **IEEE Transactions on Audio, Speech, and Language Processing**, v. 21, n. 5, p. 1060–1089, 2013. Citado na página 24.
- ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: **Knowledge Discovery and Data Mining**. [S.l.: s.n.], 1996. Citado 2 vezes nas páginas 33 e 48.
- FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. L. F. de. **Inteligência Artificial: uma Abordagem de Aprendizado de Máquina**. [S.l.]: LTC, 2011. Citado 5 vezes nas páginas 19, 23, 24, 26 e 30.
- FU, Y.; ZHU, X.; LI, B. A survey on instance selection for active learning. **Knowledge and information systems**, 2013. Citado 2 vezes nas páginas 20 e 30.
- FUREY, T. S. *et al.* Support vector machine classification and validation of cancer tissue samples using microarray expression data. **Bioinformatics**, v. 16, n. 10, p. 906–914, 2000. Citado na página 24.
- GAN, G.; MA, C.; WU, J. **Data clustering: theory, algorithms, and applications**. [S.l.]: Siam, 2007. Citado na página 36.
- HOLUB, A.; PERONA, P.; BURL, M. C. Entropy-based active learning for object recognition. **IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops**, 2008. Citado na página 28.
- HU, R.; NAMEE, B. M.; DELANY, S. J. Off to a good start: Using clustering to select the initial training set in active learning. In: **Twenty-Third International FLAIRS Conference**. [S.l.: s.n.], 2010. Citado na página 32.
- HUANG, S.-J.; JIN, R.; ZHOU, Z.-H. Active learning by querying informative and representative examples. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2010. Citado 6 vezes nas páginas 20, 28, 29, 31, 32 e 59.

- JAIN, A. K.; DUBES, R. C. **Algorithms for Clustering Data**. [S.l.]: Prentice-Hall, Inc., 1988. Citado 7 vezes nas páginas 19, 21, 25, 35, 37, 38 e 46.
- KANG, J.; RYU, K. R.; KWON, H.-C. Using cluster-based sampling to select initial training set for active learning in text classification. In: **Advances in knowledge discovery and data mining**. [S.l.]: Springer, 2004. Citado na página 32.
- LEWIS, D. D.; GALE, W. A. A sequential algorithm for training text classifiers. In: **Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.: s.n.], 1994. Citado na página 28.
- LI, J.; SANDER, J.; CAMPELLO, R. J. G. B.; ZIMEK, A. Active learning strategies for semi-supervised dbscan. In: **Advances in Artificial Intelligence**. [S.l.]: Springer, 2014. Citado 3 vezes nas páginas 20, 28 e 31.
- LUXBURG, U. V.; SCHÖLKOPF, B. Statistical learning theory: models, concepts, and results. 2008. Citado 3 vezes nas páginas 19, 24 e 25.
- MALLAPRAGADA, P. K.; JIN, R.; JAIN, A. K. Active query selection for semi-supervised clustering. In: **IEEE. 19th International Conference on Pattern Recognition**. [S.l.], 2008. Citado na página 31.
- MCCALLUM, A.; NIGAM, K. Employing em and pool-based active learning for text classification. In: **International Conference on Machine Learning**. [S.l.: s.n.], 1998. Citado na página 25.
- MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw-Hill, Inc., 1997. Citado 3 vezes nas páginas 19, 23 e 24.
- MUÑOZ-MARÍ, J.; TUIA, D.; CAMPS-VALLS, G. Semisupervised classification of remote sensing images with active queries. **IEEE Transactions on Geoscience and Remote Sensing**, 2012. Citado 5 vezes nas páginas 61, 67, 74, 80 e 86.
- NGUYEN, H. T.; SMEULDERS, A. Active learning using pre-clustering. In: **Proceedings of the twenty-first international conference on Machine learning**. [S.l.: s.n.], 2004. Citado na página 32.
- ROSS, S. **Probabilidade: Um Curso Moderno com Aplicações**. [S.l.]: Bookman, 2009. Citado 2 vezes nas páginas 93 e 94.
- ROY, N.; MCCALLUM, A. Toward optimal active learning through monte carlo estimation of error reduction. **ICML, Williamstown**, 2001. Citado na página 28.
- SAHA, T. K.; HASAN, M. A.; BURGESS, C.; HABIB, M. A.; JOHNSON, J. Batch-mode active learning for technology-assisted review. In: **Big Data (Big Data), 2015 IEEE International Conference on**. [S.l.: s.n.], 2015. Citado na página 25.
- SANDER, J.; QIN, X.; LU, Z.; NIU, N.; KOVARSKY, A. Automatic extraction of clusters from hierarchical clustering representations. In: **Advances in knowledge discovery and data mining**. [S.l.]: Springer, 2003. Citado na página 65.
- SCHEFLER, W. C. **Statistics: Concepts and Applications**. [S.l.]: Benjamin-Cummings Publishing Co., Inc., 1988. Citado 2 vezes nas páginas 56 e 95.

- SETTLES, B. **Active Learning Literature Survey**. [S.l.], 2009. Citado 6 vezes nas páginas 20, 21, 25, 27, 28 e 30.
- SETTLES, B.; CRAVEN, M.; RAY, S. Multiple-instance active learning. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2008. Citado 3 vezes nas páginas 30, 31 e 59.
- SEUNG, S. H.; OPPER, M.; SOMPOLINSKY, H. Query by committee. In: **Proceedings of the fifth annual workshop on Computational learning theory**. [S.l.: s.n.], 1992. Citado na página 28.
- SOUSA, C. A. R. de; REZENDE, S. O.; BATISTA, G. E. Influence of graph construction on semi-supervised learning. In: **Machine Learning and Knowledge Discovery in Databases**. [S.l.]: Springer, 2013. p. 160–175. Citado na página 26.
- STERCKX, L.; DEMEESTER, T.; DELEU, J.; DEVELDER, C. Using active learning and semantic clustering for noise reduction in distant supervision. In: **Workshop on Automated Base Construction at NIPS 2014**. [S.l.: s.n.], 2014. Citado na página 62.
- TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. [S.l.]: Inc. Addison-Wesley Longman Publishing Co., 2005. Citado 5 vezes nas páginas 21, 33, 35, 37 e 71.
- TONG, S.; KOLLER, D. Support vector machine active learning with applications to text classification. **The Journal of Machine Learning Research**, 2002. Citado 6 vezes nas páginas 20, 25, 27, 29, 80 e 81.
- TUIA, D.; KANEVSKI, M.; MUÑOZ-MARÍ, J.; CAMPS-VALLS, G. Cluster-based active learning for compact image classification. In: **Geoscience and Remote Sensing Symposium**. [S.l.: s.n.], 2010. Citado na página 61.
- TUIA, D.; MUÑOZ-MARÍ, J.; CAMPS-VALLS, G. Remote sensing image segmentation by active queries. **Pattern Recognition**, 2012. Citado 7 vezes nas páginas 25, 61, 67, 74, 80, 85 e 86.
- WARD, J.; JOE, H. Hierarchical grouping to optimize an objective function. **Journal of the American statistical association**, 1963. Citado na página 38.
- WARMUTH, M. K. *et al.* Active learning with support vector machines in the drug discovery process. **Journal of Chemical Information and Computer Sciences**, ACS Publications, v. 43, n. 2, p. 667–673, 2003. Citado na página 25.
- XU, Z.; YU, K.; TRESP, V.; XU, X.; WANG, J. Representative sampling for text classification using support vector machines. **Advances in Information Retrieval**, 2003. Citado na página 32.
- YANG, Y.; SONG, E.; MA, G. Hierarchical exploration based active learning with support vector machine. In: **Computational Intelligence and Natural Computing Proceedings**. [S.l.: s.n.], 2010. Citado na página 61.
- YEUNG, K. Y.; RUZZO, W. L. Principal component analysis for clustering gene expression data. **Bioinformatics**, v. 17, n. 9, p. 763–774, 2001. Citado na página 24.
- YU, K.; BI, J.; TRESP, V. Active learning via transductive experimental design. In: **ACM International Conference on Machine Learning**. [S.l.], 2006. p. 1081–1088. Citado na página 31.

ZHANG, L. *et al.* Active learning based on locally linear reconstruction. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 2011. Citado na página 31.

ZHU, J.; WANG, H.; YAO, T.; TSOU, B. K. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In: **Proceedings of the 22nd International Conference on Computational Linguistics**. [S.l.: s.n.], 2008. Citado na página 32.

ZHU, X. **Semi-Supervised Learning Literature Survey**. [S.l.], 2005. Citado na página 26.

ZHU, X.; GHAHRAMANI, Z.; LAFFERTY, J. Semi-supervised learning using gaussian fields and harmonic functions. In: **ICML**. [S.l.: s.n.], 2003. Citado 2 vezes nas páginas 26 e 82.

ZHU, X.; GOLDBERG, A. **Introduction to Semi-supervised Learning**. [S.l.]: Morgan & Claypool, 2009. (Synthesis lectures on artificial intelligence and machine learning). Citado 2 vezes nas páginas 27 e 60.

INTERVALO DE CONFIANÇA PARA PROPORÇÃO

Seja N_i o número de observações de um grupo i ao longo da hierarquia e $N_{i,l}$ o número de observações do grupo i que tem rótulo de classe l . Se t observações são consultadas ao especialista pode-se tomar X a variável aleatória que corresponde ao número de observações consultadas e rotuladas com a classe l e assumir que X segue uma distribuição Hipergeométrica com parâmetros N_i , t e $N_{i,l}$, isto é, $X \sim Hip(N_i, t, N_{i,l})$.

Perceba que $p_{i,l} = \frac{N_{i,l}}{N_i}$ corresponde a proporção real e $p_{i,l}(t) = \frac{X}{t}$ trata-se da proporção relacionada ao número de observações consultadas e rotuladas com o rótulo l .

Definição 9. Esperança: Seja Y uma variável aleatória que assume valores discretos $\{y_1, y_2, \dots, y_n\}$ e seja $P(Y = y_i)$ a função de probabilidade $P(Y = y_i)$, a esperança de Y , denotada por $E[Y]$ é definida por:

$$E[Y] = \sum_{i=1}^n y_i P(Y = y_i) \quad (\text{A.1})$$

Definição 10. Variância: A variância de uma variável aleatória Y que assume valores discretos é definida por $E[(Y - E[Y])^2]$ e denotada por $VAR[Y]$.

Uma vez que $X \sim Hip(N_i, t, N_{i,l})$ tem-se que a probabilidade de X assumir um valor v e o valor esperado $E[X]$ são dados por (ROSS, 2009):

$$P(X = v) = \frac{\binom{N_{i,l}}{v} \binom{N_i - N_{i,l}}{t - v}}{\binom{N_i}{t}} \quad (\text{A.2})$$

$$E[X] = \frac{tN_{i,t}}{N_i} \quad (\text{A.3})$$

Da distribuição de probabilidade dada pela Equação A.2 e da Definição 9 obtém-se que (ROSS, 2009):

$$E[X] = \sum_{v=0}^t vP(X=v) \quad (\text{A.4})$$

$$\begin{aligned} E[X(X-1)] &= \sum_{v=1}^t v(v-1)P(X=v) \\ E[X(X-1)] &= \sum_{v=1}^t v(v-1) \frac{\binom{N_{i,l}}{v} \binom{N_i-N_{i,l}}{t-v}}{\binom{N_i}{t}} \\ E[X(X-1)] &= \sum_{v=1}^t \frac{v(v-1) \frac{N_{i,l}!}{v!(N_{i,l}-v)!} \binom{N_i-N_{i,l}}{t-v}}{\frac{N_i!}{t!(N_i-t)!}} \\ E[X(X-1)] &= \sum_{v=1}^t \frac{(v-1) \frac{N_{i,l}(N_{i,l}-1)!}{(v-1)!(N_{i,l}-v)!} \binom{N_i-N_{i,l}}{t-v}}{\frac{N_i(N_i-1)!}{t(t-1)!(N_i-t)!}} \\ E[X(X-1)] &= \sum_{v=1}^t (v-1) \frac{N_{i,l}(N_{i,l}-1)!}{(v-1)!(N_{i,l}-v)!} \binom{N_i-N_{i,l}}{t-v} \frac{t(t-1)!(N_i-t)!}{N_i(N_i-1)!} \\ E[X(X-1)] &= \frac{tN_{i,l}}{N_i} \sum_{v=1}^t (v-1) \frac{\binom{N_{i,l}-1}{v-1} \binom{N_i-N_{i,l}}{t-v}}{\binom{N_i-1}{t-1}} \\ E[X(X-1)] &= \frac{tN_{i,l}}{N_i} \sum_{v=1}^t \frac{(v-1) \frac{\binom{N_{i,l}-1}{v-1} \binom{N_i-N_{i,l}}{t-v}}{\binom{N_i-1}{t-1}}}{\frac{(t-1)!(N_i-t)!}{(N_i-1)!}} \\ E[X(X-1)] &= \frac{tN_{i,l}}{N_i} \sum_{v=1}^t (v-1) \frac{\binom{N_{i,l}-1}{v-1} \binom{N_i-N_{i,l}}{t-v}}{\binom{N_i-1}{t-1}} \\ E[X(X-1)] &= \frac{tN_{i,l}}{N_i} E[X-1] \\ E[X(X-1)] &= \frac{tN_{i,l}}{N_i} \frac{(t-1)(N_{i,l}-1)}{N_i-1} \end{aligned} \quad (\text{A.5})$$

Dada a relação da Equação A.5 e pela Definição 10 pode-se obter o valor para a variância de $p_{i,l}(t) = \frac{X}{t}$:

$$\text{VAR}[X] = E[(X - E[X])^2] \quad (\text{A.6})$$

$$\begin{aligned}
\text{VAR}[p_{i,l}(t)] &= \text{VAR}\left[\frac{X}{t}\right] \\
\text{VAR}[p_{i,l}(t)] &= E\left[\left(\frac{X}{t} - E\left[\frac{X}{t}\right]\right)^2\right] \\
\text{VAR}[p_{i,l}(t)] &= E\left[\frac{X^2}{t^2} - 2\frac{X}{t}E\left[\frac{X}{t}\right] + E\left[\frac{X}{t}\right]^2\right] \\
\text{VAR}[p_{i,l}(t)] &= E\left[\frac{X^2}{t^2}\right] - E\left[2\frac{X}{t}E\left[\frac{X}{t}\right]\right] + E\left[E\left[\frac{X}{t}\right]^2\right] \\
\text{VAR}[p_{i,l}(t)] &= \frac{1}{t^2}E[X^2] - \frac{2}{t}E[X]E\left[\frac{X}{t}\right] + E\left[E\left[\frac{X}{t}\right]^2\right] \\
\text{VAR}[p_{i,l}(t)] &= \frac{1}{t^2}E[X^2] - \frac{2}{t^2}E[X]E[X] + \frac{1}{t^2}E[X]E[X] \\
\text{VAR}[p_{i,l}(t)] &= \frac{1}{t^2}(E[X^2] - E^2[X]) \\
\text{VAR}[p_{i,l}(t)] &= \frac{1}{t^2}(E[X^2] - E[X] + E[X] - E^2[X]) \\
\text{VAR}[p_{i,l}(t)] &= \frac{1}{t^2}(E[X(X-1)] + E[X] - E^2[X]) \\
\text{VAR}[p_{i,l}(t)] &= \frac{1}{t^2}\left(\frac{tN_{i,l}(N_{i,l}-1)(t-1)}{N_i} + \frac{tN_{i,l}}{N_i} - \frac{t^2N_{i,l}^2}{N_i^2}\right) \\
\text{VAR}[p_{i,l}(t)] &= \frac{N_{i,l}}{tN_i}\left(\frac{(N_i(tN_{i,l}-N_{i,l}-t+1)+N_i^2-N_i-tN_{i,l}(N_i-1))}{N_i(N_i-1)}\right) \\
\text{VAR}[p_{i,l}(t)] &= \frac{N_{i,l}}{tN_i}\left(\frac{(tN_iN_{i,l}-N_iN_{i,l}-tN_i+N_i+N_i^2-N_i-tN_iN_{i,l}+t)}{N_i(N_i-1)}\right) \\
\text{VAR}[p_{i,l}(t)] &= \frac{N_{i,l}}{tN_i}\left(\frac{-N_iN_{i,l}-tN_i+N_i^2+t}{N_i(N_i-1)}\right) \\
\text{VAR}[p_{i,l}(t)] &= \frac{N_{i,l}}{tN_i}\left(\frac{(N_i-N_{i,l})(N_i-t)}{N_i(N_i-1)}\right) \\
\text{VAR}[p_{i,l}(t)] &= \left(\frac{N_{i,l}}{tN_i}\right)\left(\frac{(N_i-N_{i,l})}{N_i}\right)\left(\frac{(N_i-t)}{N_i(N_i-1)}\right) \\
\text{VAR}[p_{i,l}(t)] &= \frac{p_{i,l}(1-p_{i,l})}{t}\frac{(N_i-t)}{(N_i-1)}
\end{aligned} \tag{A.7}$$

Na Equação A.7 o valor de $p_{i,l}$ não é conhecido. Entretanto, na prática admite-se a igualdade $p_{i,l} = p_{i,l}(t)$ (SCHEFLER, 1988). Assim, a Equação A.7, pode ser reescrita como:

$$\text{VAR}(p_{i,l}(t)) = \frac{p_{i,l}(t)(1-p_{i,l}(t))}{t} \frac{(N_i-t)}{(N_i-1)} \tag{A.8}$$

Dado $p_{i,l}(t)$ e sua variância $\text{VAR}(p_{i,l}(t))$ uma estimação intervalar para $p_{i,l}$ pode ser dada por:

$$[p_{i,l}(t) - \text{VAR}(p_{i,l}(t)), p_{i,l}(t) + \text{VAR}(p_{i,l}(t))] \tag{A.9}$$

Para poucas observações consultadas, especialmente para grupos capazes de alinhar as classe do problema, as observações consultadas poderão ter os mesmos rótulos informados. Neste caso, o valor para $p_{i,l}(t)$ será igual a 1 para o rótulo informado e 0 para os demais rótulos de classe, resultando em um valor de $\text{VAR}(p_{i,l}(t))$ igual a zero. Nesse caso, a estimação intervalar para $p_{i,l}$ resultaria em uma estimação pontual $[p_{i,l}(t) - \text{VAR}(p_{i,l}(t)), p_{i,l}(t) + \text{VAR}(p_{i,l}(t))] = [p_{i,l}(t), p_{i,l}(t)] = p_{i,l}(t)$.

Para evitar este tipo de estimação, em Dasgupta e Hsu (2008) o temo $\delta(t) = (N_i - N_i(t))/N_i(t)N_i$ é incluído obtendo o intervalo:

$$[p_{i,l}(t) - (\delta(t) + \text{VAR}(p_{i,l}(t))), p_{i,l}(t) + (\delta(t) + \text{VAR}(p_{i,l}(t)))] \tag{A.10}$$

Para evitar valores negativos e maiores que 1 no intervalo (A.10), define-se portanto o intervalo para a proporção $p_{i,l}$ usado por Dasgupta e Hsu (2008):

$$[\max(p_{i,l}(t) - \Delta(t), 0), \min(p_{i,l}(t) + \Delta(t), 1)] \quad (\text{A.11})$$

em que $\Delta(t) = (\delta(t) + \text{VAR}(p_{i,l}(t)))$.

TABELA DE CLASSIFICAÇÃO INDUTIVA

Tabela 4 – Média para Classificação Indutiva

Base	Algoritmo	Consulta				
		20	40	60	80	100
Banknote	HS	0.7735	0.9405	0.9652	0.9713	0.9752
	HS*	0.7695	0.9524	0.9725	0.9820	0.9818
	Margin	0.7907	0.9050	0.9650	0.9838	0.9898
	Random	0.7237	0.8695	0.8902	0.9069	0.9233
	Aodha2014	0.9826	0.9819	0.9814	0.9817	0.9815
	M-BUHS*	0.9689	0.9808	0.9839	0.9820	0.9832
Breast Cancer	HS	0.9699	0.9708	0.9718	0.9708	0.9718
	HS*	0.9728	0.9703	0.9708	0.9713	0.9703
	Margin	0.9571	0.9713	0.9713	0.9713	0.9704
	Random	0.9578	0.9632	0.9649	0.9648	0.9659
	Aodha2014	0.9722	0.9688	0.9706	0.9707	0.9707
	M-BUHS*	0.9625	0.9694	0.9708	0.9723	0.9743
Iris	HS	0.8422	0.9311	0.9333	0.9400	0.9378
	HS*	0.8889	0.9333	0.9400	0.9400	0.9378
	Margin	0.7933	0.8973	0.9133	0.9213	0.9320
	Random	0.7473	0.8747	0.9153	0.9253	0.9333
	Aodha2014	0.9378	0.9400	0.9400	0.9400	0.9400
	M-BUHS*	0.9244	0.9378	0.9400	0.9400	0.9400

Tabela 5 – Média para Classificação Indutiva

Base	Algoritmo	Consulta				
		20	40	60	80	100
Seeds	HS	0.8175	0.8524	0.8619	0.8778	0.8810
	HS*	0.8222	0.8746	0.8825	0.8857	0.8921
	Margin	0.7019	0.8329	0.8767	0.8805	0.8814
	Random	0.7067	0.8614	0.8890	0.8943	0.8924
	Aodha2014	0.8709	0.8857	0.8957	0.9043	0.9048
	M-BUHS*	0.8746	0.8825	0.8905	0.8905	0.9000
Detect Malicious	HS	0.9946	0.9946	0.9946	0.9946	0.9946
	HS*	0.9946	0.9946	0.9946	0.9946	0.9946
	Margin	0.9367	0.9873	0.9946	0.9946	0.9946
	Random	0.9798	0.9930	0.9914	0.9916	0.9927
	Aodha2014	0.9946	0.9946	0.9946	0.9946	0.9946
	M-BUHS*	0.9946	0.9946	0.9946	0.9946	0.9946

TEMPO MÉDIO DE EXECUÇÃO

Tabela 6 – Tempo médio de execução após cada consulta. Executado sobre: Desktop 3.2GHz, Intel Core i3, 8GB RAM, Windows 7 e Java 8.

Base	HS	M-BUHS*
Banknote	9.5409e-044.0031e-04	4.5293e-033.8984e-03
Breast Cancer	2.4810e-041.1833e-04	1.8013e-030.5168e-03
Dermatology	1.2939e-041.0058e-04	1.3069e-031.0385e-03
Detect Malicious	1.9789e-041.4786e-04	9.3779e-040.8736e-04
Ionosphere	9.9800e-058.9221e-05	1.0229e-031.8627e-03
Iris	4.3600e-056.2125e-05	4.9150e-042.7268e-04
Landsat	3.9876e-032.8277e-03	2.1647e-021.9565e-02
Multiple	1.3931e-031.4486e-03	1.2482e-028.9596e-03
Optical	5.8610e-034.8595e-03	3.7697e-023.2969e-02
Seeds	7.0400e-053.0010e-05	7.0240e-046.4859e-04
Texture	6.4932e-035.0249e-03	4.6464e-024.6843e-02
Wine	4.5400e-053.4226e-05	7.7040e-047.1058e-04