Universidade Federal Fluminense
Instituto de Ciência e Tecnologia
Bacharelado em Ciência da Computação

Rodrigo Magalhães Rodovalho

Proposta de um método de aprendizado multirrótulo utilizando aprendizado não supervisionado.

Rio das Ostras-RJ
2017

RODRIGO MAGALHÃES RODOVALHO

PROPOSTA DE UM MÉTODO DE APRENDIZADO MULTIRRÓTULO UTILIZANDO APRENDIZADO NÃO SUPERVISIONADO.

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Bacharel. Área de Concentração: Mineração de Dados e Inteligencia Artificial.

Orientador: Prof. Dra. FLÁVIA CRISTINA BERNARDINI

Rio das Ostras-RJ

2017

RODRIGO MAGALHÃES RODOVALHO

PROPOSTA DE UM MÉTODO DE APRENDIZADO MULTIRRÓTULO UTILIZANDO APRENDIZADO NÃO SUPERVISIONADO

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Bacharel. Área de Concentração: Mineração de Dados e Inteligencia Artificial.

Aprovada em JANEIRO de 2017.

BANCA EXAMINADORA

Prof. Dra. FLÁVIA CRISTINA BERNARDINI - Orientador
UFF

Prof. Dr. EDWIN BENITO MITACC MEZA
UFF

Prof. Dra. LEILA WEITZEL COELHO DA SILVA
UFF

Rio das Ostras-RJ
2017

Dedico este trabalho à minha família,
em especial ao meu filho
Heitor Magalhães Alves Rodovalho.

Agradecimentos

Primeiramente a Deus, pois sem Ele nada disso seria possível.

Aos meus pais **José Maurício Rodovalho Gomes** e **Cláudia Márcia Magalhães da Silva Gomes** que me apoiaram e incentivaram em todos os momentos e situações na vida, assim como na minha formação acadêmica.

À minha companheira, amiga e esposa **Yasmin Rayanne Alves Rodovalho** pelo total apoio, motivação e companheirismo.

A todos os professores que se dedicaram e empenharam em dar seu melhor para que eu pudesse chegar ao final do curso. Em especial agradeço a minha professora e orientadora **Flávia Cristina Bernardini** pelos ensinamentos, paciência e paixão pelo que faz.

A todos que me apoiaram e que de maneira direta ou indireta contribuíram para que eu pudesse chegar até aqui.

Lista de Figuras

Lista de Tabelas

Sumário

Resumo

O aprendizado multirrótulo tem por objetivo a construção de classificadores que rotulam, com mais de um rótulo, casos ainda não rotulados, como é o caso de diagnóstico de falhas em um equipamento, ou gêneros musicais de uma música. Uma questão importante do aprendizado multirrótulo está relacionado à grande quantidade de exemplos (casos de aprendizado) disponíveis, sendo cada exemplo associado a poucos rótulos, e esses, por sua vez, são oriundos de um grande conjunto de rótulos possíveis. O objetivo deste trabalho é propor um método de aprendizado multirrótulo baseado em aprendizado não-supervisionado hierárquico como uma técnica de divisão e conquista do problema. Para atingir esse objetivo, são utilizadas as ferramentas Mulan e Weka para apoiar o desenvolvimento do método a ser proposto, e são utilizadas bases de dados naturais para avaliar o desempenho do método a ser proposto.

Palavras-chave: Mineração de Dados. Aprendizado de Máquina. Aprendizado Multirrótulo.

Abstract

Multi-label learning aims to build classifiers that label, with more than one label, cases not labeled yet, such a failure diagnosis in devices, or genres of a music. An important issue of multi-label learning is related to the large number of examples (cases of learning) available, each instance associated with a few labels, and these, in turn, are derived from a large set of possible labels. The objective of this paper is to propose a multi-label learning based on a hierarchical unsupervised learning method, a division and conquer technical of the problem. To achieve this goal, Mulan and Weka tools are used to support the development of the proposed method, and natural databases are used to evaluate the performance of the proposed method.

Keywords: Data Mining. Machine Learning. Multi-label Learning.

Capítulo 1

Introdução

Com o avanço da tecnologia, o crescimento de recursos computacionais de armazenamento de dados permitiu o acúmulo de grandes quantidades de dados, o que motivou um interesse em maneiras de extrair de informações relevantes desses dados. Consequentemente, houve uma expansão na área da mineração de dados e do aprendizado de máquina. Técnicas para análises de dados foram desenvolvidas e possibilitaram o advento de sistemas computacionais capazes de absorver novas habilidades, novos conhecimentos e novos jeitos de organizar o conhecimento [Mitchell and Michell, 1997].

O Aprendizado de Máquina é muito utilizado em tarefas de Descoberta de Conhecimento de Bases de Dados (Knowlegde Discovery from Databases — KDD) [Faceli et al., 2011], cujo objetivo é extrair informações relevantes e automatizar o processo de análise de dados [Fayyad et al., 1996]. Uma tarefa comum em aprendizado de máquina é relacionada ao aprendizado supervisionado, no qual, dado um conjunto de objetos de treinamento rotulados, o objetivo é aprender uma função que rotule corretamente novos exemplos com os rótulos observados no conjunto de treinamento, como por exemplo objetos que representam indivíduos cujos rótulos são positivos ou negativos quanto a uma determinada doença. Dentre esses problemas de aprendizado supervisionado, existem aqueles nos quais cada objeto de treinamento é rotulado com mais de um rótulo, denominado aprendizado multirrótulo [Tsoumakas et al., 2009]. Aprendizado multirrótulo é uma linha de pesquisa da subárea de aprendizado de máquina, que objetiva a construção de classificadores que rotulam, com mais de um rótulo, casos ainda não rotulados, como é o caso de diagnóstico de falhas em um equipamento [Bernardini et al., 2009], ou gêneros musicais de uma música [Bernardini et al., 2013].

Uma questão importante do aprendizado multirrótulo está relacionada à grande quantidade de exemplos (casos de aprendizado) disponíveis com poucos rótulos associados, em geral, oriundo de um grande conjunto de rótulos possíveis. Há alguns trabalhos na literatura que objetivam tratar essa questão, que são descritos a seguir.

Diversos trabalhos na literatura apresentam métodos distintos para tratamento de problemas multirrótulo com múltiplos rótulos [Agrawal et al., 2013, Tang et al., 2009, Tsoumakas et al., 2008]. Duas características importantes que definem as bases de dados multirrótulos são a cardinalidade e a densidade, sendo a primeira relativa ao número médio de rótulos por exemplo, e a segunda relativa ao número médio

de rótulos por exemplo ponderado pelo número total de rótulos. Em [Rodovalho and Bernardini, 2013] estudos realizados usando conjuntos de dados artificiais multirrótulo, sugerem que a diminuição da densidade permite melhores resultados de classificação multirrótulo. Assim, o agrupamento de rótulos semelhantes pode permitir que melhores resultados sejam obtidos e, ainda, que conjuntos de dados com muitos rótulos possam ser processados por um método que explore essa característica. O aprendizado de máquina não-supervisionado tem por objetivo agrupar exemplos semelhantes, a partir de alguma métrica de similaridade ou de distância. Agrupar objetos semelhantes permite que classificadores multirrótulo possam ser construídos para cada agrupamento de objetos construído.

1.1 Objetivo e Metodologia

O objetivo deste trabalho é propor um novo método de aprendizado multirrótulo para tratar o problema de muitos rótulos.

Neste trabalho, foi utilizado o algoritmo de aprendizado não-supervisionado hierárquico baseado na abordagem de dividir para conquistar por permitir que seja selecionado em que nível se deseja selecionar os agrupamentos a serem considerados. A vantagem do algoritmo de aprendizado não-supervisionado é que os rótulos podem ser agrupados a outros rótulos semelhantes. É utilizado como conceito de semelhança a maneira que são rotulados os exemplos da base de dados. Para a avaliação do método proposto, inicialmente o método foi implementado foi utilizada a biblioteca Mulan [Tsoumakas et al., 2011b], que utiliza como base a biblioteca Weka [Holmes et al., 1994], em linguagem Java. Após, foram selecionados conjuntos de dados multirrótulo já utilizados por outros problemas na literatura, e o método proposto foi comparado a um outro método de aprendizado multirrótulo que trata de problemas com muitos rótulos, e a outros métodos clássicos para tratamento de problemas multirrótulo, todos apresentados no decorrer deste trabalho.

1.2 Organização da Monografia

Este trabalho está divido como segue:

No Capítulo 2, são apresentados os conceitos de aprendizado de máquina. Assim como suas divisões e abordagens.

No Capítulo 3, é apresentado uma proposta de um método de aprendizado multirrótulo baseado em aprendizado de máquina não-supervisionado hierárquico.

No Capítulo 4, são mostrados os resultados obtidos pela execução do método proposto e comparações de resultados com algoritmos conhecidos da literatura.

No Capítulo 5, é apresentado uma análise final dos resultados obtidos e sugestões para trabalhos futuros.

Capítulo 2

Aprendizado de Máquina

Um dos desafios da Inteligência Artificial é construir sistemas que façam a máquina aprender conceitos e/ou se adaptar ao ambiente. A subárea da Inteligência Artificial relacionada a esse tipo de problema é denominada aprendizado de máquina.[1] O aprendizado indutivo, um tipo de aprendizado, tem por objetivo inferir padrões ou conhecimento, representados por uma hipótese, ou modelo, a partir de exemplos fornecidos. O aprendizado indutivo pode ser dividido em dois tipos: aprendizado supervisionado e não supervisionado.

No aprendizado supervisionado, cada exemplo fornecido ao indutor possui um rótulo, oferecido por um supervisor especialista do domínio de onde os dados são provenientes. O objetivo do aprendizado supervisionado é construir uma hipótese, ou modelo, que rotule novos exemplos ainda não rotulados. No problema padrão de aprendizado supervisionado, a entrada do algoritmo consiste de um conjunto de objetos rotulados, ou exemplos, S, com N objetos $T_i, i = 1, ..., N$, escolhidos de um domínio X com uma distribuição $\mathcal{D}$ fixa, desconhecida e arbitrária, da forma $\{(\mathbf{x_i}, y_i), ..., (\mathbf{x_N}, y_N)\}$ para alguma função desconhecida y = f($\mathbf{x}$). Os $\mathbf{x_i}$ são tipicamente vetores da forma $(x_{i1}, x_{i2}, ..., x_{iM})$ com valores discretos ou numéricos, e X_{ij} refere-se ao valor do atributo j, denominado X_j, do exemplo T_i. Os valores y_i referem-se ao valor do atributo Y, frequentemente denominado classe. Os valores de y são tipicamente pertencentes a um conjunto discreto de classes $L = \{l_1, ..., l_R\}$, quando se trata de classificação, ou ao conjunto de números reais em caso de regressão.

Um algoritmo comumente utilizado para aprendizado de máquina é o C4.5 para construção de árvores de decisão, e também é utilizado neste trabalho. C4.5 é um algoritmo bastante conhecido na literatura que pode ser usado para realizar classificações, que se aplica ao aprendizado supervisionado monorrótulo. O C4.5 utiliza um conjunto de entradas rotuladas para criar uma árvore de decisão [Gholap, 2012, Quinlan, 1993]. Os nós internos de uma árvore de decisão são relativos a diferentes atributos do domínio e os ramos entre os nós são possíveis valores que os atributos podem ter nas amostras observadas, e por fim os nós terminais, ou nós folha, indicam a classificação y. A vantagem desse algoritmo é que não há necessidade de definição de parâmetros, e ainda apresenta em geral bons resultados em diversos domínios.

[1] Neste trabalho, para efeitos de simplificação, quando o termo aprendizado for utilizado, está se referindo ao aprendizado de máquina

Quando um exemplo é associado mais de uma classe, o problema é denominado multirrótulo. Daí, classificadores que rotulam exemplos com mais de um rótulo são necessários. Para induzir modelos com essa característica, deve ser utilizado o aprendizado de máquina multirrótulo. Para um algoritmo de aprendizado multirrótulo, a entrada consiste de um conjunto de exemplos S, com N objetos rotulados, ou exemplos, $T_i, i = 1, ..., N$, escolhidos de um domínio X com uma distribuição $\mathcal{D}$ fixa, desconhecida e arbitrária, da forma $\{(\mathbf{x_i}, Y_i), ..., (\mathbf{x_N}, Y_N)\}$. $L = \{\lambda_1, ..., \lambda_q\}$ é o conjunto de rótulos possíveis do domínio D, e $Y_i \subseteq L$, ou seja, Y_i é o conjunto de rótulos associado ao *i-ésimo* objeto. A saída de um algoritmo de aprendizado supervisionado de modelos multirrótulos é um classificador $\mathbf{h}$, que classifica um exemplo $\mathbf{x_i}$ com o conjunto $Z_i = \mathbf{h}(\mathbf{x_i})$, o qual é o conjunto de classes preditas por $\mathbf{h}(\mathbf{x})$ para o exemplo $\mathbf{x_i}$. Há diversos métodos propostos na literatura para indução de modelos multirrótulo [Tsoumakas et al., 2010, Calembo et al., 2011, Alvares-Cherman et al., 2012, da Gama et al., 2013].

No aprendizado não supervisionado, os exemplos não são rotulados, e o objetivo desse aprendizado é realizar descoberta de conhecimento por investigação. Nesse caso, analogamente ao aprendizado supervisionado, o conjunto de objetos não-rotulados S é composto por N objetos $T_i, i = 1, ..., N$, escolhidos de um domínio X com uma distribuição $\mathcal{D}$ fixa, desconhecida e arbitrária, da forma $\{\mathbf{x_1}, ..., \mathbf{x_N}\}$, onde $\mathbf{x_i}$ são tipicamente vetores da forma $(x_{i1}, x_{i2}, ..., x_{iM})$ com valores discretos ou numéricos, e x_{ij} refere-se ao valor do atributo j, denominado $\mathbf{X}_j$, do exemplo T_i.

Métodos de agrupamento podem ser organizados em dois tipos [Rokach, 2009]: agrupamento particional e agrupamento hierárquico. No primeiro tipo, um conjunto de exemplos é dividido em uma partição simples de k grupos, enquanto que no segundo, exemplos são organizados em grupos e subgrupos a partir de uma sequência de partições aninhadas [Marcacini, 2014].

Na seção a seguir — Seção 2.1 —, é descrito o aprendizado não-supervisionado hierárquico, pois um algoritmo pertencente a essa categoria de algoritmos é utilizado neste trabalho; e na Seção 2.2, é descrito com detalhes, os métodos de aprendizado multirrótulo.

2.1 Aprendizado de Máquina Não-Supervisionado Hierárquico

O aprendizado não supervisionado pode ser definido como um modo de encontrar padrões nos dados sem que haja a interferência de um supervisor externo. O aprendizado não supervisionado, engloba diferentes tarefas não supervisionadas, tais como a sumarização, a clusterização e a redução de dimensionalidade, sendo as duas últimas consideradas pilares do aprendizado não supervisionado [Ghahramani, 2003].

A redução de dimensionalidade consiste na redução do número de atributos. O objetivo dessa técnica é obter um conjunto de atributos a partir da redução do espaço de busca pela solução. Portanto o conjunto obtido possui menor dimensionalidade em relação ao conjunto original, entretanto a qualidade da solução final deve ser mantida [Nogueira, 2009]. Embora a técnica de redução de dimensionalidade seja amplamente utilizada, nesse trabalho o nosso enfoque se concentra na técnica de clusterização.

Clusterização, ou análise de cluster, consiste na busca de agrupar exemplos de dados baseando-se na similaridade ou distância entre eles [Doni, 2004]. O objetivo é conseguir determinar os grupos, a fim de obter homogeneidade dentro dos grupos e heterogeneidade entre eles. Devido a grande capacidade de

armazenamento de dados, esse grande volume de dados podem gerar muitas combinações de grupos, o que dificulta a sua análise, devido ao grande custo computacional.

A fim de resolver esse impasse, foram desenvolvidas várias técnicas que auxiliam na formação dos grupos. Algumas características são vitais a essas técnicas, para que elas consigam um resultado satisfatório, como por exemplo, ser capaz de lidar com dados com alta dimensionalidade, habilidade para lidar com diferentes tipos de dados, entre outros [Zaïane et al., 2002]. Ainda nessa seção, são apresentadas técnicas e métricas que auxiliam nessa análise - Técnicas de agrupamentos hierárquicos e medidas de distâncias respectivamente.

Dois fatores são fundamentais no processo de agrupamento: (1) uma medida de distância e (2) uma estratégia de agrupamento. A maneira com que os grupos são obtidos está diretamente relacionada à medida de distância escolhida, que determinam como a distância entre dois elementos é calculada. Essa escolha depende dos tipos de atributos que representam os exemplos [Marcacini, 2014]. Para indução de modelos de agrupamento a partir dos dados, são utilizados métodos e algoritmos, que correspondem às estratégias de agrupamento [Hastie et al., 2009].

2.1.1 Medidas de distância

Medidas de distância são necessárias quando se quer obter de um conjunto de dados complexo, uma estrutura simples de grupos [Kasznar et al., 2009]. Essas medidas são calculadas entre os elementos a serem agrupados. A partir da utilização dessas medidas podem ser definidas as relações intra-cluster, ou seja, as relações definidas, para que elementos pertençam a um mesmo cluster. Tais medidas (ou métricas) são descritas a seguir.

Distância Euclidiana

Distância euclidiana é considerada uma distância geométrica no espaço multidimensional. Considerando dois exemplos $\mathbf{x_1} = (x_{11}, x_{12}, ..., x_{1M})$ e $\mathbf{x_2} = (x_{21}, x_{22}, ..., x_{2M})$ ela pode ser calculada conforme definido na Equação 2.1, onde i é a dimensão dos vetores.

$$d_{x_1,x_2} = \sqrt{(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 + ... + (x_{1M} - x_{2M})^2} = \sqrt{\sum_{i=1}^{M}(x_{1i} - x_{2i})^2} \tag{2.1}$$

Distância Euclidiana Quadrática

Distância euclidiana quadrática é definida pela Equação 2.2:

$$d_{x_1,x_2} = (x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 + ... + (x_{1M} - x_{2M})^2 = \sum_{i=1}^{M}(x_{1i} - x_{2i})^2 \tag{2.2}$$

Distância Manhattan

Distância Manhattan calcula a distância que seria percorrida para chegar de um ponto de dados para o outro se um caminho do tipo grade for seguido. A distância Manhattan entre os dois itens é a soma das

diferenças dos seus componentes correspondentes, definida pela Equação 2.3.

$$d_{x_1,x_2} = (|x_{11} - x_{21}|) + (|x_{12} - x_{22}|) + ... + (|x_{1M} - x_{2M}|) = \sum_{i=1}^{M}(|x_{1i} - x_{2i}|) \tag{2.3}$$

Distância Chebychev

Distância Chebychev é utilizada, na situação que se deseja diferenciar dois elementos, se houver apenas uma das dimensões diferentes. A distância de chebychev é definida pela Equação 2.4:

$$d_{x_1,x_2} = max(|x_{11} - x_{21}|) + (|x_{12} - x_{22}|) + ... + (|x_{1M} - x_{2M}|) \tag{2.4}$$

2.1.2 Métodos de Agrupamento Hierárquicos

Os métodos hierárquicos de agrupamento realizam uma série de sucessivos agrupamentos (clusters) ou sucessivas divisões de elementos, onde esses elementos são agregados ou desagregados [Doni, 2004]. A distância entre os clusters é usada como critério para formação dos mesmos [Carvalho et al., 2009]. Pode acontecer de um exemplo pertencer a mais de um grupo, ou até mesmo, ocorrer de cada exemplo possuir um grau de pertinência associado aos grupo. A ocorrência de alguma dessas situações é chamada de sobreposição [Marcacini, 2014].

A clusterização hierárquica pode ser feita utilizando duas abordagens: aglomerativa – iniciando com tantos clusters quantos objetos e então unindo-os em novos clusters – ou divisiva – iniciando com um cluster apenas e dividindo-o em novos clusters [Carvalho et al., 2009], ambas apresentadas a seguir.

Métodos aglomerativos

Os dados são inicialmente distribuídos de modo que cada exemplo represente um cluster e, então, esses clusters são recursivamente agrupados considerando alguma medida de similaridade ou distância, até que todos os exemplos pertençam a apenas um cluster [Berkhin, 2006]. Na Figura 2.1 é ilustrado esse processo, que é iniciado pelo nível inferior da árvore apresentada. Em clusterização hierárquica a árvore resultante do processo de clusterização é denominada dendograma [Faceli et al., 2011]. Essa abordagem utiliza uma estratégia *bottom-up* de agrupamento. Existem vários algoritmos baseados nessa abordagem. A diferença entre esses algoritmos é o critério utilizado para definir as distâncias entre grupos, ou seja, as distâncias inter-cluster [Doni, 2004].

Em geral, os algoritmos aglomerativos utilizam os passos de um algoritmo padrão exibido no Algoritmo 1. A diferença entre os algoritmos ocorre no passo 5, onde a função de distância é utilizada. A matriz de distância $D_{N\times N}$ é inicialmente definida pela distância entre cada par de exemplos $(\mathbf{x_i}, \mathbf{x_j})$, sendo x_i e $x_j \in S$, isto é, $i, j = 1, ..., N$. Na primeira iteração do laço iniciado no passo 2, os exemplos U e V são clusters contendo um único exemplo pertencentes ao conjunto S. A partir da segunda iteração U e V podem possuir mais de um elemento. Assim, no passo 3, a menor distância é obtida a partir de uma busca pelo exemplos U e V na matriz D. No passo 5, a atualização das distâncias na matriz D é feita de maneira que os exemplos U e V se tornam um único exemplo na matriz, a partir daí, é calculada

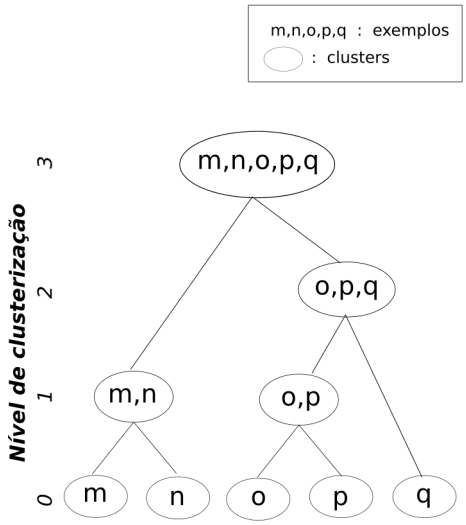

Figura 2.1: Exemplo de clusterização realizada por métodos aglomerativos.

a distância entre o exemplo d_{UV} que representa U e V e cada exemplo W restante da matriz, essa distância é denominada $d_{(UV)W}$ e pode ser calculada por seis diferentes técnicas de ligação, descritas a seguir.

Algoritmo 1: Algoritmo padrão utilizado por métodos aglomerativos.

Entrada: Conjunto de dados $S = \{\mathbf{x_i}, ..., \mathbf{x_N}\}$

Saída : Dendograma que representa o conjunto de grupos

1 Iniciar com **N** grupos, cada grupo contendo um exemplo x_i e uma matriz de distância $\mathbf{D}_{N \times N}$

2 **repeat**

3 Encontrar a menor distância $\mathbf{d_{UV}}$;

4 Atualizar a matriz **D**, removendo os elementos **U** e **V**;

5 Atualizar a matriz **D**, inserindo as novas distâncias do cluster formado pelos exemplos $\mathbf{U} \cup \mathbf{V}$;

6 **until *N-1, no qual todos exemplos estarão em um único grupo***;

Simple Linkage: Conhecido também como, método de ligação por vizinho mais próximo, esse método utiliza a distância de menor valor, definida pela Equação 2.5, onde d_{UV}, d_{UW} e d_{VW} são as distâncias entre os elementos UV, UW e VW, respectivamente.

$$d_{(UV)W} = min(d_{UW}, d_{VW}) \tag{2.5}$$

Segundo [ANDERBERG, 1973], algumas características dessa técnica são:

- Geralmente, grupos muito próximos podem não ser identificados;
- Possibilita a detecção de grupos que possuem formas não-elípticas;
- Apresenta pouca tolerância a ruído;

- Demonstram bons resultados utilizando distâncias euclidianas assim como outras distâncias; e
- Possui a tendência de formar longas cadeias (encadeamento).

Complete Linkage: Conhecido também como, método de ligação do vizinho mais distante, esse método utiliza a distância máxima, que é dada pela Equação 2.6:

$$d_{(UV)W} = max(d_{UW}, d_{VW}) \tag{2.6}$$

Average Linkage: Conhecido também como, método de ligação por média, esse método utiliza a Equação 2.7, onde N_U e N_V são os números de elementos no grupo U e V, respectivamente; d_{UW} e d_{VW} são as distâncias entre os elementos UW e VW, respectivamente.

$$d_{(UV)W} = \frac{(N_U.d_{UW} + N_V.d_{VW})}{N_U + N_V} \tag{2.7}$$

Centroid Linkage: Conhecido também como, método de ligação por centróide, esse método utiliza a Equação 2.8, onde N_U e N_V são os números de elementos no grupo U e V, respectivamente; d_{UW} e d_{VW} são as distâncias entre os elementos UW e VW, respectivamente.

$$d_{(UV)W} = \frac{(N_U.d_{UW} + N_V.d_{VW})}{N_U + N_V} - \frac{N_U.N_V.d_{UW}}{(N_U + N_V)^2} \tag{2.8}$$

Segundo [Doni, 2004], são algumas características dessa técnica:

- Algoritmo se mostra satisfatório em termos de resultados quando exposto à presença de ruídos;
- Passível de ocorrência do fenômeno da reversão, isto ocorre quando a distância entre os centróides é menor que a distância entre grupos já formados, consequentemente fará com que os novos grupos se formem ao um nível inferior aos grupos existentes; e
- Esse método não é muito utilizado, pois devido a ocorrência do fenômeno da reversão, o resultado é um dendograma confuso.

Median Linkage: Conhecido também como, método de ligação por mediana, esse método utiliza a seguinte expressão para o cálculo da distância:

$$d_{(UV)W} = \frac{d_{UV} + d_{VW}}{2} - \frac{d_{UV}}{4} \tag{2.9}$$

- onde d_{UW} e d_{WV} são as distâncias entre os elementos UW e VW, respectivamente.

Segundo [Doni, 2004], são essas algumas características desse método:

- Demonstra resultados satisfatórios quando os grupos são de tamanhos diferentes;
- Quando permutado os elementos na matriz de similaridade ou distância, pode apresentar resultado diferente; e

- Algoritmo se mostra satisfatório em termos de resultados quando exposto à presença de *outliers*.

Ward's Linkage: Método de ligação de Ward, nesse método a distância é calculada pela expressão:

$$d_{(UV)W} = \frac{((N_W + N_U).d_{UW} + (N_W + N_V).d_{VW} - N_W.d_{UV})}{N_W + N_U + N_V} \tag{2.10}$$

- onde N_U e N_W são os números de elementos no grupo U e V, respectivamente; d_{UW} e d_{VW} são as distâncias entre os elementos UW e VW, respectivamente.

Segundo [Doni, 2004], são essas algumas características desse método:

- Demonstra bons resultados utilizando distâncias euclidianas assim como outras distâncias;
- Se o número de elementos em cada grupo for praticamente igual, o método pode apresentar resultados insatisfatórios;
- Possui a tendência de combinar grupos com poucos elementos; e
- Sensível à presença de *outliers*.

Métodos divisivos

O processo inicia-se com apenas um agrupamento contendo todos os dados e segue dividindo-o recursivamente segunda alguma métrica até que alcance algum critério de parada, frequentemente o número de clusters desejados [Berkhin, 2004]. Na Figura 2.2 é ilustrado esse processo em que, ao contrário da abordagem aglomerativa, aqui o processo se inicia pelo nó raiz da árvore (nível 0 na Figura 2.2).

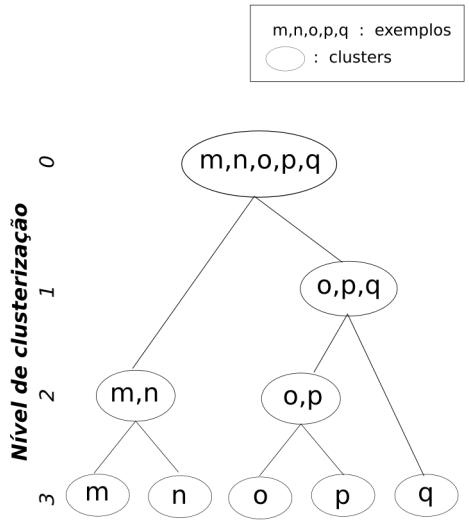

Figura 2.2: Exemplo de clusterização realizada por métodos divisivos

Devido à exigência de maior capacidade computacional, os métodos divisivos são pouco citados na literatura em relação aos métodos aglomerativos [Kaufman and Rousseeuw, 1990]. A seguir é apresen-

tado o método divisivo proposto por [Macnaughton-Smith et al., 1964], denominado MACNAUGHTON-SMITH, que evita o problema da complexidade da abordagem divisiva.

MACNAUGHTON-SMITH: O custo computacional demandado por métodos divisivos é alto, o que pode tornar a implementação inviável, caso o número de elementos seja grande e conjunto de divisões possíveis for todo considerado, o número de iterações pode aumentar exponencialmente. Entretanto o método proposto por MacNaughton-Smith é capaz de evitar esse problema [Doni, 2004]. O pseudo-código do algoritmo de MacNaughton-Smith é apresentado no Algoritmo 2. No passo 5, a distância D_m é calculada através da média aritmética da distância de cada exemplo em relação aos demais exemplos, todos pertencentes ao mesmo grupo G_j. No passo 9, a distância D_i é calculada através da média aritmética entre cada exemplo em relação aos demais exemplos, todos pertencentes ao mesmo grupo G_j, que nesse passo contém os exemplos que não foram separados do grupo. Já no passo 10, a distância D_a é calculada através da média aritmética entre as distâncias entre cada exemplo do grupo G_j (que não foram separados) e cada exemplo do grupo F_j (exemplos separados do grupo G_j).

Algoritmo 2: Pseudo-código do algoritmo divisivo de MACNAUGHTON-SMITH

Entrada: Conjunto de dados $S = \{\mathbf{x_i}, ..., \mathbf{x_N}\}$

Saída : Dendograma que representa o conjunto de grupos

1 **j=1**
2 **repeat**
3 Selecionar o grupo $\mathbf{G_j}$ com maior número de exemplos $\mathbf{N_j}$;
4 Iniciar uma matriz $\mathbf{D}_{N_j \times N_j}$;
5 Calcular a distância $\mathbf{D_m}$ de cada exemplo do grupo $\mathbf{G_j}$ em relação aos demais;
6 **while** $\mathbf{D_m} > 0$ **do**
7 Remover o exemplo $\mathbf{x}$ com maior $\mathbf{D_m}$ do grupo $\mathbf{G_j}$;
8 Armazenar o exemplo $\mathbf{x}$ no grupo $\mathbf{F_j}$;
9 (re)Calcular a distância $\mathbf{D_i}$ entre os exemplos que restaram no grupo $\mathbf{G_j}$;
10 (re)Calcular a distância $\mathbf{D_a}$ entre cada exemplo do grupo $\mathbf{G_j}$ e o grupo $\mathbf{F_j}$;
11 $\mathbf{D_m} = \mathbf{D_i} - \mathbf{D_a}$;
12 **end**
13 **j=j+1**
14 **until** *restarem apenas grupos com dois exemplos*;
15 **repeat**
16 Selecionar o grupo $\mathbf{H}$ com maior distância média;
17 Dividir o grupo $\mathbf{H}$;
18 **until** *que todos grupos sejam divididos*;

2.2 Métodos de Aprendizado Multirrótulo

Existem duas categorias principais nas quais os métodos de aprendizado multirrótulo podem ser agrupados [Tsoumakas and Katakis, 2007]. Apesar de serem métodos multirrótulo, a regra para realizar essa categorização está associada ao modo que algoritmos de classificação monorrótulo são utilizados [Tsoumakas et al., 2009]. A primeira categoria é denominada de *Transformação de Problema*. Nessa categoria, problemas de classificação multirrótulo são transformados em um ou mais problemas de classificação monorrótulo [Cherman, 2013]. A partir dessa transformação, o processo de classificação é executado da mesma maneira que em problemas de classificação monorrótulo [Modi and Panchal, 2012], ou seja, para cada problema monorrótulo transformado, são utilizados algoritmos de classificação monorrótulo para resolvê-los [Cherman, 2013]. Métodos pertencentes a categoria de transformação de problema também são chamados de métodos independente de algoritmo [Modi and Panchal, 2012]. Já na segunda categoria, denominada *Adaptação de Algoritmo*, nenhuma transformação do problema é realizada [Cherman, 2013]. Portanto, o problema multirrótulo é tratado diretamente, através de modificações em algoritmos existentes [Modi and Panchal, 2012]. Essa abordagem de criação de métodos específicos para tratar problemas multirrótulo é chamada de abordagem dependente de algoritmo [Faceli et al., 2011].

A seguir são apresentados alguns exemplos de métodos pertencentes às ambas categorias.

2.2.1 Métodos de Transformação de Problema

Métodos de transformação de problema são baseados na abordagem independente de algoritmo. Daí, para resolver o problema podem ser utilizados algoritmos de aprendizado monorrótulo. O processo para resolução do problema é feito através da realização da transformação do problema multirrótulo original em um conjunto de problemas de classificação monorrótulo. Essa transformação pode ser baseada em dois tipos [Faceli et al., 2011]: Baseada nos Rótulos e Baseada nos Exemplos.

Na transformação baseada nos rótulos, tomando um problema onde $|L|$ é o numero de rótulos, $|L|$ classificadores são construídos. Cada classificador é associado a um rótulo e treinado para resolver um problema de classificação binária [Faceli et al., 2011]. Um popular método de transformação de problemas baseado nos rótulos das classes é o *Binary Relevance* (BR) [Tsoumakas et al., 2009] que decompõe um problema de classificação multirrótulo em vários diferentes problemas de classificação binária monorrótulo, um para cada $|L|$ rótulos diferente no conjunto original S [Cherman et al., 2011]. Outro método de transformação de problema baseado nos rótulos é o *Classifier Chain* (CC) [Read et al., 2009], que combina eficiência computacional do BR e a possibilidade de usar dependências entre rótulos para classificação. O método CC envolve $|L|$ transformações binárias, uma para cada rótulo, como no BR, o que difere os dois métodos é o fato de que para cada modelo binário, o atributo do espaço é estendido com 0/1 rótulos relevantes de todos os classificadores anteriores, formando assim uma cadeia de classificadores.

Na transformação baseada em exemplos, o conjunto de rótulos associado a cada exemplo é redefinido [Faceli et al., 2011]. Um exemplo de método desse tipo amplamente utilizado é o *Label Powerset* (LP)[Tsoumakas et al., 2009].

Na Tabela 2.1 é apresentado um exemplo de conjunto de dados multirrótulo, que contém quatro exemplos $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}$ e $\mathbf{x_4}$ e seus respectivos rótulos associados. Esse conjunto de exemplo serve como base para exibir uma demonstração de execução dos métodos BR, CC e LP.

	Y
$\mathbf{x_1}$	$Y_1 = \{y_1, y_2\}$
$\mathbf{x_2}$	$Y_2 = \{y_2, y_3\}$
$\mathbf{x_3}$	$Y_3 = \{y_1, y_3, y_4\}$
$\mathbf{x_4}$	$Y_4 = \{y_2\}$

Tabela 2.1: Exemplo de conjunto de dados multirrótulo

Na Tabela 2.2 é apresentado o processo de transformação realizado pelo método BR, resultando em quatro conjunto de dados monorrótulo. Para classificação de uma nova instância, BR dá como saída a união dos rótulos positivamente preditos pelos $|L|$ classificadores [Tsoumakas et al., 2011a]. O BR é muito utilizado e apresenta resultados satisfatórios para diversos problemas. Entretanto sua limitação é não levar em conta as informações de relacionamento entre rótulos, isto é, a dependência de rótulos [Cherman, 2013]. Além da limitação relacionada à dependência de rótulos, BR pode se tornar ineficaz pelo alto custo computacional caso a base de dados possua grande quantidade de rótulos [Bi and Kwok, 2013].

	Y
x_1	y_1
x_2	$-y_1$
x_3	y_1
x_4	$-y_1$

	Y
x_1	y_2
x_2	y_2
x_3	$-y_2$
x_4	y_2

	Y
x_1	$-y_3$
x_2	y_3
x_3	y_3
x_4	$-y_3$

	Y
x_1	$-y_4$
x_2	$-y_4$
x_3	y_4
x_4	$-y_4$

Tabela 2.2: Conjuntos monorrótulo resultantes da aplicação do método BR no conjunto multirrótulo ilustrado na Tabela 2.1

Na Tabela 2.3 é apresentado o processo de transformação realizado pelo método CC, resultando em quatro conjunto de dados monorrótulo. Para classificação de uma nova instância, CC dá como saída um conjunto de rótulos preditos através das classificações em cadeia. Isto é, propagação dos resultados de cada classificadores através da cadeia de classificadores. O CC inicia o processo de classificação pelo primeiro classificador da cadeia e, após realizar a predição, repassa o resultado para o próximo classificador na cadeia com intuito de melhorar a próxima predição. Esse processo se repete $|L|$ vezes.

O **Label Powerset (LP)** é um método de transformação baseado em exemplos, que transforma um problema multirrótulo em um problema de classificação multiclasse monorrótulo, onde os possíveis valores para atributos da classe de transformação são o conjunto de únicos e distintos subconjuntos de rótulos presentes no conjunto de treino original. A aprendizagem a partir de exemplos multirrótulo correspondem em encontrar um mapeamento a partir do espaço de características dos conjuntos de rótulos, ou seja, o poder dos conjuntos de todos os rótulos.

	Dependência de rót.	Y
x_1	{0,0,0,0}	y_1
x_2	{0,0,0,0}	$-y_1$
x_3	{0,0,0,0}	y_1
x_4	{0,0,0,0}	$-y_1$

	Dependência de rót.	Y
x_1	{1,0,0,0}	y_2
x_2	{1,0,0,0}	y_2
x_3	{1,0,0,0}	$-y_2$
x_4	{1,0,0,0}	y_2

	Dependência de rót.	Y
x_1	{1,1,0,0}	$-y_3$
x_2	{1,1,0,0}	y_3
x_3	{1,1,0,0}	y_3
x_4	{1,1,0,0}	$-y_3$

	Dependência de rót.	Y
x_1	{1,1,1,0}	$-y_4$
x_2	{1,1,1,0}	$-y_4$
x_3	{1,1,1,0}	y_4
x_4	{1,1,1,0}	$-y_4$

Tabela 2.3: Conjuntos monorrótulo resultantes da aplicação do método CC no conjunto multirrótulo ilustrado na Tabela 2.1

Na Tabela 2.4 é apresentado o resultado da execução do método LP tornando um conjunto de dados multirrótulo num conjunto de dados monorrótulo multiclasse.

	Y
$\mathbf{x}_1$	$y_{1,2}$
$\mathbf{x}_2$	$y_{2,3}$
$\mathbf{x}_3$	$y_{1,3,4}$
$\mathbf{x}_4$	y_2

Tabela 2.4: Conjunto monorrótulo multiclasse resultante da aplicação do método LP no conjunto multirrótulo ilustrado na Tabela 2.1

Dado um novo exemplo, o classificador monorrótulo do LP dá como saída a classe mais provável, que é um conjunto de rótulos [Modi and Panchal, 2012]. Logo, a determinação da classe mais provável é dado de acordo com o valor de predição obtido através de um algoritmo multiclasse treinado com um conjunto de exemplos que foi gerado a partir da transformação do problema [Cherman, 2013]. Diferentemente do método BR, o método LP leva em consideração a dependência de rótulos [Modi and Panchal, 2012]. Uma limitação do LP se dá ao aumento exponencial do número de possíveis rótulos, por consequência de considerar dependências dos rótulos durante a classificação, quando uma quantidade grande ou moderada de rótulos são considerados. Daí, LP pode ter sua performance comprometida caso existam rótulos no conjunto de treino que representem muito poucos exemplos. Quando isso de fato ocorre, é conhecido como problema de desbalanceamento de classe [Cherman et al., 2011].

2.2.2 Métodos de Adaptação de Algoritmo

Métodos de adaptação de algoritmo se utilizam da abordagem dependente de algoritmo. Logo, com o intuito de tratar os problemas de classificação multirrótulo diretamente como um todo, novos algoritmos são propostos. Por se tratar de um algoritmo específico, pode apresentar resultados melhores do que métodos que seguem abordagem independente de algoritmo para um determinado problema de classificação real [Faceli et al., 2011]. Um exemplo de método de adaptação de algoritmo proposto é o *Hierarchy Of Multi-label classifERs* (HOMER) [Tsoumakas et al., 2008].

O HOMER utiliza a técnica de projeto de algoritmos, dividir e conquistar. Com isso, HOMER constrói uma hierarquia de classificadores multirrótulo, onde cada um lida com um conjunto de rótulos muito menor comparado com o conjunto total de rótulos, por conseguinte, um maior balanceamento na distribuição dos exemplos [Tsoumakas et al., 2008].

Tal hierarquia de classificadores é obtida através da construção de uma árvore, onde cada nó n contém um subconjunto de rótulos e cada nó interno n contém a união dos subconjunto de rótulos de seus filhos. Um metarrótulo de um nó n, μ_n, é o conjunto disjunto dos rótulos contidos nesse nó. Caso um exemplo de treino tenha associado algum rótulo que está contido no nó n, o exemplo é anotado com μ_n. Cada nó interno n tem associado um classificador $\mathbf{h}_n$, que utiliza os metarrótulos dos nós filhos para realizar a predição. Na Figura 2.3 é apresentado um exemplo simples da hierarquia construída pelo HOMER para uma tarefa de classificação multirrótulo com 8 rótulos.

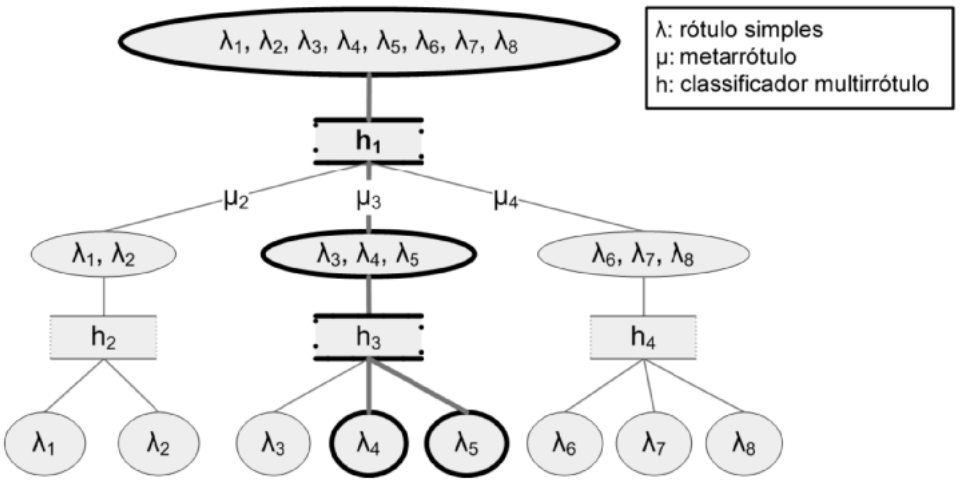

Figura 2.3: Hierarquia de rótulos e classificadores construída pelo HOMER.

Um dos principais processos internos do HOMER é a distribuição uniforme de um conjunto de rótulos em k subconjuntos disjuntos de modo que os rótulos semelhantes são colocados juntos e os rótulos que não possuem nenhuma semelhança são colocados aparte. Para execução dessa tarefa o HOMER utiliza um algoritmo denominado de *balanced k-Means* [Tsoumakas et al., 2008]. O algoritmo *balanced k-Means* é uma extensão do algoritmo *k-Means* [MacQueen et al., 1967], e o que difere os dois algoritmos é a utilização de uma constante explícita referente ao tamanho de cada cluster e a limitação do número de iterações usando um parâmetro especificado pelo usuário. O modelo chave do algoritmo *balanced k-*

means é que para cada cluster i é mantido uma lista de rótulos, C_i, ordenada em ordem ascendente da distancia dos rótulos para o centróide do cluster c_i. Quando uma inserção de um rótulo na lista ordenada de cluster é feita na posição correta, porém isso pode ocasionar em um estouro do tamanho máximo de rótulos permitidos, na lista de rótulos desse cluster, é selecionado o último rótulo, no caso o mais distante. Esse rótulo selecionado é inserido na lista do próximo cluster mais próximo. Isto pode levar a $k-1$ inserções adicionais em cascata no pior caso [Tsoumakas et al., 2008]. No Algoritmo 3 é apresentado o pseudo-código do algoritmo *balanced k-Means*, que recebe como entrada o número de clusters k, um conjunto de rótulos L, informações dos rótulos L_{info} e o valor limite de iterações do algoritmo, e dá como saída k subconjuntos de L.

O algoritmo HOMER recebe como entrada a quantidade máxima de clusters e um algoritmo multirrótulo para treinar os classificadores.

Algoritmo 3: Algoritmo *Balanced k-Means*

Entrada: número de cluster k, conjunto de rótulos L, informações dos rótulos L_{info}, limite de iterações it

Saída : k clusters balanceados de rótulos

```
1  for i ← 1 to k do
2      //inicializa clusters e seus centros;
3      C_i ← 0;
4      c_i ← membro aleatório de L;
5  end
6  while it > 0 do
7      for each λ ∈ L do
8          for i ← 1 to k do
9              d_λi ← distancia(λ, c_i, L_info);
10         end
11         finalizado ← false;
12         v ← λ;
13         while not finalizado do
14             j ← arg min d_vi;
15             Insere ordena(v, d_v) na lista ordenada C_j;
16             if |C_j| > [|L|/k] then
17                 v ← remove último elemento de C_j;
18                 d_vj ← ∞;
19             else
20                 finalizado ← true;
21             end
22         end
23     end
24     recalcula centros;
25     it ← it − 1;
26 end
27 return C_1, ..., C_k;
```

O método HOMER foi construído para lidar com problemas de classificação multirrótulo com domínios que possuam muitos rótulos de maneira efetiva e computacionalmente eficiente. A partir dessa estratégia de hierarquia de classificadores, o HOMER melhora a predição com complexidade linear para treinamento, e logarítmica para teste no que se refere a quantidade total de rótulos [Tsoumakas et al., 2008].

2.3 Características de Conjuntos de Dados Multirrótulo

Em alguns conjuntos de dados multirrótulo, o número de classes de cada exemplo é pequeno se comparado ao número total de exemplos N, enquanto em outros, o valor de N é grande. Esse número pode ser um parâmetro que influencia o desempenho dos diferentes métodos de classificação multirrótulo, como mostrado em [Rodovalho and Bernardini, 2013]. Existem duas medidas para avaliar as características de um conjunto de dados: cardinalidade $Card(S)$ e densidade $Dens(S)$ [Tsoumakas et al., 2010].

A cardinalidade de um conjunto de dados multirrótulo S – $Card(S)$ – é dada pelo número médio de rótulos dos exemplos $T_i \in S$, e é independente do número de possíveis rótulos $|L|$ – Equação 2.11. Essa medida é utilizada para quantificar o número de rótulos alternativos que caracterizam os exemplos de um conjunto de dados multirrótulo.

A densidade de um conjunto de dados multirrótulo S – $Dens(S)$ – é dada pelo número médio de rótulos dos exemplos que pertencem a S dividido pelo número total de rótulos $|L|$ – Equação 2.12. A densidade de rótulo leva o número rótulos possíveis em consideração.

$$Card(S) = \frac{1}{N}\sum_{i=1}^{N}|Y_i| \tag{2.11}$$

$$Dens(S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i|}{|L|} \tag{2.12}$$

2.4 Aprendizado Multirrótulo em problemas com muitos rótulos

Realizar tarefas de aprendizado multirrótulo aplicadas a problemas com muitos rótulos é uma tarefa complexa. Uma série de fatores podem contribuir para que dificultar a qualidade na resolução deste problema, como é o caso de bases que possuem baixa densidade, causada pela ruim distribuição dos rótulos, essa má distribuição oriunda da grande quantidade de rótulos [**?**]. Existem diversos trabalhos que visam tratar esse problema, assim como esse trabalho de conclusão de curso, nesta seção são apresentados alguns trabalhos com abordagens dististas, para cada trabalho descrito é apresentada uma breve descrição comparativa entre a abordagem para resolver esse tipo de problema e a apresentada nesta monografia.

Em [Agrawal et al., 2013] é proposta uma solução baseada no aprendizado de uma árvore *gating*, construída no espaço de características (*feature space*) e formada a partir da divisão desse espaço entre pai e seus filhos, com o objetivo de diminuir então, a quantidade de dados e rótulos que cada um tem que lidar. Cada região do espaço de características possui somente um pequeno número de rótulos

ativos, e cada rótulo ativo representa a união dos rótulos de todos os dados de treino contidos nessa região. A partir da árvore *gating* são geradas florestas aleatórias multirrótulo através da utilização de um classificador baseado em aleatorização, denominado MLRF (*Multi-Label Random Forest*). Uma floresta aleatória é composta por um conjunto de árvores de decisão, onde cada uma é treinada a partir de dados e características aleatórias. A predição é realizada da seguinte maneira, é preciso descobrir em que região o novo exemplo se encontra, daí, utiliza-se o classificador treinado com os rótulos ativos daquela região. Apesar da solução apresentada em [Agrawal et al., 2013] que utiliza informações do espaço de características para construir o modelo, possuir uma abordagem diferente deste trabalho, que usa o espaço de rótulos para construir o modelo, ambos podem ser comparados em relação ao tipo de aprendizado multirrótulo para treinar classificadores e ao modo que os rótulos são relacionados. Neste trabalho é utilizado aprendizado multirrótulo não-supervisionado enquanto em [Agrawal et al., 2013] é usado aprendizado multirrótulo supervisionado. Ambos trabalhos possuem uma tarefa de agrupar de alguma maneira os rótulos, em [Agrawal et al., 2013] essa tarefa se resume na definição de quais rótulos ativos estão em cada região utilizando um algoritmo discriminativo supervisionado, por outro lado, neste trabalho, é usado um método não supervisionado hierárquico para agrupar os rótulos em clusters.

Em [Tang et al., 2009] é apresentada uma solução para tratar problemas multirrótulo com muitos rótulos denominada *Metalabeler*, composta por 2 etapas. Na primeira etapa, é obtido um ranqueamento de classes para cada exemplo $\mathbf{x}$ no conjunto de todas classes, onde cada classe possui dados representados no espaço de características (*feature space*) e cada dado pode ter um ou mais rótulos associados. Tal ranqueamento é obtido através de vetores de pontuações originados utilizando classificadores multiclasse gerados usando a estratégia Um-Contra-Todos da técnica Máquinas de Vetores Suporte (SVM, do inglês, *Support Vector Machines*). Na segunda etapa, é realizada a descoberta do número de classes *top* a serem obtidas a partir do ranqueamento de classes. Para determinar a quantidade de classes *top*, é construído um conjunto de metadados, onde cada metadado é a composição de um metarrótulo e uma metacaracterística. Metarrótulo representa o número de rótulos associados a cada exemplo e metacaracterística representa o valor da transformação de um exemplo $\mathbf{x}$ por uma função que pode ser baseada em conteúdo, pontuação ou ranqueamento. A partir de um exemplo $\mathbf{x}$ e a quantidade máxima de rótulos, é criado um metadado utilizado como parâmetro para uma função f_{md} que mapeia as metacaracterísticas em relação ao metarrótulo. A técnica SVM Um-Contra-Todos é usada na resposta da função f_{md}, resultando assim em um metamodelo aprendido. A predição de um novo exemplo $\mathbf{x}$ é realizada inicialmente aprendendo um metamodelo, a partir desse metamodelo, é realizada a descoberta do número de rótulos associados ao exemplo $\mathbf{x}$, daí então, é realizada a seleção dos rótulos baseados nas classes que possuem as melhores pontuações dos metarrótulos. Diferente de [Tang et al., 2009], na solução proposta nessa monografia, não é utilizada a estratégia de criar metainformações a respeito dos rótulos, nem mesmo utilizar uma técnica para resolver problema puramente de classificação multiclasse.

Em [Bi and Kwok, 2013] é proposto um método de seleção de rótulos baseado na estratégia de amostragens aleatórias, onde a probabilidade de amostragem de cada rótulo, é calculada usando pontuação de importância no melhor subespaço de ranqueamento da matriz de rótulo, que representa sua importância entre todos os rótulos. A tarefa de obter os rótulos mais importantes é considerada como

um problema de seleção de subconjuntos de colunas (CSSP, do inglês *Column Subset Selection Problem*). Dado que a seleção de um subconjunto contendo k rótulos foi realizada, para cada rótulo pertencente a esse subconjunto é treinado um classificador binário, o que implica na utilização de k classificadores binários.

Já em [Xu et al., 2016], é apresentada uma solução para o problema de rótulos de cauda (em inglês, *tail labels*). Rótulos de cauda são considerados *outliers* que podem ocorrer quando utilizada a abordagem de reduzir a quantidade de rótulos através de uma matriz de rótulos, abordagem essa considerada como aproximação *low-rank*. Aproximação *low-rank* consiste em um problema de minimização, onde são usadas duas matrizes, uma de dados e uma de aproximação, a segunda delas, necessariamente possui uma classificação reduzida em relação a primeira. A matriz de aproximação é construída a partir do cálculo de uma função de custo que mede o ajuste entre ambas as matrizes [Markovsky, 2011]. É realizada uma decomposição da matriz de rótulos em duas partes. Onde a primeira parte é utilizada para realizar a aproximação *low-rank* e descrever as correlações entre rótulos. Por outro lado, a segunda parte, é um componente esparso responsável por capturar a influência entre os rótulos de cauda. Após o processo de decomposição, características de entrada são usadas para realizar a predição dos rótulos baseado no aprendizado de modelos de regressão.

Diferente deste trabalho, em [Bi and Kwok, 2013, Xu et al., 2016] são apresentadas soluções que utilizam a estratégia de ranqueamento de uma matriz de rótulos, objetivando na redução da quantidade de rótulos.

Em [Bi and Kwok, 2013, Agrawal et al., 2013, Tang et al., 2009, Xu et al., 2016] são apresentadas soluções que tratam o problema de classificação multirrótulo para grande quantidade de rótulos, por conseguinte fazem parte dos trabalhos relacionados, entretanto não são diretamente relevantes para esta monografia como o trabalho apresentado em [Tsoumakas et al., 2008], no qual é proposto um algoritmo projetado baseado na técnica de divisão e conquista, que constrói uma hierarquia de classificadores multirrótulo. Cada classificador lida com um conjunto de rótulos muito menor comparado com o total de rótulos. Esse algoritmo, denominado HOMER (do inglês, *Hierarchy Of Multi-label classifERs*) é utilizado para fins de comparação de desempenho em relação ao método proposto neste trabalho.

2.5 Avaliação de Algoritmos de Aprendizado Multirrótulo

2.5.1 Medidas de Avaliação

Para avaliar os classificadores multirrótulo, existem três grupos de medidas para avaliação induzida: baseada em instâncias, baseadas em rótulos e baseadas em *ranking* [Dimou et al., 2009]. Neste trabalho, somente são usados os primeiros dois grupos de medidas, pois *ranking* multirrótulo não é o foco desse trabalho. Do primeiro grupo, são usadas neste trabalho *Hamming Loss* (Ham) — Função de perda que mede a quantidade de exemplos rotulados erroneamente. Portanto, como uma função de perda, seu valor ótimo é zero — Outras medidas utilizadas desse grupo são: *Subset Accuracy* (SA) e F, definidas pelas Equações 2.13 à 2.15[2], respectivamente. Do segundo grupo, são usados as versões micro e macro da

[2]Na Eq. 2.13, Δ representa a diferença simétrica entre dois conjuntos.

medida $F1$. Medidas baseada em rótulos são calculadas baseadas em falso positivos f_p, falso negativos f_n, verdadeiro positivos t_p e verdadeiro negativos t_n, isto é, medidas do tipo $B(t_p, t_n, f_p, f_n)$ podem ser usadas nesse caso. Dado que t_{p_l}, t_{n_l}, f_{p_l} e f_{n_l} são verdadeiro positivos, verdadeiro negativos, falso positivos e falso negativos para cada rótulo $l \in L$, a versão micro de medidas B, denotada por B_-, é dada pela Eq. 2.16, enquanto que a versão macro de medidas B, denotada por B^-, é dada pela Eq. 2.17. Neste trabalho, foi considerada como medida B a medida $F1$, definida pela Eq. 2.18.

$$Ham(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \Delta Z_i|}{|L|} \tag{2.13}$$

$$SA(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N} I(Z_i = Y_i) \tag{2.14}$$

$$F(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \tag{2.15}$$

$$B_-(\mathbf{h}, S) = \frac{1}{|L|}\sum_{i=1}^{|L|} B(t_{p_i}, t_{n_i}, f_{p_i}, f_{n_i}) \tag{2.16}$$

$$B^-(\mathbf{h}, S) = \frac{1}{|L|} B(\sum_{i=1}^{|L|} t_{p_i}, \sum_{i=1}^{|L|} t_{n_i}, \sum_{i=1}^{|L|} f_{p_i}, \sum_{i=1}^{|L|} f_{n_i}) \tag{2.17}$$

$$F1(t_p, t_n, f_p, f_n) = \frac{2 \times f_p}{2 \times t_p + f_n + f_p} \tag{2.18}$$

2.5.2 Comparação entre algoritmos

Para realizar a comparação da performance entre algoritmos multirrótulo segundo [Friedman, 1937] são indicados testes estatísticos não paramétricos. Como neste trabalho é necessário comparar o método proposto com outros métodos e em múltiplos domínios foi selecionado o teste de *Friedman*. Após a execução desse teste, caso a hipótese nula seja rejeitada, deve ser executado um pós-teste, e para este trabalho foi selecionado o teste de *Nemenyi* [Nemenyi, 1962]. Ambos os testes são descritos a seguir.

Teste de Friedman: Consiste em utilizar os valores das medidas de desempenho de cada algoritmo para cada conjunto de dados, comparando-os baseado em um ranqueamento de desempenho, ordenado dos melhores resultados para os piores. Como resultado do teste, podemos rejeitar ou não a hipótese nula, que consiste em afirmar que, a média da performance dos A algoritmos são iguais – H_0: $m_1 = m_2 = ... = m_A$. Essa afirmação é analisada após o cálculo da estatística, definida pela Equação 2.19. Caso o resultado do teste de Friedman aponte que todos os algoritmos são equivalentes, a hipótese nula H_0 foi aceita. Por outro lado, caso a hipótese nula seja rejeitada, indica que os algoritmos possuem diferentes desempenhos, e ocorre quando a estatística calculada é maior que $F_{A-1,(A-1)(N-1)}$, onde $F_{A-1,(A-1)(N-1)}$ representa a distribuição de probabilidade F com A-1 e $(A$-1$)(N$-1) graus de liberdade [Faceli et al., 2011], onde χ_F^2 é definido pela Equação 2.20 e A é a quantidade de algoritmos, N é o número de conjuntos de dados e R_j

representa as médias ranqueamentos médios dos diferentes algoritmos, onde j representa um algoritmo dentre A algoritmos.

$$F_F = \frac{(N-1)\chi_F^2}{N(A-1)-\chi_F^2} \tag{2.19}$$

onde:

$$\chi_F^2 = \frac{12N}{A(A+1)}\left[\sum_j R_j^2 - \frac{A(A+1)^2}{4}\right] \tag{2.20}$$

O desempenho de dois algoritmos é significativamente diferente se o valor que corresponde à média dos ranqueamento se diferencia pelo menos do valor de diferença crítica CD (do inglês *Critical Difference*) [Demšar, 2006]. O valor da diferença crítica é calcula pela Equação 2.21.

$$CD = q_\alpha \sqrt{\frac{A(A+1)}{6N}} \tag{2.21}$$

Teste de Nemenyi: Bastante utilizado como pós-teste para o teste de Friedman, o teste de Nemenyi utiliza as média dos ranqueamentos dos algoritmos gerados pelo teste de Friedman e calcula uma estatística q sobre a diferença dessas médias.

Para cada par de algoritmos é calculada a estatística q, apresentada na Equação 2.22 onde R_{j1} e R_{j2} representa a média dos ranqueamentos do primeiro e o segundo algoritmo do par calculado, respectivamente.

$$q = \frac{R_{j1} - R_{j2}}{\sqrt{\frac{A(A+1)}{6N}}} \tag{2.22}$$

Pelo teste de Nemenyi, podemos afirmar com segurança que o desempenho de dois algoritmos são significativamente diferentes caso as diferenças médias dos ranqueamentos forem maiores ou iguais ao valor de diferença crítica CD [Demšar, 2006]. Os possíveis valores de q_α para o teste de Nemenyi estão disponíveis na Tabela A.1 do Apêndice A.

Neste trabalho os resultados do teste de Nemenyi são mostrados de forma gráfica, através de um simples diagrama apresentado em [Demšar, 2006]. Na Figura 2.4 é ilustrado os resultados do teste de Nemenyi onde são comparados entre si quatro diferentes algoritmos, denominados por Alg1, Alg2, Alg3 e Alg4. É utilizada uma escala de 1 a 4, pois são quatro algoritmos sendo comparados, onde os valores médios de suas posições no ranqueamento calculados no teste de Friedman são utilizados para posicionar os algoritmos. O valor de diferença crítica (CD) é representado acima da escala de ranqueamentos, posicionado a partir da esquerda. Nesse exemplo em específico o valor de CD utilizado foi 2.850. Caso o resultado do teste indique que certos algoritmos não possuam diferenças estatísticas, então esses algoritmos serão interligados por uma linha horizontal. Todos os dados utilizados para construir essa representação gráfica, assim como o resultado do grupamento são apresentados na Tabela 2.5.

Pela análise da Figura 2.4, pode-se identificar que:

- Alg1 apresenta um resultado significativamente superior a Alg4;

Algoritmo	Média dos Ranqueamento	Grupos	
Alg1	1.167	A	
Alg2	2.250	A	B
Alg3	2.667	A	B
Alg4	3.667		B

Tabela 2.5: Exemplo de dados de entrada e resultado final do teste de Nemenyi.

- Alg1, Alg2 e Alg3 não possuem diferenças estatísticas sobre seus resultados;
- Alg2, Alg3 e Alg4 não possuem diferenças estatísticas sobre seus resultados;
- Nada pode ser concluído sobre os resultados de Alg2 e Alg3, porque eles pertencem a duas populações e isso não é permitido em termos estatísticos; e
- Alg1 e Alg4 possuem diferenças estatísticas sobre seus resultados pois cada um pertence a somente uma e diferente população.

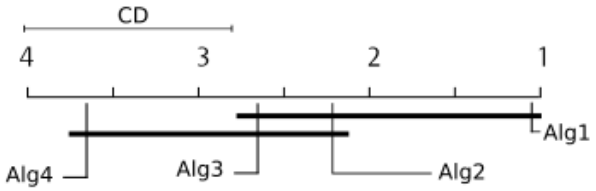

Figura 2.4: Representação gráfica do teste de Nemenyi a partir dos dados da Tabela 2.5.

Capítulo 3

Proposta de um Método de Aprendizado Multirrótulo baseado em Aprendizado de Máquina Não-Supervisionado Hierárquico

O método proposto nesse trabalho é denominado MUDIH – *MUltiLabel method based on DIvisive Hierarchical clustering*. O MUDIH é um método de aprendizado multirrótulo que utiliza da abordagem dependente de algoritmo, o que significa dizer, que sua classificação é de método de adaptação de algoritmo. Para tarefas de análises de dados, MUDIH utiliza a técnica de clusterização hierárquica utilizando a abordagem divisiva. Em outro trabalho em desenvolvimento também orientado pela professora orientadora deste trabalho, é explorado o uso da clusterização hierárquica utilizando a abordagem aglomerativa. Para apoiar o desenvolvimento foram utilizadas as ferramentas Mulan [Tsoumakas et al., 2011b] e Weka [Hall et al., 2009]. Nesse capítulo são apresentados: uma breve descrição das ferramentas utilizadas, incluindo o formato de entrada de dados que compõem uma base de dados multirrótulo; e uma descrição do MUDIH.

3.1 Ferramentas Utilizadas

WEKA: O WEKA (*Waikato Environment for Knowledge Analysis*) [Holmes et al., 1994] é composto por um conjunto de algoritmos de aprendizado de máquina e ferramentas de pré-processamento de dados desenvolvidos para resolver problemas de mineração de dados. Em sua vasta coleção de algoritmos, estão incluídos algoritmos de classificação, regressão, clusterização, regras de associação, seleção de atributos e ferramentas de visualização [Hall et al., 2009]. WEKA foi desenvolvido com o intuito de ajudar pesquisadores e profissionais de mercado. Logo, através da sua arquitetura modularizada, permite que sejam construídas outras aplicações de mineração de dados utilizando WEKA como base, isso se dá

através de uma *API* (do inglês, *Application Programming Interface*) descomplicada.

Mulan: Mulan [Tsoumakas et al., 2011b] é uma biblioteca de código aberto, escrita na linguagem Java e construída baseada no WEKA. Mulan foi desenvolvida com o intuito de dar suporte a realização de tarefas de aprendizado de máquina a partir de conjuntos de dados multirrótulo. Tal suporte a conjuntos multirrótulo não é fornecido pelo WEKA. Mesmo tendo WEKA como base, a biblioteca Mulan possui vários aspectos independentes, além de conter interfaces para grande parte das classes principais. Uma enorme quantidade de algoritmos para classificações multirrótulo é disponibilizada pela Mulan, além de conter um *framework* de avaliação que calcula uma grande variedade de medidas de avaliação multirrótulo [Tsoumakas et al., 2011b]. Nos parágrafos a seguir, são apresentados respectivamente, como as informações referentes ao conjunto de dados multirrótulo são formatadas para serem interpretadas pela biblioteca Mulan e como devem ser implementados novos métodos de aprendizado multirrótulo respeitando a *API* do Mulan.

Formato de entrada do conjunto de dados multirrótulo: Assim como toda ferramenta de análise e processamento de dados, é necessário algum tipo de organização dos dados para que seja possível sua interpretação. Como especificação para os conjuntos de dados, Mulan necessita de dois arquivos de texto [Tsoumakas et al., 2011b]. O primeiro deles é um arquivo ARFF (*Attribute Relation File Format*), que contém informações sobre exemplos, atributos e suas relações. Um atributo possui um nome, um tipo de dados e um intervalo de valores [Holmes et al., 1994]. Existe uma peculiaridade na representação do arquivo ARFF quando se trata de um conjunto de dados multirrótulo. Neste caso, é preciso que os rótulos sejam declarados como atributos nominais, que podem conter o valor ”0”ou ”1”. Se o rótulo estiver presente no exemplo o atributo recebe o valor ”1”, caso contrário recebe ”0”. Representado em formato XML, o segundo arquivo contém a declaração dos rótulos e possíveis hierarquias entre eles [Tsoumakas et al., 2011b]. Nas Tabelas 3.1 e 3.2 é apresentado um exemplo de conjunto de dados multirrótulo definido pelos arquivos ARFF e XML, respectivamente. O conjunto mostrado contém quatro atributos, três rótulos e seis exemplos.

Estrutura de codificação (Mulan *API*): O Mulan também dispõe de uma *API* para ser mais facilmente utilizada. Para implementação de um método de classificação multirrótulo é necessário estender a classe base *MultiLabelLearnerBase*. Entretanto, caso a implementação desejada seja um método multirrótulo que utilize no seu processo interno algum outro método multirrótulo, a classe base a ser estendida é a *MultiLabelMetaLearner*. Nos dois casos, são obrigatórias as implementações de dois métodos herdados, o primeiro deles é denominado *buildInternal*, onde deve conter a implementação específica do método proposto, a partir de um conjunto de dados de treino. No segundo método, *makePredictionInternal*, é especificada a implementação da predição dos dados em específico baseado no conjunto de dados treinado. Ainda, o Mulan oferece para os seus usuários um *framework* de avaliação para cálculos de medidas. De modo que possa usufruir dos recursos disponibilizados, o algoritmo MUDIH foi construído seguindo o design desta *API*. Portanto é uma subclasse de *MultiLabelMetaLearner*.

```
@relation ConjuntoMultirrotulo
@attribute Atributo1 numeric
@attribute Atributo2 numeric
@attribute Atributo3 numeric
@attribute Atributo4 numeric
@attribute Lambda1 {0,1}
@attribute Lambda2 {0,1}
@attribute Lambda3 {0,1}
@data
0.09,0.80,0.45,5.76,0,1,1
0.27,0.48,0.82,0.11,1,0,0
2.79,0.75,0.57,0.19,0,1,0
0.64,0.80,0.48,6.71,0,0,1
0.64,0.57,0.55,0.25,1,0,1
0.76,0.13,0.15,0.21,0,1,1
```

Tabela 3.1: Exemplo de arquivo ARFF de um conjunto de dados multirrótulo.

```
<?xml version="1.0" encoding="utf-8"?>
<labels xmlns="http://mulan.sourceforge.net/labels">
<label name="Lambda1"></label>
<label name="Lambda2"></label>
<label name="Lambda3"></label>
</labels>
```

Tabela 3.2: Exemplo de arquivo XML de um conjunto de dados multirrótulo.

3.2 O método MUDIH

A ideia principal do MUDIH é treinar um classificador para cada subconjunto obtido a partir da separação do conjunto de dados original, por meio de clusterização baseada na relação entre os rótulos. Dado um novo exemplo, é realizada a predição através da combinação de predições parciais obtidas por meio da execução dos classificadores treinados em cada subconjunto gerado.

MUDIH utiliza no seu processamento interno um algoritmo multirrótulo utilizado para treinar um classificador, e também uma versão modificada do algoritmo de MacNaughton-Smith, apresentado na Seção 6, como base no processo de clusterização. Para a tarefa de clusterização é necessário que seja especificada a quantidade máxima de clusters desejado. A necessidade dessa quantidade especificada motivou a modificação do algoritmo original de MacNaughton-Smith para suportar esse critério de parada. Tanto o algoritmo multirrótulo quanto o valor máximo de clusters permitido são inseridos pelo usuário.

Para execução da tarefa de clusterização é necessária alguma medida de distância para que possam ser inicializadas as matrizes de distância. MUDIH trata cada rótulo como uma representação de um ponto em um espaço de N dimensões, onde N é a quantidade de exemplos no conjunto de dados. Portanto para montar as matrizes de distância é calculada a distância euclidiana entre esses pontos. Essa distância foi escolhida por ser utilizada em diversos trabalhos na literatura. Utilizando como exemplo o conjunto de dados multirrótulo apresentado na Tabela 3.1, as Tabelas 3.3 e 3.4 apresentam a representação dos rótulos em pontos e a matriz de distância gerada a partir desses pontos, respectivamente. Deve ser observado que, nesse caso, o que se quer obter é a distância entre os rótulos. Assim os dados exibidos na Tabela 3.4 é a matriz transposta dos dados relativos somente aos rótulos dos dados exibidos na Tabela 3.3. Ainda para fins de simplificações, Lambda1, Lambda2 e Lambda3 da Tabela 3.1 é respectivamente λ_1, λ_2 e λ_3 das Tabelas 3.3 e 3.4.

Rótulo	Ponto (6 dimensões)
λ_1	(0,1,0,0,1,0)
λ_2	(1,0,1,0,0,1)
λ_3	(1,0,0,1,1,1)

Tabela 3.3: Representação dos rótulos através pontos no espaço pelo MUDIH.

	λ_1	λ_2	λ_3
λ_1	0	d(λ_1,λ_2)	d(λ_1,λ_3)
λ_2	d(λ_2,λ_1)	0	d(λ_2,λ_3)
λ_3	d(λ_3,λ_1)	d(λ_3,λ_2)	0

	λ_1	λ_2	λ_3
λ_1	0	2,24	2
λ_2	2,24	0	1,73
λ_3	2	1,73	0

Tabela 3.4: Matriz intermediária e resultante gerada a partir da representação dos rótulos através pontos no espaço pelo MUDIH.

No Algoritmo 4 é apresentado em pseudo-código o método MUDIH contendo as implementações dos métodos *buildInternal* (linhas 2 a 32) e *makePredictionInternal* (linhas 33 a 37). O método aceita como entrada o número de clusters k, um algoritmo multirrótulo A e um conjunto multirrótulo de treino S que tem L como conjunto de rótulos associados. MUDIH dá como saída a avaliação de performance do classificador multirrótulo. No passo 1, é criada uma lista c que contém os clusters gerados pelo processo de clusterização, e c é inicializada com um grupo inicial contendo todos os rótulos. No passo 3 é realizada a transposição da matriz de exemplos, resultando numa matriz que contém a representação dos rótulos através de pontos no espaço. Do passo 4 ao 26 possui a mesma sequência de passos do algoritmo de MACNAUGHTON-SMITH apresentado no Algoritmo 2, exceto pelos passos 16,17 e 25 que fazem atualizações na lista de clusters gerados, e o passo 19 que verifica se o número de clusters gerados é igual k. Caso essa verificação seja verdadeira, a etapa de clusterização é terminada. No passo 28, para cada cluster G_i gerado da lista c é criado um novo conjunto de dados multirrótulo S_i contendo somente os rótulos contidos em G_i e no passo seguinte um classificador é treinado utilizando o algoritmo A e o conjunto S_i. No passo 35 é realizada a predição para um novo exemplo através da combinação dos

resultados das predições de cada classificadores treinados no passo 30.

Algoritmo 4: Algoritmo MUDIH

Entrada: número de clusters $\boldsymbol{k}$, algoritmo multirrótulo $\boldsymbol{A}$,base de dados multirrótulo de treino S com N elementos

Saída : Avaliação de performance do classificador multirrótulo

1 $c \leftarrow$ grupo inicial G_1 com todos os rótulos de L;
2 `buildInternal(`S`):`
3 **Traspor matriz de exemplos de M dimensões para N dimensões considerando somente os dados dos rótulos**;
4 **j=1**
5 **repeat**
6 Selecionar de c o grupo G_j com maior número de elementos N_j;
7 Iniciar uma matriz $D_{N_j \times N_j}$;
8 Calcular a distância média D_m de cada elemento do grupo G_j em relação aos demais;
9 **while** $D_m >= 0$ **do**
10 Remover o rótulo λ com maior D_m do grupo G_j;
11 Armazenar o rótulo λ no grupo F_j;
12 (re)Calcular D_i entre os elementos que restaram no grupo G_j;
13 (re)Calcular D_a entre cada elemento do grupo G_j e o grupo F_j;
14 $D_m = D_i - D_a$;
15 **end**
16 Atualizar grupo G_j em c;
17 Adicionar grupo F_j em c;
18 **j=j+1**
19 **if** *quantidade de grupos em* $c == k$ **then**
20 **break**;
21 **until** *restarem em c apenas grupos com dois elementos*;
22 **repeat**
23 Selecionar em c o grupo **G** com maior distância média;
24 Dividir o grupo **G** em dois G_1 e G_2;
25 Atualizar grupos em c
26 **until** *que todos grupos sejam divididos*;
27 **i=1**;
28 **for** *cada grupo* G_i *pertencente a* c **do**
29 Criar um novo conjunto de dados S_i contendo somente os rótulos contidos no grupo G_i;
30 Treinar um classificador $\mathbf{h}_i$ utilizando **A** e o conjunto S_i;
31 **end**
32 **end**;
33 `makePredictionInternal(`*NovoExemplo* $\boldsymbol{e}$`):`
34 **for** *cada classificador* $\mathbf{h_i}$ *treinado no passo 30* **do**
35 Realizar predição do exemplo $\boldsymbol{e}$;
36 **end**
37 **return** combinação das predições obtidas;

Nas Figuras 3.1 e 3.2 são apresentados respectivamente o funcionamento do métodos *buildInternal* e *makePrediction* sendo 2 o número máximo de cluster.

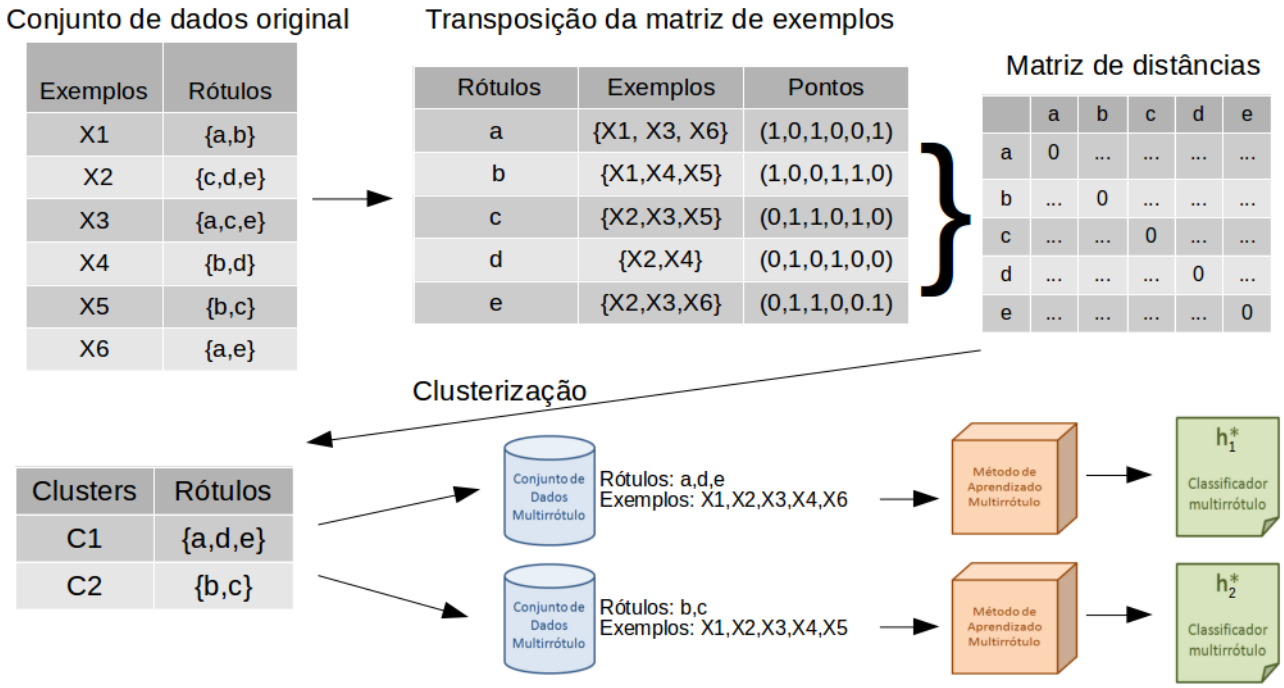

Figura 3.1: MUDIH - *buildInternal*.

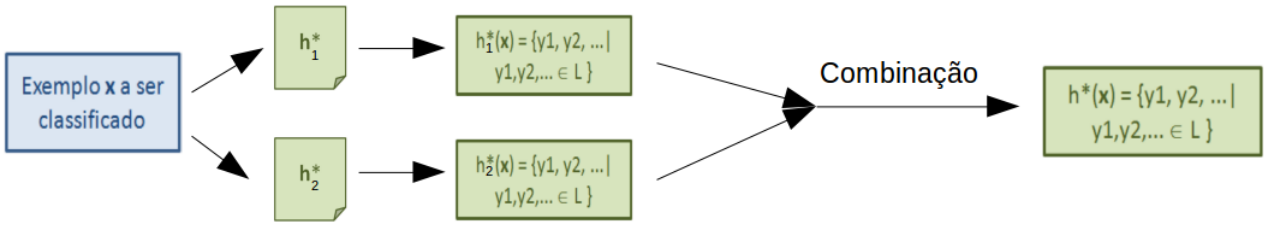

Figura 3.2: MUDIH - *makePrediction*.

Capítulo 4

Experimentos Realizados

Para conduzir os experimentos, foram realizados testes comparativos de performance entre o MUDIH e os métodos multirrótulo descritos na Seção 2.2 — BR, LP, CC e HOMER. O J48, foi utilizado como algoritmo de aprendizado supervisionado monorrótulo base para os métodos multirrótulo. O J48 é disponibilizado pela ferramenta WEKA. Tal algoritmo é uma implementação do algoritmo C4.5.

4.1 Descrição das Bases de Dados

Nesta seção, as bases de dados utilizadas neste trabalho são brevemente descritas, apresentando seus respectivos domínios e atributos. Foram usadas bases de dados naturais escolhidas aleatoriamente, para avaliar o desempenho do método MUDIH. No total foram utilizadas seis bases de dados, que se diferenciam por algumas características. A seguir são apresentadas cada uma delas, e na Tabela 4.1 são apresentadas as características das bases de dados, onde Domínio é o domínio do conjunto de dados S; $Card(S)$ e $Dens(S)$ são, respectivamente, a cardinalidade e densidade de S, definidas pelas Equações 2.11 e 2.12; q é o número de rótulos, N é o número de exemplos, $Attr$ é a quantidade de atributos e $Attr_t$ é o tipo do atributo. Todas as bases de dados utilizadas neste trabalho estão disponíveis em *http://mulan.sourceforge.net/datasets-mlc.html*.

- **Emotions:** Base construída a partir de um experimento baseado na detecção e classificação das emoções em músicas. Características musicais como ritmo e timbre foram utilizados para diferenciação das emoções. O experimento envolveu diversos gêneros musicais [Trohidis et al., 2008].
- **Enron:** Base que contem informações sobre mensagens de emails. Foram extraídas informações obtidas a partir de uma investigação de fraude contábil praticada em 2001 pela empresa do ramo de distribuição energética, a Enron [Klimt and Yang, 2004].
- **Genbase:** Base com dados de domínio de biologia. Essa base contém informações sobre famílias de proteínas [Diplaris et al., 2005].
- **Medical:** Base que contem informações clínicas de pacientes anônimos, referentes a dados iniciais e problemas médicos [Pestian et al., 2007].

- **Scene:** Base que contem dados sobre imagens, e foi construída para realizar classificação de cenários [Boutell et al., 2004].
- **Yeast:** Base formada por perfis filogenéticos e coleções de dados relacionados à expressões gênicas [Elisseeff and Weston, 2001].

	Características						
Base	Domínio	$Card(S)$	$Dens(S)$	q	N	$Attr$	$Attr_t$
Emotions	Música	1.869	0.311	6	593	72	Numérico
Enron	Texto	3.378	0.064	53	1702	1001	Nominal
Genbase	Biologia	1.252	0.046	27	662	1186	Nominal
Medical	Texto	1.245	0.028	45	978	1449	Nominal
Scene	Imagem	1.074	0.179	6	2407	294	Numérico
Yeast	Biologia	4.237	0.303	14	2417	103	Numérico

Tabela 4.1: Características das bases de dados utilizadas neste trabalho.

4.2 Metodologia Experimental

Para realizar a comparação dos algoritmos de aprendizado e suas respectivas avaliações, foi usada a técnica de validação cruzada por k partições, conhecida como *k-fold cross-validation* [Refaeilzadeh et al., 2009], onde k é o número de partições. Neste método é realizada uma divisão dos dados em k partições. Na primeira iteração do algoritmo, a partição k é usada como conjunto de teste, e as outras partições são usadas para compor o conjunto de treinamento. Na segunda iteração, a partição $k-1$ é utilizada para teste e as outras partições para conjunto de treinamento. O processo se repete k vezes. Na Figura 4.1 é apresentado o funcionamento do método *k-fold cross-validation* com $k = 10$. A cada iteração i, um modelo h_i é treinado, e obtém-se um valor de uma medida de avaliação Med_i que representa a performance do modelo sobre o respectivo conjunto de teste S_{te_i}. No final de todas iterações, é calculada a média de todas as medidas obtidas, assim é possível dar como saída a medida $\overline{Med}$. Para os experimentos realizados neste trabalho o valor de k escolhido foi 10.

Na Tabela 4.2 é mostrado o número de rótulos por clusters construídos utilizando a versão modificada do algoritmo hierárquico MacNaughton-Smith implementada no MUDIH. Para o experimento realizado neste trabalho a quantidade máxima de clusters escolhida foi 5. Contudo, é possível perceber para as bases Emotions e Scene, o processo de clusterização resultou em 6 clusters. Isto se ocorreu pois ambas as bases possuem 6 rótulos, logo no passo 16 do Algoritmo 4, a condição não é satisfeita. Porém a condição do passo 18 é satisfeita. Portanto no processo de clusterização que possuía 4 clusters para ambos, foram divididos nos passo 19 a 21, resultando em 6 clusters.

Nas Tabelas 4.3 a 4.7 são apresentados os resultados aproximados das medidas de avaliação descritas na Seção 2.5.1 - *Hamming Loss* – $Ham(\mathbf{h}, S)$, *Subset Accuracy* – $SA(\mathbf{h}, S)$, Medida F – $F(\mathbf{h}, S)$, *Micro F1* – $F1\_(\mathbf{h}, S)$, *Macro F1* – $F1^{-}(\mathbf{h}, S)$. Tais dados foram obtidos a partir da execução dos

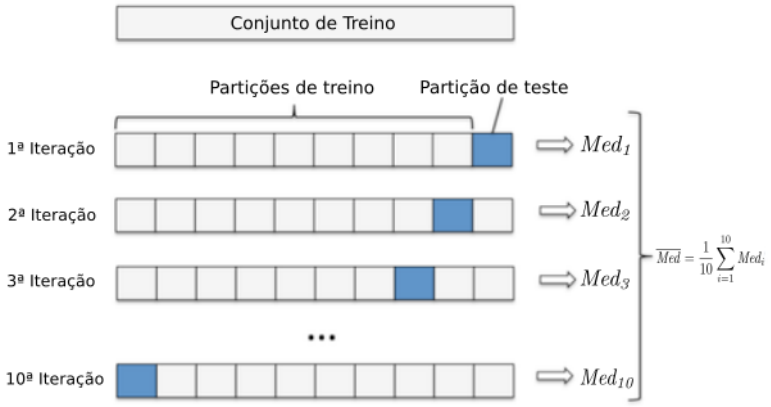

Figura 4.1: Exemplo de execução do método de validação cruzada por 10 partições. Imagem adaptada de [Raschka, 2016].

Base	Número de rótulos por cluster	#Clusters
Emotions	1,1,1,1,1,1	6
Enron	2,1,1,48,1	5
Genbase	1,1,1,23,1	5
Medical	1,1,1,41,1	5
Scene	1,1,1,1,1,1	6
Yeast	2,2,2,6,2	5

Tabela 4.2: Número de rótulos por cluster após processo de clusterização do MUDIH.

algoritmos BR, LP, CC, MUDIH e HOMER. Para os algoritmos MUDIH e HOMER foram realizadas duas execuções para cada, variando a utilização do método LP e CC. Nessas tabelas, o melhor resultado obtido para conjunto de dados foi sublinhado, neste caso, o resultado com valor exato, isto é, sem aproximação.

Hamming Loss							
Conjunto de Dados	LP	MUDIH-LP	HOMER-LP	CC	MUDIH-CC	HOMER-CC	BR
EMOTIONS	0.28±0.02	0.25±0.02	0.27±0.02	0.26±0.02	0.25±0.02	0.26±0.03	0.25±0.02
ENRON	0.07±0.00	0.06±0.00	0.07±0.00	0.05±0.00	0.05±0.00	0.06±0.00	0.05±0.00
GENBASE	0.002±0.002	0.002±0.002	0.002±0.002	0.001±0.001	0.001±0.001	0.002±0.001	0.001±0.001
MEDICAL	0.013±0.002	0.011±0.002	0.012±0.002	0.010±0.002	0.010±0.001	0.011±0.001	0.010±0.001
SCENE	0.14±0.01	0.14±0.09	0.14±0.01	0.14±0.02	0.14±0.09	0.14±0.01	0.14±0.01
YEAST	0.28±0.02	0.27±0.01	0.27±0.01	0.27±0.01	0.25±0.01	0.26±0.01	0.25±0.01

Tabela 4.3: Resultados obtidos para medida *Hamming Loss*

A seleção dos métodos utilizados para realizar a comparação de resultados foi tomada baseando-se em algumas características desses algoritmos, que são:

- **Clássicos na literatura:** É o caso dos métodos BR e LP, por serem referências amplamente utilizados em muitos outros trabalhos;

Subset Accuracy							
Conjunto de Dados	LP	MUDIH-LP	HOMER-LP	CC	MUDIH-CC	HOMER-CC	BR
EMOTIONS	0.21±0.05	0.18±0.05	0.22±0.05	0.25±0.03	0.18±0.05	0.23±0.06	0.18±0.05
ENRON	0.11±0.03	0.10±0.02	0.11±0.02	0.13±0.01	0.10±0.01	0.11±0.01	0.10±0.02
GENBASE	0.97±0.01	0.97±0.01	0.97±0.02	0.97±0.02	0.97±0.02	0.96±0.02	0.97±0.02
MEDICAL	0.67±0.05	0.66±0.05	0.68±0.04	0.68±0.04	0.66±0.04	0.68±0.03	0.66±0.04
SCENE	0.55±0.04	0.43±0.05	0.56±0.03	0.54±0.04	0.43±0.04	0.53±0.04	0.43±0.03
YEAST	0.13±0.02	0.08±0.01	0.11±0.01	0.15±0.02	0.10±0.02	0.11±0.02	0.07±0.02

Tabela 4.4: Resultados obtidos para medida *Subset Accuracy*

Medida F							
Conjunto de Dados	LP	MUDIH-LP	HOMER-LP	CC	MUDIH-CC	HOMER-CC	BR
EMOTIONS	0.51±0.03	0.56±0.04	0.54±0.04	0.55±0.04	0.56±0.04	0.56±0.04	0.56±0.04
ENRON	0.44±0.03	0.47±0.03	0.46±0.02	0.53±0.03	0.52±0.04	0.50±0.02	0.53±0.03
GENBASE	0.99±0.01	0.99±0.01	0.98±0.01	0.99±0.00	0.99±0.00	0.99±0.01	0.99±0.00
MEDICAL	0.76±0.03	0.78±0.02	0.78±0.04	0.79±0.04	0.78±0.03	0.78±0.02	0.78±0.03
SCENE	0.60±0.04	0.58±0.03	0.61±0.04	0.60±0.04	0.57±0.03	0.61±0.04	0.57±0.03
YEAST	0.52±0.03	0.54±0.02	0.52±0.01	0.53±0.02	0.56±0.02	0.53±0.02	0.56±0.02

Tabela 4.5: Resultados obtidos para medida F

Micro *F1*							
Conjunto de Dados	LP	MUDIH-LP	HOMER-LP	CC	MUDIH-CC	HOMER-CC	BR
EMOTIONS	0.55±0.02	0.60±0.03	0.57±0.03	0.59±0.03	0.60±0.03	0.59±0.04	0.60±0.03
ENRON	0.43±0.02	0.48±0.02	0.46±0.02	0.54±0.03	0.55±0.03	0.51±0.02	0.55±0.03
GENBASE	0.98±0.02	0.98±0.02	0.98±0.02	0.99±0.06	0.99±0.06	0.98±0.01	0.99±0.06
MEDICAL	0.76±0.03	0.79±0.02	0.77±0.04	0.81±0.03	0.81±0.03	0.79±0.02	0.81±0.03
SCENE	0.60±0.04	0.62±0.02	0.61±0.04	0.60±0.04	0.62±0.02	0.61±0.04	0.62±0.02
YEAST	0.54±0.03	0.56±0.02	0.55±0.01	0.55±0.02	0.59±0.01	0.56±0.02	0.59±0.02

Tabela 4.6: Resultados obtidos para medida Micro *F1*

Macro *F1*							
Conjunto de Dados	LP	MUDIH-LP	HOMER-LP	CC	MUDIH-CC	HOMER-CC	BR
EMOTIONS	0.54±0.02	0.59±0.03	0.56±0.04	0.58±0.03	0.59±0.03	0.58±0.04	0.59±0.03
ENRON	0.23±0.04	0.26±0.03	0.27±0.04	0.31±0.06	0.31±0.05	0.27±0.03	0.32±0.05
GENBASE	0.95±0.03	0.95±0.03	0.93±0.04	0.96±0.03	0.96±0.03	0.95±0.02	0.96±0.03
MEDICAL	0.69±0.05	0.72±0.04	0.69±0.07	0.76±0.04	0.76±0.04	0.73±0.05	0.76±0.04
SCENE	0.61±0.03	0.63±0.02	0.62±0.03	0.61±0.04	0.63±0.02	0.63±0.04	0.63±0.02
YEAST	0.38±0.03	0.40±0.02	0.39±0.01	0.40±0.02	0.39±0.01	0.37±0.02	0.39±0.02

Tabela 4.7: Resultados obtidos para medida Macro *F1*

- **Relacionamento entre rótulos**: É o caso do CC, que trata a dependência entre rótulos na tarefa de classificação; e
- **Semelhança com o método proposto:** É o caso do HOMER, que possui características seme-

Medida	Q (valor observado)	p-valor	Rejeita H_0
Ham	26.183	0	Sim
SA	20.289	0,002	Sim
F	12.073	0,06	Não
$F1\_$	28.642	<0,0001	Sim
$F1^-$	23.122	0,001	Sim

Tabela 4.8: Estatísticas dos resultados dos algoritmos.

lhantes ao método MUDIH.

4.3 Análise dos Resultados

Para comparar os resultados obtidos foi utilizado o teste estatístico de Friedman, e como pós-teste o teste de Nemenyi, ambos descritos na Seção 2.5.2. A seguir são apresentados os resultados dos testes de Friedman e Nemenyi aplicados aos dados de performance dos algoritmos referentes às medidas de avaliação mostrados nas Tabelas 4.3 a 4.7. Para facilitar a interpretação dos testes, temos que:

- Hipótese nula H_0 – As amostras vêm de uma mesma população (média das performances dos algoritmos são iguais);
- Hipótese alternativa H_a – As amostras não vêm de uma mesma população (média são diferentes).

Na Tabela 4.8 são apresentadas as estatísticas descritivas dos resultados dos algoritmos e o resultado do teste de Friedman para as medidas $Ham(\mathbf{h}, S)$; $SA(\mathbf{h}, S)$; $F(\mathbf{h}, S)$; $F1\_(\mathbf{h}, S)$; $F1^-(\mathbf{h}, S)$. O valor Q (valor crítico) para todas as medidas é 12.592 e o nível de confiança dos testes é de 95%.

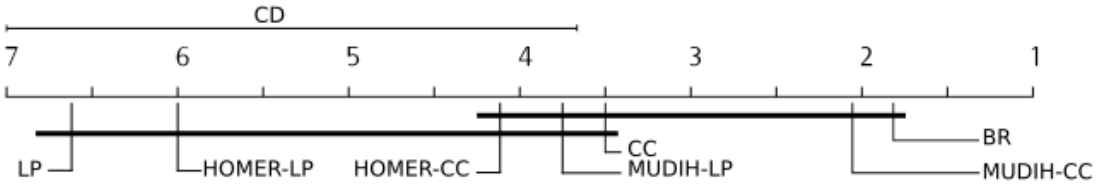

Figura 4.2: Representação gráfica das comparações múltiplas pareadas obtidas através do teste de Nemenyi para medida *Hamming Loss*.

Nas Figuras 4.2 a 4.5 são exibidos os resultados do teste de Nemenyi para as medidas cuja hipótese nula H_0 foi rejeitada segundo o teste de Friedman que, para medida a medida *Hamming Loss* [1]:

- LP apresenta um resultado significativamente superior aos algoritmos BR e MUDIH-CC;

[1] É importante observar que a medida *Hamming Loss* tem zero como valor ótimo. Assim, a interpretação do ranqueamento deve ser feita de maneira inversa.

- HOMER-LP apresenta um resultado significativamente superior aos algoritmos BR e MUDIH-CC;
- LP possui diferenças estatísticas em relação aos algoritmos BR e MUDIH-CC;
- HOMER-LP possui diferenças estatísticas em relação aos algoritmos BR e MUDIH-CC;
- LP, HOMER-LP, HOMER-CC, MUDIH-LP e CC não possuem diferenças estatísticas sobre seus resultados;
- HOMER-CC, MUDIH-LP, CC, MUDIH-CC e BR não possuem diferenças estatísticas sobre seus resultados; e .
- Nada pode ser concluído sobre os resultados dos algoritmos HOMER-CC, MUDIH-LP e CC.

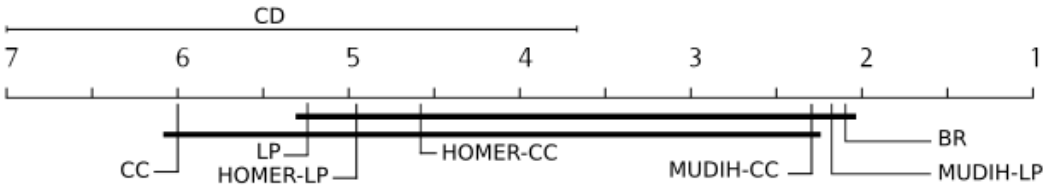

Figura 4.3: Representação gráfica das comparações múltiplas pareadas obtidas através do teste de Nemenyi para medida *Subset Accuracy*.

Pode-se observar analisando a Figura 4.3 que, para a medida *Subset Accuracy*:

- MUDIH-LP obteve uma boa posição no ranqueamento, em segundo lugar;
- BR e MUDIH-LP apresenta um resultado significativamente superior ao algoritmo CC;
- BR e CC possuem diferenças estatísticas;
- MUDIH-LP e CC possuem diferenças estatísticas sobre seus resultados;
- BR, MUDIH-LP, MUDIH-CC, HOMER-CC, HOMER-LP e LP não possuem diferenças estatísticas sobre seus resultados;
- MUDIH-CC, HOMER-CC, HOMER-LP, LP e CC não possuem diferenças estatísticas sobre seus resultados; e
- Nada pode ser concluído sobre os resultados dos algoritmos MUDIH-CC, HOMER-CC, HOMER-LP e LP.

Pode-se observar, a Figura 4.4, analisando que, para a medida Micro *F1*:

- LP apresenta um resultado significativamente superior ao BR e MUDIH-CC;
- MUDIH-CC possui diferenças estatísticas em relação aos algoritmos LP e HOMER-LP;

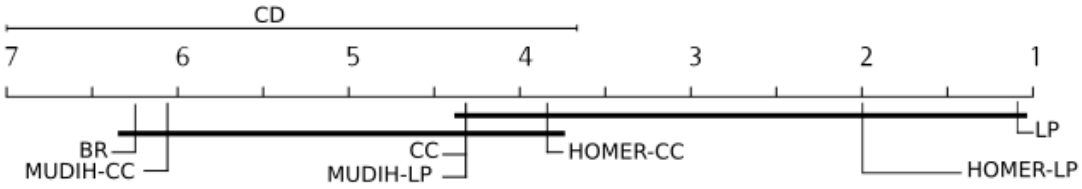

Figura 4.4: Representação gráfica das comparações múltiplas pareadas obtidas através do teste de Nemenyi para medida Micro *F1*.

- BR possui diferenças estatísticas em relação aos algoritmos LP e HOMER-LP;
- HOMER-LP possui diferenças estatísticas com os algoritmos BR e MUDIH-CC;
- LP, HOMER-LP, HOMER-CC, CC e MUDIH não possuem diferenças estatísticas sobre seus resultados;
- HOMER-CC, CC, MUDIH-LP, MUDIH-CC e BR não possuem diferenças estatísticas sobre seus resultados;
- Nada pode ser concluído sobre os resultados dos algoritmos HOMER-CC, MUDIH-LP e CC; e
- Os algoritmos MUDIH-LP e MUDIH-CC não obtiveram bom posicionamento no ranqueamento.

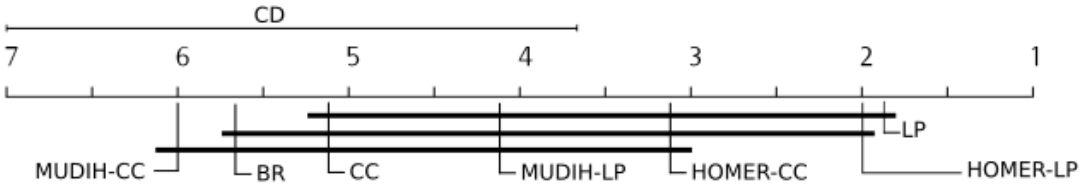

Figura 4.5: Representação gráfica das comparações múltiplas pareadas obtidas através do teste de Nemenyi para medida Macro *F1*

Analisando a Figura 4.5, pode-se observar que os resultados para a medida Macro *F1*:

- LP apresenta um resultado significativamente superior aos algoritmos BR e MUDIH-CC;
- LP possui diferenças estatísticas em relação aos algoritmos BR e MUDIH-CC;
- HOMER possui diferenças estatísticas em relação ao algoritmos MUDIH-CC;
- LP, HOMER-LP, HOMER-CC, MUDIH-LP e CC não possuem diferenças estatísticas sobre seus resultados;

- HOMER-LP, HOMER-CC, MUDIH-LP,CC e BR não possuem diferenças estatísticas sobre seus resultados;
- HOMER-CC, MUDIH-LP,CC, BR e MUDIH-CC não possuem diferenças estatísticas sobre seus resultados;
- Nada pode ser concluído sobre os resultados dos algoritmos HOMER-LP, HOMER-CC, MUDIH-LP, CC e BR; e
- Os algoritmos MUDIH-CC não obteve bom posicionamento no ranqueamento.

Capítulo 5

Conclusão e Trabalhos Futuros

Neste trabalho é proposto um novo método que trata o problema de classificação multirrótulo, através da construção de classificadores multirrótulo para cada subconjunto de rótulos obtido a partir da separação do conjunto de dados original, por meio de clusterização hierárquica baseada na relação entre os rótulos.

Para prova de conceito do método proposto, foram selecionadas bases de dados já pré-processadas, utilizadas em outros trabalhos da literatura, a fim de verificar o poder de predição do método proposto em relação a outros bastante conhecidos em aprendizado multirrótulo — os métodos BR, LP, CC — e outro método semelhante ao aqui proposto — o método HOMER. A partir dos resultados de performance, foram realizados testes estatísticos com intuito de validar esses resultados.

O teste estatístico de Friedman mostrou que, das 5 medidas analisadas, 4 delas tiveram diferenças estatísticas nos resultados, são elas: *Hamming Loss*, *Subset Accuracy*, versão micro e macro da medida *F1*.

Os experimentos indicam que algoritmo *MUDIH* apresenta uma performance melhor do que os demais algoritmos em alguns cenários (bases de dados), ou seja, para duas das seis bases de dados o método MUDIH apresenta resultados equiparados aos métodos da literatura ou melhores resultados para duas bases de dados segundo as medidas *Hammin Loss*, F e versões micro e macro da medida *F1*; para quatro das seis bases, o método apresenta melhores resultados segundo a versão micro da medida *F1*; e para três das seis bases, o método proposto apresenta melhores resultados segundo a versão macro da medida *F1*. Assim, para quatro das seis bases bons resultados foram obtidos para quatro das cinco medidas analisadas, o que são considerados resultados promissores.

Para todas as medidas exceto a medida F, o teste de Friedman rejeitou a hipótese nula H_0, portanto foram realizados 4 pós testes utilizando o teste de Nemenyi. Os testes de Nemenyi realizados mostram que o método proposto apresenta melhores resultados que alguns dos métodos comparados, mas não está na primeira posição do ranking. No entanto, o método aparece no mesmo grupo dos melhores resultados em algumas situações, o que também é considerado promissor.

Como limitações deste trabalhos podem ser apresentados os seguintes itens:

- Somente algumas bases de dados foram selecionadas aleatoriamente do site entretanto existem diversas bases de dados que poderiam ser utilizadas;

- Os algoritmos MUDIH e HOMER recebem como parâmetro um algoritmo multirrótulo. Outros algoritmos multirrótulo podem ser utilizados;
- Como parâmetro de todos os algoritmos multirrótulo, neste trabalho foi utilizado o J48. Entretanto poderiam ser utilizados outros algoritmos monorrótulo;
- Outras métricas de distâncias podem ser exploradas utilizando o método MUDIH.

Como trabalhos futuros devem ser realizados experimentos com outras bases de dados com domínios e tamanhos diferentes, a fim de analisar o desempenho do método proposto. Ainda, os resultados obtidos neste trabalho serão comparados com os outros trabalhos relacionados além do trabalho que propõe o HOMER, e com os resultados obtidos no trabalho em desenvolvimento que utiliza a abordagem aglomerativa.

Referências Bibliográficas

[Agrawal et al., 2013] Agrawal, R., Gupta, A., Prabhu, Y., and Varma, M. (2013). Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd international conference on World Wide Web*, pages 13–24. ACM.

[Alvares-Cherman et al., 2012] Alvares-Cherman, E., Metz, J., and Monard, M. C. (2012). Incorporating label dependency into the binary relevance framework for multi-label classification. *Expert Systems with Applications*, 39(2):1647–1655.

[ANDERBERG, 1973] ANDERBERG, M. R. (1973). Cluster analysis for applications.

[Berkhin, 2004] Berkhin, P. (2004). Survey of clustering data mining techniques, 2002. *Accrue Software: San Jose, CA*.

[Berkhin, 2006] Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer.

[Bernardini et al., 2013] Bernardini, F., Silva, R., and Meza, E. (2013). Analyzing the influence of cardinality and density characteristics on multi-label learning. In *Anais do Encontro Nacional de Inteligência Artificial e Computacional 2013*.

[Bernardini et al., 2009] Bernardini, F. C., Garcia, A. C. B., and Ferraz, I. N. (2009). Artificial intelligence based methods to support motor pump multi-failure diagnostic. *Engineering Intelligent Systems*, 17(2).

[Bi and Kwok, 2013] Bi, W. and Kwok, J. T.-Y. (2013). Efficient multi-label classification with many labels. In *ICML (3)*, pages 405–413.

[Boutell et al., 2004] Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771.

[Calembo et al., 2011] Calembo, K. N., Bernardini, F. C., and Martins, C. B. (2011). Proposta de um método de combinaçao de classificadores para construçao de classificadores multi-rótulo. In *Conferência Latinoamericana de Informática—CLEI*, volume 2011.

[Carvalho et al., 2009] Carvalho, A. X. Y., Albuquerque, P. H. M., Almeida Junior, G. R. d., Guimarães, R. D., and Laureto, C. R. (2009). Clusterização hierárquica espacial com atributos binários.

[Cherman, 2013] Cherman, E. A. (2013). *Aprendizado de máquina multirrótulo: explorando a dependência de rótulos e o aprendizado ativo*. PhD thesis, Universidade de São Paulo.

[Cherman et al., 2011] Cherman, E. A., Monard, M. C., and Metz, J. (2011). Multi-label problem transformation methods: a case study. *CLEI Electronic Journal*, 14(1):4–4.

[da Gama et al., 2013] da Gama, P. P., Bernardini, F. C., and Zadrozny, B. (2013). Rb: A new method for constructing multi-label classifiers based on random selection and bagging. *Learning and Nonlinear Models*, 11(1).

[Demšar, 2006] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

[Dimou et al., 2009] Dimou, A., Tsoumakas, G., Mezaris, V., Kompatsiaris, I., and Vlahavas, I. (2009). An empirical study of multi-label learning methods for video annotation. In *2009 Seventh International Workshop on Content-Based Multimedia Indexing*, pages 19–24. IEEE.

[Diplaris et al., 2005] Diplaris, S., Tsoumakas, G., Mitkas, P. A., and Vlahavas, I. (2005). Protein classification with multiple algorithms. In *Panhellenic Conference on Informatics*, pages 448–456. Springer.

[Doni, 2004] Doni, M. V. (2004). Análise de cluster: Métodos hierárquicos e de particionamento. Trabalho de Graduação, Universidade Presbiteriana Mackenzie.

[Elisseeff and Weston, 2001] Elisseeff, A. and Weston, J. (2001). A kernel method for multi-labelled classification. In *Advances in neural information processing systems*, pages 681–687.

[Faceli et al., 2011] Faceli, K., Lorena, A. C., Gama, J., and Carvalho, A. (2011). Inteligência artificial: Uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*.

[Fayyad et al., 1996] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (1996). *Advances in knowledge discovery and data mining*, volume 21. AAAI press Menlo Park.

[Friedman, 1937] Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701.

[Ghahramani, 2003] Ghahramani, Z. (2003). Unsupervised learning. In Bousquet, O., von Luxburg, U., and Rätsch, G., editors, *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*, pages 72–112. Springer.

[Gholap, 2012] Gholap, J. (2012). Performance tuning of j48 algorithm for prediction of soil fertility. *arXiv preprint arXiv:1208.3943*.

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). Unsupervised learning. In *The elements of statistical learning*, pages 485–585. Springer.

[Holmes et al., 1994] Holmes, G., Donkin, A., and Witten, I. H. (1994). Weka: A machine learning workbench. In *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, pages 357–361. IEEE.

[Japkowicz and Shah, 2011] Japkowicz, N. and Shah, M. (2011). *Evaluating learning algorithms: a classification perspective*. Cambridge University Press.

[Kasznar et al., 2009] Kasznar, I. K., Gonçalves, B. M. L., and Bento, M. (2009). Técnicas de agrupamento clustering. *Revista Científica e Tecnológica*.

[Kaufman and Rousseeuw, 1990] Kaufman, L. and Rousseeuw, P. J. (1990). Finding groups in data. an introduction to cluster analysis. *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley, 1990*, 1.

[Klimt and Yang, 2004] Klimt, B. and Yang, Y. (2004). The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, pages 217–226. Springer.

[Macnaughton-Smith et al., 1964] Macnaughton-Smith, P., Williams, W., Dale, M., and Mockett, L. (1964). Dissimilarity analysis: a new technique of hierarchical sub-division.

[MacQueen et al., 1967] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[Marcacini, 2014] Marcacini, R. M. (2014). Aprendizado de máquina com informação privilegiada: abordagens para agrupamento hierárquico de textos. Tese de Doutorado, ICMC/USP.

[Markovsky, 2011] Markovsky, I. (2011). *Low rank approximation: algorithms, implementation, applications*. Springer Science & Business Media.

[Mitchell and Michell, 1997] Mitchell, T. M. and Michell, T. (1997). Machine learningmcgraw-hill series in computer science.

[Modi and Panchal, 2012] Modi, H. and Panchal, M. (2012). Experimental comparison of different problem transformation methods for multi-label classification using meka. *International Journal of Computer Applications*, 59(15).

[Nemenyi, 1962] Nemenyi, P. (1962). Distribution-free multiple comparisons. In *Biometrics*, volume 18, page 263. INTERNATIONAL BIOMETRIC SOC 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210.

[Nogueira, 2009] Nogueira, B. M. (2009). Avaliação de métodos não-supervisionados de seleção de atributos para mineração de textos.

[Pestian et al., 2007] Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D. J., Johnson, N., Cohen, K. B., and Duch, W. (2007). A shared task involving multi-label classification of clinical free text.

In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 97–104. Association for Computational Linguistics.

[Quinlan, 1993] Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.

[Raschka, 2016] Raschka, S. (2014 (Acessado em 28 de Agosto, 2016)). *Machine Learning FAQ*.

[Read et al., 2009] Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2009). Classifier chains for multi-label classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 254–269. Springer.

[Refaeilzadeh et al., 2009] Refaeilzadeh, P., Tang, L., and Liu, H. (2009). Cross-validation. In *Encyclopedia of database systems*, pages 532–538. Springer.

[Rodovalho and Bernardini, 2013] Rodovalho, R. and Bernardini, F. C. (2013). Using artificial datasets to analyze how cardinality and density influence multi-label learning. In *Brazilian Conference on Intelligent Systems 2014*.

[Rokach, 2009] Rokach, L. (2009). A survey of clustering algorithms. In *Data mining and knowledge discovery handbook*, pages 269–298. Springer.

[Tang et al., 2009] Tang, L., Rajan, S., and Narayanan, V. K. (2009). Large scale multi-label classification via metalabeler. In *Proceedings of the 18th international conference on World wide web*, pages 211–220. ACM.

[Trohidis et al., 2008] Trohidis, K., Tsoumakas, G., Kalliris, G., and Vlahavas, I. P. (2008). Multi-label classification of music into emotions. In *ISMIR*, volume 8, pages 325–330.

[Tsoumakas and Katakis, 2007] Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. In *International Journal Data Warehousing and Mining*, volume 3, page 1–13.

[Tsoumakas et al., 2008] Tsoumakas, G., Katakis, I., and Vlahavas, I. (2008). Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, pages 30–44.

[Tsoumakas et al., 2009] Tsoumakas, G., Katakis, I., and Vlahavas, I. (2009). Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 1–19. Springer.

[Tsoumakas et al., 2010] Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010). Mining multi-label data. In *In: (Orgs.) Data Mining and Knowledge Discovery Handbook, 2nd ed.* Springer.

[Tsoumakas et al., 2011a] Tsoumakas, G., Katakis, I., and Vlahavas, I. (2011a). Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):1079–1089.

[Tsoumakas et al., 2011b] Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., and Vlahavas, I. (2011b). Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414.

[Xu et al., 2016] Xu, C., Tao, D., and Xu, C. (2016). Robust extreme multi-label learning. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA August*, pages 13–17.

[Zaïane et al., 2002] Zaïane, O. R., Foss, A., Lee, C.-H., and Wang, W. (2002). On data clustering analysis: Scalability, constraints, and validation. In Cheng, M.-S., Yu, P. S., and 0001, B. L., editors, *PAKDD*, volume 2336 of *Lecture Notes in Computer Science*, pages 28–39. Springer.

Apêndice A

Este apêndice apresenta a Tabela estatística A.1 adaptada de [Japkowicz and Shah, 2011], que é utilizada para realização de testes de hipóteses do teste de Nemenyi. A Tabela A.1 apresenta os valores críticos q_α da distribuição *t de Student*, especificamente para o teste de Nemenyi, onde *gl* representam graus de liberdade, α o nível de significância e A a quantidade de algoritmos.

		Quantidade de algoritmos								
gl	α	2	3	4	5	6	7	8	9	10
5	0.05	2.57	3.25	3.69	4.01	4.26	4.48	4.65	4.81	4.94
	0.01	4.03	4.94	5.52	5.95	6.30	6.59	6.84	7.05	7.24
6	0.05	2.45	3.07	3.46	3.75	3.98	4.17	4.33	4.47	4.59
	0.01	3.71	4.48	4.97	5.35	5.64	5.88	6.09	6.27	6.43
7	0.05	2.36	2.94	3.31	3.58	3.79	3.97	4.12	4.24	4.36
	0.01	3.50	4.19	4.62	4.96	5.21	5.43	5.61	5.78	5.92
8	0.05	2.31	2.86	3.20	3.46	3.66	3.82	3.96	4.08	4.19
	0.01	3.36	3.99	4.38	4.68	4.92	5.12	5.28	5.43	5.56
9	0.05	2.26	2.79	3.12	3.37	3.55	3.71	3.84	3.95	4.06
	0.01	3.25	3.84	4.21	4.49	4.71	4.89	5.04	5.18	5.30
10	0.05	2.23	2.74	3.06	3.29	3.47	3.62	3.75	3.86	3.96
	0.01	3.17	3.73	4.08	4.34	4.55	4.72	4.86	4.99	5.10
11	0.05	2.20	2.70	3.01	3.23	3.41	3.56	3.68	3.78	3.88
	0.01	3.10	3.64	3.97	4.22	4.42	4.58	4.72	4.84	4.94
12	0.05	2.18	2.67	2.97	3.19	3.36	3.50	3.62	3.73	3.81
	0.01	3.05	3.57	3.89	4.13	4.31	4.47	4.60	4.72	4.82
13	0.05	2.16	2.64	2.93	3.15	3.32	3.45	3.57	3.67	3.76
	0.01	3.01	3.51	3.82	4.05	4.23	4.38	4.50	4.62	4.72
14	0.05	2.14	2.62	2.91	3.12	3.28	3.42	3.53	3.63	3.71
	0.01	2.98	3.46	3.76	3.98	4.16	4.30	4.43	4.53	4.62
15	0.05	2.13	2.60	2.88	3.09	3.25	3.38	3.49	3.59	3.68
	0.01	2.95	3.42	3.71	3.93	4.10	4.24	4.36	4.46	4.55
16	0.05	2.12	2.58	2.86	3.06	3.22	3.35	3.46	3.56	3.64

	0.01	2.92	3.39	3.67	3.88	4.04	4.19	4.30	4.40	4.49
17	0.05	2.11	2.57	2.84	3.04	3.20	3.32	3.44	3.53	3.61
	0.01	2.90	3.35	3.63	3.84	4.00	4.14	4.25	4.35	4.43
18	0.05	2.10	2.55	2.83	3.03	3.17	3.30	3.41	3.51	3.59
	0.01	2.88	3.32	3.60	3.80	3.96	4.09	4.20	4.30	4.38
19	0.05	2.09	2.54	2.81	3.01	3.16	3.29	3.39	3.48	3.56
	0.01	2.86	3.30	3.57	3.77	3.92	4.05	4.16	4.26	4.34
20	0.05	2.09	2.53	2.80	2.99	3.15	3.27	3.37	3.46	3.54
	0.01	2.84	3.28	3.55	3.74	3.90	4.02	4.13	4.22	4.31
24	0.05	2.06	2.50	2.76	2.95	3.09	3.21	3.31	3.40	3.48
	0.01	2.80	3.22	3.47	3.66	3.80	3.92	4.02	4.11	4.19
30	0.05	2.04	2.47	2.72	2.90	3.04	3.15	3.25	3.34	3.41
	0.01	2.75	3.15	3.39	3.57	3.71	3.82	3.92	4.00	4.07
40	0.05	2.02	2.43	2.68	2.86	2.99	3.10	3.20	3.27	3.34
	0.01	2.70	3.09	3.32	3.49	3.61	3.72	3.81	3.89	3.96
60	0.05	2.00	2.40	2.64	2.81	2.94	3.05	3.14	3.22	3.29
	0.01	2.66	3.03	3.25	3.41	3.53	3.63	3.71	3.79	3.85
120	0.05	1.98	2.38	2.60	2.77	2.90	3.00	3.08	3.16	3.22
	0.01	2.62	2.97	3.18	3.33	3.44	3.54	3.62	3.68	3.75
∞	0.05	1.96	2.34	2.57	2.73	2.85	2.95	3.03	3.10	3.16
	0.01	2.57	2.91	3.11	3.25	3.37	3.45	3.53	3.59	3.65

Tabela A.1: Valores críticos da distribuição *t de Student* para utilização aplicada ao teste de Nemenyi.